

ANALYSIS OF DYNAMIC BIOLOGICAL SYSTEMS IMAGERY

by

Cameron Drew Dudiak

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Mathematics

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2022

©COPYRIGHT

by

Cameron Drew Dudiak

2022

All Rights Reserved

DEDICATION

This thesis is dedicated to my parents, Andy and Lisa Dudiak. Thank you for teaching me how to learn, and for all of the support you have given me throughout my life.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the extensive mathematical advice and knowledge provided by Dr. Scott McCalla. Additionally, I would like to thank Dr. James Wilking for his incredible joy of science, and dedication to teaching and acceptance for all. I would like to acknowledge Kathleen Ostrem (B.A. Student) for her tireless work in tackling the immense difficulties posed in controlling a microbiological system (Chapter 2). The work conducted in Chapter 3 would not be possible without the confocal imaging obtained from specialist Dr. Barkan Sidar, as well as biological expertise and analysis from Katrina Lyon (Ph.D. candidate) and Dr. Diane Bimczok. A special thank you goes out to Dr. Rama Bansil for her expertise and passion in bio-physics, mucus and organoids. I would also like to express my appreciation for the math department at Montana State University, as well as all faculty who have supported my endeavors in learning.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Bacteria and Partial Differential Equations (PDEs)	2
1.2 Organoids and Microrheology	2
2. NONLOCAL MODELS IN BIOLOGY	4
2.1 Cucker–Smale Flocking	4
2.2 Local and Nonlocal Models for Criminal Activity	6
2.3 Nonlocal Swarming Model	10
3. PAENIBACILLUS DENDRITIFORMIS BOUNDARY TRACK- ING AND OPTIMIZED MODELING	13
3.1 Introduction	13
3.2 Developing the Imaging System	14
3.3 Experimental Setup	17
3.4 Image Processing and Analysis	17
3.5 Nonlocal Model for Repulsive Colony Boundaries	20
3.6 Obtaining Optimal Model Parameters	24
3.7 Materials and Methods	26
3.7.1 Bacterial Strain and Growth Media	26
3.7.2 LB and Plate Inoculation	27
3.7.3 Boundary Tracing Routine	28
3.7.4 Model Training Routine	29
3.8 Results	31
3.9 Conclusion	41
4. PARTICLE TRACKING AND MICRORHEOLOGY IN HUMAN GASTRIC ORGANIDS	42
4.1 Introduction	42
4.2 Particle Tracking	43
4.3 Microrheology Experiments	47
4.4 Methods	52
4.4.1 Particle Tracking Routine	52
4.4.2 Microrheology Routine	54
4.5 Results	55
4.6 Conclusion	58
5. CONCLUSION	60

TABLE OF CONTENTS – CONTINUED

REFERENCES CITED.....	61
-----------------------	----

LIST OF FIGURES

Figure	Page
3.1 The imaging system I developed to obtain six bacterial growth timelapses in one experimental period.	15
3.2 Sample output from the Boundary Tracing Routine. Though only three timesteps are shown in the <i>above</i> figure, the routine returns <i>green (light gray)</i> parameterizations $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ for all colonies, at all time points.	30
3.3 Timelapses of growing bacterial colonies, their obtained parameterizations $\varphi_{t_j}^i$ and their respective optimized nonlocal model boundaries $\gamma_{t_j}^i$. These timelapses were used to find time scalar K . The <i>yellow (light gray)</i> curves are obtained $\varphi_{t_j}^i$, the <i>red (dark gray)</i> curves are simulated $\gamma_{t_j}^i$ using optimal parameters (K, C^*, σ^*) . Timelapses are arranged as columns - <i>left</i> : one-colony timelapse ; <i>center</i> : one-colony timelapse ; <i>right</i> : two-colony timelapse with 2cm spacing.	33
3.4 Timelapses of growing bacterial colonies, their obtained parameterizations $\varphi_{t_j}^i$ and their respective optimized nonlocal model boundaries $\gamma_{t_j}^i$. The <i>yellow (light gray)</i> curves are obtained $\varphi_{t_j}^i$, the <i>red (dark gray)</i> curves are simulated $\gamma_{t_j}^i$ using optimal parameters (C^*, σ^*) . Timelapses are arranged as columns - <i>left</i> : two-colony timelapse with 1cm inoculation spacing; <i>right</i> : two-colony timelapse with 2cm inoculation spacing.	35

LIST OF FIGURES – CONTINUED

Figure	Page
3.5 Timelapses of growing bacterial colonies, their obtained parameterizations $\varphi_{t_j}^i$ and their respective optimized nonlocal model boundaries $\gamma_{t_j}^i$. The <i>yellow (light gray)</i> curves are obtained $\varphi_{t_j}^i$, the <i>red (dark gray)</i> curves are simulated $\gamma_{t_j}^i$ using optimal parameters (C^*, σ^*) . Timelapses are arranged as columns - <i>left</i> : two-colony timelapse with 2cm inoculation spacing; <i>right</i> : additional two-colony timelapse with 2cm inoculation spacing.	36
3.6 Parameters C (<i>left</i>) and σ (<i>right</i>) with respect to colony spacing. <i>Red (gray)</i> ‘x’: learned (optimized) model parameters ; <i>Blue (dark gray)</i> ‘*’: extrapolated model parameters.	37
3.7 Parameters C (<i>left</i>) and σ (<i>right</i>) with respect to the number of colonies plated. As in previous, <i>Red (gray)</i> ‘x’: learned (optimized) model parameters ; <i>Blue (dark gray)</i> ‘*’: extrapolated model parameters.	38
3.8 Initial colony (frame) tracings are obtained as <i>yellow (light gray)</i> $\varphi_{t_0}^i$ and used in conjunction with extrapolated parameters (C^E, σ^E) to obtain modeled <i>red (dark gray)</i> bacterial growth tracings, $\gamma_{t_j}^i$. Timelapses are arranged as columns - <i>left</i> : two-colony 3cm spaced timelapse using extrapolated $(C^E, \sigma^E) = (0.65, 60)$; <i>center</i> : three-colony 2cm spaced timelapse using extrapolated $(C^E, \sigma^E) = (0.425, 30)$; <i>right</i> : additional ‘test’ three-colony 2cm spaced timelapse using extrapolated $(C^E, \sigma^E) = (0.425, 30)$	40

LIST OF FIGURES – CONTINUED

Figure	Page
4.1 ‘Organoid 4’ injected with $1\mu\text{m}$ fluorescent beads - <i>top</i> : Large scale view of ‘Organoid 4’ captured in bright-field ; <i>bottom left</i> : Tracking scale ($20\times$ objective, additional $4\times$ optical zoom) bright-field view of ‘Organoid 4’ at particle tracking position 3 of 8 (24hr timepoint) ; <i>bottom right</i> : Corresponding fluorescent-field image (same region and timepoint as <i>Bottom left</i>).	45
4.2 Example of beads tracked in ‘Organoid 4’ (position 3/8, 24hr timepoint) (<i>left</i>) and their respective plot- ted tracks (<i>right</i>). The tracks display two distinct behaviors: caged diffusion and free diffusion.	47
4.3 Time dependent ensemble MSD profiles of $1\mu\text{m}$ fluorescent beads diffusing through different control media. The <i>bold</i> profiles represent data obtained with the desired imaging conditions ($20\times$ objective, and additional $4\times$ optical zoom). Control MSDs in various media - <i>blue</i> : phosphate buffered saline (PBS) ; <i>black</i> : human gastric mucus (HGM) at 15mg/mL concentration ; <i>red</i> : human gastric mucus (HGM) at 30mg/mL concentration.	50
4.4 MSDs and resultant ensemble complex modulus from ‘Organoid 4’ 24hr after injection - <i>left</i> : indi- vidual (<i>green (light gray)</i>) and ensemble (<i>blue (dark gray)</i>) MSD profiles of all beads tracked ; <i>right</i> : the resultant elastic (<i>red (gray circles)</i>) and viscous (<i>blue (dark gray x’s)</i>) moduli.	52

LIST OF FIGURES – CONTINUED

Figure	Page
4.5	
Calculated $G'(1\frac{rad}{s})$, $G''(1\frac{rad}{s})$ and $\tan\delta(1\frac{rad}{s})$ values for all videos in all organoids at all timepoints following injection. In all plots, <i>red (light gray)</i> points represent ensemble values for one particular position imaged in one one organoid. The <i>black</i> points represent all <i>red</i> points averaged at one timepoint after injection. <i>Top left: $G'(1\frac{rad}{s})$ (elasticity) ; top right: $G''(1\frac{rad}{s})$ (viscosity) ; bottom: $\tan\delta(1\frac{rad}{s})$ ($G''(1\frac{rad}{s})/G'(1\frac{rad}{s})$) values.</i>	59

ABSTRACT

Biological systems pose considerable challenges when attempting to isolate experimental variables of interest and obtain viable data. Developments in image analysis algorithms and techniques allow for further mathematical interpretation, model integration, and even model optimization (‘training’). We formulate two distinct methods for obtaining robust quantitative data from time-series imagery of two biological systems: *Paenibacillus dendritiformis* bacterial colonies, and human gastric organoids. Boundary parameterizations of *P. dendritiformis* are extracted from timelapse image sequences displaying colony repulsion, and are subsequently used to ‘train’ a previously developed nonlocal PDE model through the means of error minimization between observation and simulation. Particle tracking is conducted for small colloidal beads embedded within human gastric organoids, and then used to perform particle tracking analysis. This information is analyzed to quantify the local complex viscoelastic properties of organoids’ interior mucosal environment.

CHAPTER ONE

INTRODUCTION

Image analysis techniques form a vast and growing field with applications ranging from nearly static geospatial information systems, to much more dynamic biological systems. The process of obtaining statistically relevant data from biological systems poses many difficult challenges. Varying experimental conditions may significantly affect the biological system's dynamic behavior and must be monitored and controlled. Because of this, quantitative biological analysis is generally conducted from imagery obtained in a controlled, lab-type, environment. Even so, biological systems pose extreme difficulties in terms of reproducible results and experimental repeatability. Within this thesis, two dynamic biological systems were first imaged over time and then analysed to produce quantitative results from evolving two-dimensional imagery.

Chapter 2 provides a review of nonlocal models and their applications in modern applied mathematics research of biological systems. This nonlocality proves essential to multiple biological behaviors and has recently been extended to repulsive behavior tied to bacterial competitive inhibition [13]. The nonlocal model developed in [13] is of particular interest for this thesis and is later tuned in accordance with obtained imagery of bacterial growth. Chapter 3 will discuss the approach taken to obtain time-dependent imagery data for growing *Paenibacillus dendritiformis* (bacteria capable of competitive inhibition) colony boundaries. The boundary data obtained is then used to train the nonlocal model developed in [13]. Chapter 4 reviews an entirely separate image analysis problem—microrheology analysis within human gastric organoids. This thesis is meant to serve as an exploratory study of image analysis problems in

biological applications, obtaining data and extracting relevant physical information from such observables.

1.1 Bacteria and Partial Differential Equations (PDEs)

Competitive inhibition is a process whereby bacterial colonies repel (sometimes, self-repel) one another in attempt to outcompete the others in search of more desirable spatial regions—where growth media is plentiful and competition is low. In this particular study, *Paenibacillus dendritiformis* is the bacteria of interest—their growth is driven by utilizing nutrient deficient agar plates as a minimal means for colony survival [3]. Through the construction of a unique imaging system, the repulsive bacterial growth was captured in still image sequence format as parameterized curves of the colony boundaries. An optimization routine was next developed which minimizes the error between observed colony boundaries and simulated curves. The model used for simulation was previously developed as a nonlocal PDE model [13] to qualitatively describe bacterial growth according to initial inoculation patterns, as well as the self-repulsive and attractive-repulsive kernels. The kernels are reduced through nondimensionalization to just two (three, with time scaling) parameters, which are tuned according to the observations made through imaging experiments. After learning certain statistical trends for parameters dependent on inoculation patterns, parameter extrapolations are made to test other bacterial growth timelapses obtained.

1.2 Organoids and Microrheology

Organoids are self-assembled tissue cultures commonly used in biological research and drug testing. Little to no information is known about the mechanical

properties of the mucus present within organoids of any type. In this study, established microrheology routines were utilized to survey the viscoelastic properties within organoids. Microrheology is the process whereby a viscoelastic fluid is surveyed by tracking colloidal particles within the medium [1, 12]. Human gastric organoids were first injected with $1\mu\text{m}$ colloidal fluorescent beads and were then imaged using a confocal microscope. Using the obtained imagery, the particles' two-dimensional motion was tracked and later used to evaluate the viscoelastic properties present within the human gastric organoids.

CHAPTER TWO

NONLOCAL MODELS IN BIOLOGY

This chapter is an exploration and summary of several nonlocal models and their current applications in various settings. The study seeks to teach a broad understanding of nonlocal models as used in modern applied mathematics research. Nonlocality is the idea that an agent within a system is not just affected by its immediate (local) surroundings, but also by the entire population of agents within the system. The study first examines the nonlocal model developed in [7] which describes evolutionary flocking behavior for an agent based system. This model was an early attempt in adapting the concepts of nonlocality to biological systems. Further sections in this chapter examine the models used in multiple nonlocal applications of behavior for living systems; including, criminal activity ‘hotspot’ development, Lévy flights in crime, and evolutionary trends due to swarming behavior. This study provides a brief summary for each model examined and discusses the importance of nonlocality in biological system behavior.

2.1 Cucker–Smale Flocking

It has been observed that a population of birds or fish travelling in $\mathbb{R}^3$, will often assume a common velocity of motion after a given period of time. In [7] Cucker and Smale develop a nonlocal counterpart to a previously developed local model describing the evolution of flocks [20]. In Viscek’s model [20], agent i (bird i) adjusts its velocity towards the average velocity presented by its neighbors. The following will address Cucker and Smale’s approach to modifying Viscek’s local model to a nonlocal flocking model.

Just as in [7], we will now consider Euclidean space $\mathbb{E}^3$ instead of the real coordinate space $\mathbb{R}^2$ —for bird i the model becomes [7, 20]:

$$x_i(t+1) = x_i(t) + v_i(t), \quad (2.1)$$

$$v_i(t+1) = \frac{1}{n_i(t)} \sum_{j \in N_i(t)} v_j(t), \quad (2.2)$$

where position and velocity $x_i, v_i \in \mathbb{E}^3$ for birds $i = 1, \dots, k$ and time $t = 0, 1, 2, \dots$. Additionally $N_i(t) = \{j \leq k; \|x_i(t) - x_j(t)\| \leq r\}$, with $n_i(t) = \#N_i(t)$ for some $r > 0$. Note in Viscek's model, $N_i(t)$ restricts interactions to a particular radius r . Extending (2.2) to matrix form, the authors arrive at

$$v(t+1) - v(t) = -D_x^{-1} L_x v(t),$$

where D_x is a $k \times k$ -degree matrix with the i^{th} diagonal entry as $\sum_{j \leq k} a_{ij}$ (where $a_{ij} \in A$, definition to follow) and the Laplacian matrix $L_x = D_x - A_x$. Here, both model's adjacency matrix entries assume the same form [7]:

$$a_{ij} = \eta(\|x_i - x_j\|^2). \quad (2.3)$$

However, the $\eta(y)$ term assumes a different form between models. In Viscek's model η assumes the local form,

$$\eta(y) = \begin{cases} 1 & \text{if } y \leq r^2 \\ 0 & \text{otherwise,} \end{cases}$$

for some $r > 0$. Here, bird i only has interactions with neighbors defined by radius r with uniform weighting. In contrast, Cucker and Smale's model provides a continuous and smooth interactions formulations for agent i (bird i), over the entire flock. With

this nonlocal assumption, the adjacency matrix assumes the same form as before, (2.3), with newly defined

$$\eta(y) = \frac{K}{(\sigma^2 + y)^\beta}, \quad (2.4)$$

where $K, \sigma > 0$ and $\beta \geq 0$ are fixed. This adjustment provides the nonlocal framework within the Cucker–Smale Flocking model [7]:

$$x(t+1) = x(t) + \Delta t v(t), \quad (2.5)$$

$$v(t+1) = \left(Id - \frac{L_x}{\|L_x\|} \right) v(t), \quad (2.6)$$

where Δt is a fixed time step in the numerical model, Id is the $k \times k$ -degree identity matrix and $\|L_x\|$ is the operator norm of L_x .

Authors additionally consider the evolution for the system in continuous time arriving at [7]:

$$x' = v, \quad (2.7)$$

$$v' = -L_x v. \quad (2.8)$$

The remainder of [7] discusses convergence for continuous and discrete time. The model examined exemplifies the behavior of flocked birds (agents) approaching one constant velocity due to nonlocal effects within the population (flock). This model is an important example of the use of nonlocal operators, in this case (2.4), in a biological behavioral setting.

2.2 Local and Nonlocal Models for Criminal Activity

This section aims to discuss the major difference between local and nonlocal operators as applied to modeling criminal behavior. In [17], an agent based

model of criminal activity was developed with aims to reproduce repeat/near-repeat victimization occurrences. Repeat/near-repeat victimization phenomenon is the occurrence of heightened crime risk in an area following a nearby crime. Though the criminals movement is driven by local attractions, this locality limits model agent dynamics and does not model criminal behavior completely.

The model presented in [17] exemplifies the dynamics of offending criminals driven by the attractiveness scalar field $A(\vec{x}, t)$ on a two-dimensional lattice. For simplicity, denote $\vec{x}$ as x for the remainder of this section. The value of the field $A(x, t)$, at location $x = (i, j) \in \mathbb{Z}^2$ gives the stochastic rate of offending for $n(x, t)$ criminals at location x . The attractiveness of a point x is $A(x, t) \equiv A_x(t) \equiv A_x^0 + B_x(t)$, where A_x^0 is temporally static and $B_x(t)$ evolves in time [18]. For every timestep δt , the probability for a criminal to commit a crime is $p_v(x, t) = 1 - e^{-A(x, t)\delta t}$. This probability is based on a standard Poisson process with $A_x(t)\delta t$ the expected number of events at a point $x \in \mathbb{Z}^2$ during time interval δt . If the criminal commits a crime at this time, they are removed from the scheme. If on the other hand they do not commit a crime, they move to an adjacent neighbor x' with probability

$$p_m(x', t; x) = \frac{A(x', t)}{\sum_{x'' \sim x} A(x'', t)}. \quad (2.9)$$

Here, $x'' \sim x$ denotes the immediately adjacent neighborhood of x [17]. Criminals move with a particular speed which is dependent on grid step length l and timestep δt . Therefore an appropriate choice of l and δt must be made to represent a realistic speed of movement. This movement of criminals is modeled as a biased random walk where criminals choose grid steps with greatest probability values $p_m(x', t; x)$. In the real world, criminals often victimize inherently local areas due to their familiarity of the location, the ease of travel and their ability to easily return home. For this

reason, (2.9) takes into account only immediately adjacent neighbors' attractiveness—criminal movement from location x is not influenced by $A(y, t)$ where $y \notin x'' \sim x$.

After criminal behavior for time t has ceased, the attractiveness field is updated by locally smoothing dynamic attractiveness and increasing the attractiveness of perpetrated sites by amount θ . Additionally, the attractiveness field decays exponentially with weight ω . Given the number of events $E_x(t)$ at point x for time t , the attractiveness for the following timestep is defined as:

$$A_x(t + \delta t) = B'_x(t)(1 - \omega\delta t) + \theta E_x(t) + A_x^0, \quad (2.10)$$

$$B'_x(t) = (1 - \eta)B_x(t) + \frac{\eta}{4} \sum_{x' \sim x} B_{x'}(t),$$

where $\eta \leq 1$ [17]. Note that authors restrict the number of neighbors to 4, preserving local dependence; in [18] this value is parameter z , a measure of locality. Lastly, to further simulate criminal behavior, ex-criminals are reintroduced back into the model with a spatially uniform rate Γ .

When executing the simulation, the discrete model revealed three general types of behavior: stationary hotspots, transitory hotspots, or no hotspots at all [17]. In a previous paper [18], authors went further to derive homogeneous equilibrium values for the scenario where all sites have the same attractiveness $\bar{A}$ and on average the same number of criminals $\bar{n}$. In this case, the equilibrium values for the homogeneous system are:

$$\bar{B} = \frac{\theta\Gamma}{\omega}, \quad \bar{n} = \frac{\Gamma\delta t}{1 - e^{-\bar{A}\delta t}}, \quad (2.11)$$

where all parameters are as previously stated [18].

Criminals were converted to criminal density $\rho(x, t)$, and the limit $l, \delta t \rightarrow 0$ is taken, with the added constraint $\frac{l^2}{\delta t} = D$ (D is the diffusion coefficient). Taking the

continuum limit of their discrete model, authors arrive at the two coupled PDEs [17]:

$$\frac{\partial A}{\partial t} = \eta \nabla^2 A - A + A^0 + \rho A, \quad (2.12)$$

$$\frac{\partial \rho}{\partial t} = \nabla \cdot \left[\nabla \rho - \frac{2\rho}{A} \nabla \right] - \rho A + \bar{A} - A^0. \quad (2.13)$$

Details for this derivation can be found in [18]. Two of the three behaviors present in the discrete model are present in the PDE system. Authors believe statistical noise contributed to the formation of transitory hotspots and because Eqs. (2.12,2.13) are derived using only expectation values, the statistical noise is removed and transitory hotspot formation is not observed [17]. The discrete model was extended through the continuum limit to a continuous PDE system—though, it remains local in that its operators do not extend throughout the entire domain.

It is reasonable to assume nonlocality is essential in the evolutionary behavior of criminals and crime density over two-dimensional space. As in previous, burglary hotspot models are generally defined to include only local attractions, with criminals' motion constricted to specific neighborhoods. In [6], authors aim to create a nonlocal model in which criminals move by Lévy flights—biased random motion where agents may move great distances efficiently towards some region of higher attractiveness.

Starting with a cellular automata model, authors next take the continuum limit resulting in a system of two PDEs; a local PDE governing the attractiveness field, and a nonlocal criminal density PDE [6]:

$$\frac{\partial A}{\partial t} = \eta A_{xx} - A + \alpha + A\rho, \quad (2.14)$$

$$\frac{\partial \rho}{\partial t} = D[A\Delta^s \left(\frac{\rho}{A} \right) - \frac{\rho}{A} \Delta^s(A)] - A\rho + \beta. \quad (2.15)$$

To generate super-diffusive behavior (the Lévy Flights) into the criminal behavior model, a nonlocal operator, the fractional Laplacian Δ^s was included in the criminal density equation (2.15). It is important to note that crimes are restricted to stationary targets and residences, where burglaries are treated as random probability events. Criminals may undergo long jumps given an attractive enough burglary site. The attractiveness field which drives these burglars can be decomposed into two parts; the static background term, which is consistent across time and the dynamic term, which evolves with time. The dynamic term takes into account repeat/near-repeat and broken windows effects. The relative attractiveness weights, for a criminal to move from one point to another, are then calculated using a power law distribution and taking into account the attractiveness field. This naturally leads to transition probabilities for a criminal moving from one point to another [6].

When the continuum system was analyzed, induced Turing instabilities were observed to generate hotspots—“localized pulses, rather than dispersion-type waves” [6]. Both the continuous and discrete systems developed in this paper observe the development of hot spots. When authors adjust the effects of the fractional Laplacian, hotspot formation is significantly altered such that the nonlocal operator is directly related to the number of hotspots formed. In this section both local and nonlocal models for criminal activity have been observed to aid in the generation of hotspots. By utilizing the fractional Laplacian, [6] incorporates nonlocal Lévy flights as means for more realistic human behavior, when compared to localized neighbor-neighbor interactions, as in [18].

2.3 Nonlocal Swarming Model

The papers [4] and [8] both explore two-dimensional pattern forming solutions to a nonlocal model consisting of particles driven by the gradient flow of a nonlocal

interaction potential [8]:

$$\frac{dx_j}{dt} = \frac{1}{N} \sum_{n,k \neq j} F(|x_j - x_k|) \frac{x_j - x_k}{|x_j - x_k|}, \quad j = 1, \dots, N. \quad (2.16)$$

This equation arises from the nonlocal pairwise energy formulation [8]:

$$E(\vec{x}) = \sum_{i \neq j} P(|x_i - x_j|), \quad (2.17)$$

where P is some potential. Additionally, the paper restricts its study to two families of interaction forces $F(r) = -P'(r)$ [8]: the power law force

$$F(r) = r^p - r^q; \quad 0 \leq p < q, \quad (2.18)$$

and a smoothed step discontinuity

$$F(r) = \tanh[(1-r)a] + b; \quad 0 < a; -\tanh(a) < b < 1. \quad (2.19)$$

Interaction potentials respective to these forces have short range repulsive and long range attractive tendencies—both of which are significant to the formation of steady state ring solutions as seen in [4]. The ring is a particular solution to (2.16) where particles can be viewed as a one-dimensional curve. Taking the limit $N \rightarrow \infty$ gives the continuum PDE system [4]:

$$\frac{\partial}{\partial t} \left(\rho \left| \frac{\partial z}{\partial \alpha} \right| \right) = 0, \quad \frac{\partial z}{\partial t} = K * \rho. \quad (2.20)$$

Here, $z(\alpha; t)$ is the curve parameterization, $\rho(\alpha; t)$ is particle density, and

$$K * \rho = \int F(|z(\alpha) - z(\alpha')|) \frac{z(\alpha) - z(\alpha')}{|z(\alpha) - z(\alpha')|} \rho(\alpha', t) dS(\alpha'),$$

where dS denotes the arclength element [4]. Here, $K * \rho$ is the nonlocal operator in this model system. Nonlocality proves essential in the formation of the steady state ring solutions examined later in [8]—specific details for stability characterization of the ring solutions, as well as further analysis are available in [4].

The next chapter will make use of a nonlocal model, posed by authors of [13], which uses a nonlocal operator formulated specifically according to mean curvature of the evolving fronts. In addition to this nonlocal forcing, a normal force provides the necessary framework to model bacterial colony boundary evolution as driven by mean curvature flow and nonlocal attractive-repulsive interactions of bacterial colony boundaries.

CHAPTER THREE

PAENIBACILLUS DENDRITIFORMIS BOUNDARY TRACKING AND
OPTIMIZED MODELING3.1 Introduction

In recent years, nonlocal models have become increasingly useful in biological modeling applications. As previously mentioned, these models take into account interactions across the entire domain of interest, not just those in the immediate neighborhood. Though there are many biological applications for which nonlocal models have proved useful, little to no research has been conducted with the interest of training these models according to actual obtained data for the modeled system.

This chapter details the process of collecting growing bacterial colony boundary data through analysis of the obtained timelapse imagery. Competitive-inhibitory (*Paenibacillus dendritiformis*) bacteria—which develop arrested fronts as result of competition for resources—are prepared, plated and then imaged while growing on the nutrient-deficient agar plates. The time series still imagery is next used to parameterize bacterial boundaries over time, with the specific interest of capturing the repulsive behaviors exhibited. The study next examines a particular model developed in [13] which aims to qualitatively predict the boundary of growth for all colonies plated. By minimizing the error between observational data and simulation, optimal model parameters are extracted for various inoculation conditions. The parameters learned are then *loosely* extrapolated to further inoculation conditions and the resulting simulation is tested with actual obtained bacterial growth imagery.

Before proceeding with exact processes needed to conduct this study, background information is necessary to understand the biological behavior of interest.

Competitive inhibition is the process whereby bacterial colonies display short-range repulsive behavior. *Paenibacillus dendritiformis* is the specific bacterium which is imaged for this study and used to examine the nonlocal model capabilities of [13]. Specifically, the C-morphotype (rod-shaped growth) of *P. dendritiformis* displays the desired competitive-inhibitory response when multiple colonies are inoculated on the same nutrient-deficient agar plate [3]. Upon recognizing the competition for needed resources, bacterium within the colony secrete a lethal biosurfactant. The biosurfactant secreted is known as sibling lethal factor (SLF) protein [5]. The effect this competitive response has on bacterial colonies can effectively be reduced to the behavior of colony boundaries in the system. The nonlocal model and methods used in this study restrict the observational data of interest to colony boundaries, where their growth and repulsion only depend on colony boundaries within the system.

3.2 Developing the Imaging System

To capture still image sequences for multiple plates within one experimental period, a temperature and humidity controlled imaging system was developed. The imaging setup, as pictured in figure 3.1, has dimensions of 22in length, 22in width, and 30in height and is covered with black, completely opaque, welded acrylic sheeting. The box is completely light tight and sufficiently air tight by utilizing weather seals along all edges of the lockable door. There is one DSLR camera within the incubator which, currently, has the capability to image six plates during one experiment. The main supports for the structure were constructed from half inch PVC piping which holds the camera in a stable/vertical manner pointing downwards. Approximately eight inches below the camera lens lies a turntable holding six plates spaced evenly such that, 60 degrees of rotation allows the next plate to be imaged in the exact manner as the previously imaged plate. This motion is powered by a stepper motor

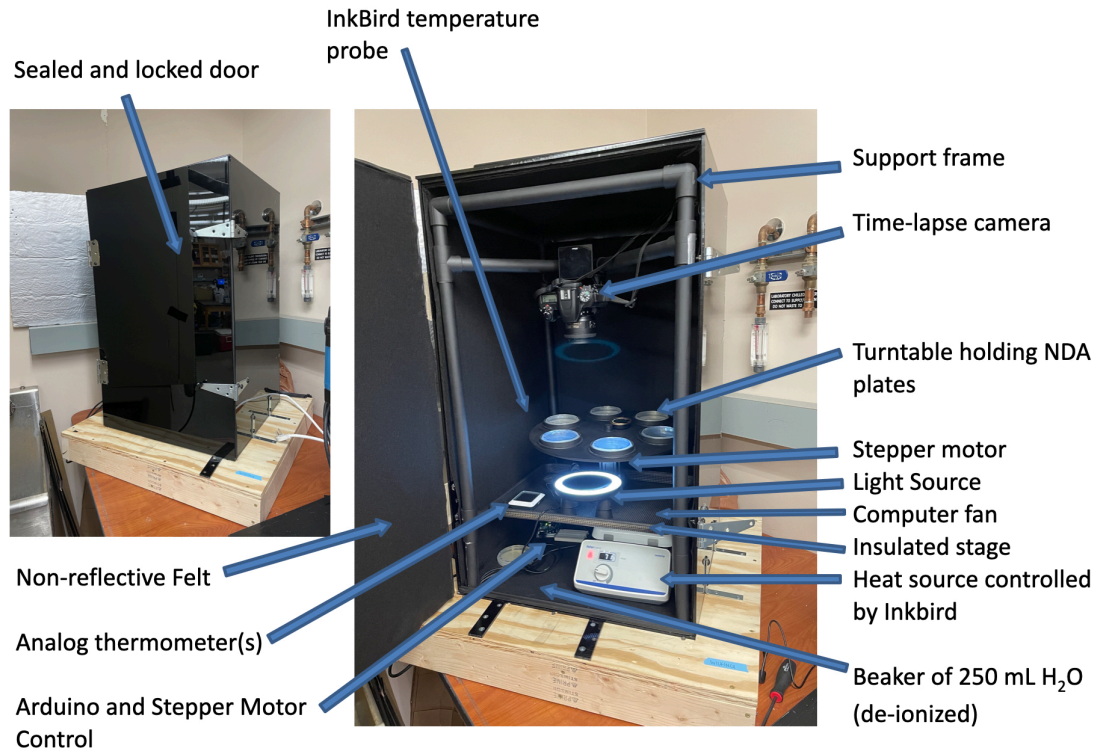


Figure 3.1: The imaging system I developed to obtain six bacterial growth timelapses in one experimental period.

which is triggered via an Arduino micro-controller programmed to rotate 60 degrees, every five minutes. Therefore, each image sequence will have 30-minute time intervals between each frame, or, frame rate of 2 frames per hour.

Lighting conditions for this setup were extremely difficult due to the similar optical qualities of nutrient deficient agar (growth media comprising the 2D surface imaged in this study—NDA) and our growing bacterial colonies. The colonies' boundaries are of particular interest for this study, thus there must be a clear distinction between boundaries and the NDA media upon which they live. To address this challenge, an individual ring light (with diameter slightly larger than the plates) was placed about 4 inches below the imaging position such that the camera cannot

receive direct light. Additionally, any exposed PVC within the incubator was painted matte black to avoid light interference. Similarly, all shiny, or glaring, surfaces within the incubator were covered with a particular black fabric which was found to have minimal light interference. By eliminating glare or reflection within the incubator, light illuminates only the plate immediately above the ring and below the camera. Light intensity and position were systematically varied until the images captured were sufficiently good.

Temperature and humidity are both controlled and monitored, allowing for experiments to be repeatable and maximally informative. As the NDA plates have been sealed, humidity is of less importance and was simply controlled by placing a 250mL beaker of deionized water inside the sealed incubator prior to each experiment. Because the incubator is sealed and water content within the incubator should not escape, the humidity is able to remain relatively constant. Internal temperature was controlled by a hotplate which lies 5 inches below the platform holding the stepper motor. Using an InkBird temperature controller, the hotplate is turned on when the internal temperature dips 0.25°C below the desired temperature (29.5°C was found to be ideal), and is turned off when the temperature rises 0.25°C above the desired temperature. Utilizing a thermal probe placed near the turntable, this controller constantly adjusts power to the hotplate, thus, controlling the internal temperature. In fear of temperature gradients being established, a small computer fan was added to circulate the internal air. Temperature and humidity are monitored and recorded using a Temtop-brand thermocouple and hygrometer, along with multiple analog thermometers. I next conducted experiments to identify the incubator's ideal climate to grow *P. dendritiformis* bacteria, which exhibit the repulsive phenomenon. A relative humidity of 30% and temperature of 29.5°C proved to be an ideal environment for the bacteria.

3.3 Experimental Setup

I next developed an experimental setup protocol to best provide consistency during and between experiments. The protocol is essential to capturing bacterial growth and repulsion in a controlled environment. The details for the exact steps taken to prepare the plates can be found in the ‘Materials and Methods’ section.

Once the plates are prepared, they are immediately loaded onto the 6 turntable slots within the imaging system. Temperature and humidity controls are checked and adjusted if necessary. Next, the camera is programmed to capture an image every 5 minutes—coinciding with each stepper motor rotation. Camera focus and the turntable’s position are now adjusted with live view, easily enabling proper image composition and focus. The stepper motor is turned on, the camera capture is started and the door is locked.

Plates are left for four days, at which point they are removed from the incubator and all data is retrieved. Raw images are first examined to determine the first and last frames of interest. The bacteria often experience a brief pause before expanding outwards, resulting in up to 24 hours of stationary imagery. The first image used for tracking must be one obtained once the bacteria have already started growing. When the bacteria reach within one centimeter of the plate boundary, or the competitive front has ceased, the timelapses are concluded.

3.4 Image Processing and Analysis

The desired raw images are imported into MATLAB where they are cropped and separated into their respective *mod6* timelapse. From here on, each image sequence is processed independently. Batch exposure control, image smoothing (10%) and curve manipulation are performed in Adobe Camera Raw. RGB curve manipulation

is conducted to further differentiate the bacterial colony and background media. It is important to note the curve adjustments made are not entirely necessary, and all processing may be conducted solely in MATLAB. That being said, Camera Raw’s ability to perform curve adjustments makes boundary tracing (mentioned later) much simpler. Images are exported as gray-scale tiff files (dimensions: 1080 pixels by 1080 pixels) and are now ready for the boundary tracing routine.

Images are once again imported into MATLAB where the first image’s bacterial colony boundaries are obtained using the script ‘PD_boundary_0’. Once the first image’s boundaries have successfully been obtained, a separate script, ‘PD_boundary_F’, obtains the boundary data for the remaining images in the sequence. I wrote both scripts, ‘PD_boundary_0’ and ‘PD_boundary_F’, specifically for this routine. The boundary tracing routine consists of multiple steps and multiple functions—instructions on how to run scripts, ‘PD_boundary_0’ and ‘PD_boundary_F’, can be found in the ‘Materials and Methods’ section.

Within the first script, ‘PD_boundary_0’, prior to tracing the first image’s boundaries, a few image filters are applied. First, a Gaussian filter is added to further smooth the image. This filter utilizes a Gaussian kernel with an adjusted standard deviation of 5 pixels. The nonlocal model later used in fitting the obtained data focuses on a smooth curve outlining the bacterial colonies. Finger-like extensions and jagged colony edges cause problems in boundary tracing and are of different scale from the nonlocal repulsion of interest. At this point, a manual search for binary threshold, which best defines the bacterial boundaries, is performed, using MATLAB’s ‘imbinarize’ function. The ‘imbinarize’ filter converts the grayscale image to a binary image of equal dimension; assigning ones to entries higher than the defined threshold and zeros otherwise. Defining the binary threshold is one of the most cumbersome steps in the routine, but is of upmost importance and must be performed for each

image sequence. This search could be automated in the future by minimizing some term involving curvature, arc length and local step specifications; however, this is beyond the scope of this study.

The binary image is next run through MATLAB's 'bwboundary' function which traces the boundaries of all objects: collections of ones, in the image. With all boundaries traced in the image defined, it is up to the user to choose which points begin the tracing of the actual colonies. By selecting one point from each of the n colonies, the program stores these entries as initial tracing points for MATLAB's 'bwtraceboundary' function. The number of colonies is previously defined by the user, along with the beginning and end images of interest. Next, the script executes MATLAB's 'bwtraceboundary' with predefined starting points and initial step direction. The function works by searching in the predefined direction for another value zero pixel which edges the value one colony. Essentially, this is a nearest 'desirable' neighbor technique. For the remaining parameterization, the function searches for pixels which edge the colony and are connected to the existing parameterization. Once the 'bwtraceboundary' function is completed, the complete parameterizations of all colonies are stored and the remaining frames can be analysed.

The user must next run the script 'PD_boundary_F'. The code first applies all filters that were used in 'PD_boundary_0' to the remaining images in the sequence. The binary threshold used in the filter 'imbinarize' does not change throughout the sequence because lighting conditions are kept constant within the incubator. Once the filters are applied, boundaries from all colonies are collected from the remaining frames, using 'bwtraceboundary'. The initial step directions used for each colony do not change, however, the initial points of interest do need adjustment for each new frame. In order to address the changing starting points used in the tracer function, the script searches the pixels nearby the last frames' initial point for each colony. The

script searches for these new initial points using a two dimensional biased random search with a normal distribution centered around the previously used starting point. With this method, the manual search for initial points is automated and completely contained within the script ‘PD_boundary_F’.

The boundary tracing routine is complete when all colonies have successfully been traced for all image frames in time. Every timestep t_j , the boundary parameterization is stored, and in the end, a complete timelapse of the boundaries, as parameterized curves, has been obtained. For discrete time points $\mathbf{t} = \{t_0, \dots, t_F\}$, the parameterization of colony i is stored as $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ such that $(\mathbf{x}, \mathbf{y}) = [(x_1, y_1), \dots, (x_k, y_k)]$, where $k \in \{1, \dots, L\}$. As such, the collection of evolving parameterizations becomes $\bar{C} = \{\varphi^1, \dots, \varphi^N\}$. Length of the parameterization L is defined, arbitrarily, by ‘bwtraceboundary’ and can vary with time. Later, when using the obtained boundary to train a numerical scheme, length L of the parameterization will be constrained to temporally constant $\bar{L}$ through polygonal interpolation.

3.5 Nonlocal Model for Repulsive Colony Boundaries

Certain bacteria—in our experiments, *Paenibacillus dendritiformis*—present a phenomena known as competitive inhibition, whereby arrested fronts develop when colonies are near enough to one another. In [13] a nonlocal model was developed with aims to model this phenomena. Nonlocal models have been used extensively to study physical systems, such as the movement of vortex sheets and vortex patches [10, 13]. These systems feature abrupt jump discontinuities between regions of different vorticity which lead authors of [10] to a nonlocal equation governing the evolution of the moving fronts. Competing interactions between bacterial fronts lead to a similar class of nonlocal evolution laws governing pattern formation in certain biological systems [13]. As a result, authors of [13] adapt the framework for

studying domain boundaries in vortex driven processes to understand the evolution and eventual cessation of interacting biological fronts. The model utilizes nonlinear, nonlocal interactions, which lead to mutually repulsive forces capable of arresting the motion of all colonies in the closed boundary system. For the experiments conducted within this study, Petri dish edges act as the eventual boundaries.

This nonlocal model reduces the complexity commonly found in comparable coupled systems of reaction-diffusion equations which are also used to study counter-propagating front dynamics. While still capturing the essential pattern forming mechanisms, this model only considers codimension-one interfaces—the repulsive fronts. It is important to note that the model framework used can be extended to multiple dimensions, yet in this instance the framework is restricted to a specific setting: two-dimensional pattern formation as result of competing sibling bacteria (*Paenibacillus dendritiformis*).

The general class of models presented in [13] describes the evolution of a collection, $C = \{\gamma^1, \dots, \gamma^N\}$, of N mutually interacting two dimensional curves:

$$\gamma^i(x, t) : D \times \mathbb{R}^+ \mapsto \mathbb{R}^2, \gamma^i(x, t) = \begin{bmatrix} \gamma_1^i(x, t) \\ \gamma_2^i(x, t) \end{bmatrix}.$$

Here, D denotes a fixed Lagrangian coordinate domain. A nonlocal force $v^i(x, t)$ acts in a known direction, to leading order, $\omega^i(x, t)$ on $\gamma^i(x, t)$. The magnitude of the nonlocal force is dependent on self-repulsion and the repulsive-attractive effects from the remaining colonies $\gamma^j \in C$. From these assumptions, the nonlocal velocity field $\mathbf{v}_{NL}^i(x, t) := v_{NL}^i(x, t)\omega^i(x, t)$ is obtained with magnitude given by:

$$v_{NL}^i(x, t) := \oint_D f_i \left(\frac{1}{2} |\gamma^i(x, t) - \gamma^i(z, t)|^2 \right) |\partial_x \gamma^i(z, t)| dz$$

$$+ \sum_{j \neq i} \oint_D g_{ij} \left(\frac{1}{2} |\gamma^i(x, t) - \gamma^j|^2 \right) |\partial_x \gamma^j(z, t)| dz,$$

where $f_i(s)$ are the kernels describing self interaction of γ^i and $g_{ij}(s)$ are the kernels describing the nonlocal attractive-repulsive influence of γ^j on γ^i . Along with the nonlocal forces, every interface undergoes additional motion driven by a combination of curvature $\kappa_{\gamma^i}(x, t)$ and constant normal growth [13]. These assumptions lead to a general class of front evolutions:

$$\partial_t \gamma^i(x, t) = [\epsilon^i \kappa_{\gamma^i}(x, t) + c^i] \mathbf{n}_{\gamma^i}(x, t) + \mathbf{v}_{NL}^i(x, t). \quad (3.1)$$

Here, $\mathbf{n}_{\gamma^i}(x, t)$ is the outward normal unit vector for each curve γ^i .

Model parameters are eventually reduced to functional forms of the nonlocal kernels $f_i(s)$ and $g_{ij}(s)$, the relative curvature strength ϵ^i and the constant speed of normal growth c^i . To reduce parameters further, the kernels are identical nonlocal interactions with Gaussian decay

$$f_i(s) = g_{ij}(s) = g_{C,\sigma}(s) := \left(\frac{C}{\sqrt{2\pi}\sigma} \right) e^{-s/\sigma^2}, \quad (3.2)$$

where C and σ are the strength and length scale parameters, respectively. Authors of [13] next assume identical curvature $\epsilon^i \equiv \epsilon$ and identical normal growth $c^i \equiv c$ which, via nondimensionalization, allows them to set $\epsilon = c = 1$. Therefore, the system of equations (3.1) is governed by a single relative strength parameter, $\tilde{C} = \frac{C}{c}$, and a single relative length scale parameter, $\tilde{\sigma} = \frac{\sigma c}{\epsilon}$. As such, the model has two governing parameters, with omitted tilde, C and σ . These parameters will later be tuned to fit the model to obtained data.

In [13], a numerical simulation was constructed in accordance with (3.1) using identical Gaussian kernels (3.2). It is important to note that the numerical simulation

describes the qualitative growth of repulsive bacterial colonies and ignores the sub-scale details present on colony edges. Authors use a simple implicit update for curvature and explicit updates for all other forces acting on γ^i to numerically integrate (3.1). Exact solutions to (3.1) are unattainable using this simulation. Asymptotic analysis, however, was used to estimate key features in the experimental runs. For example, the model possesses the means necessary to obtain counter-propagating fronts with a quantifiable separation s . Parameter C is found to provide a leading order approximation consistent with the colony's self-interaction. To approximate the nonlocal forcing from an additional colony $\mathbf{p}$ on γ , authors use Laplace's method to approximate

$$\int_{-\pi}^{\pi} e^{-|\mathbf{p}-\gamma(z)|^2/2\sigma^2} |\partial_x \gamma(z)| dz \approx \frac{\sqrt{2\pi}\sigma}{\sqrt{1-s\kappa_\gamma(z_0)}} e^{-s^2/2\sigma^2},$$

where $s = |\mathbf{p} - \gamma(z)|$ is the minimum separation between fronts γ and $\mathbf{p}$, and $\kappa_\gamma(z_0)$ is the curvature of γ at the point with minimum separation from $\mathbf{p}$. In nondimensionalized variables,

$$[1 - C + \kappa_\gamma(z_0)] \sqrt{1 - s\kappa_\gamma(z_0)} = C e^{-s^2/2\sigma^2}$$

can be used to solve for the arrested front separation s .

This numerical simulation proves capable of producing accurate qualitative time and spatial (two-dimensional) dependent boundary data for repulsive *Paenibacillus dendritiformis* colonies, as seen in experiments from [13], and those conducted later in this study. Through previously described minimalistic model framework, the numerical simulation relies on relative strength, C and relative length scale, σ , of the counter-propagating front evolution. The remainder of this study aims to use previously established optimization techniques—specifically, the Nelder–Mead

simplex method—to fit the numerical model to newly obtained biological data.

3.6 Obtaining Optimal Model Parameters

I next developed an optimization routine to tune parameters of the numerical model (3.1) to fit the evolving simulated boundaries, $C = \{\gamma^1, \dots, \gamma^N\}$, to the obtained time-dependent colony boundary parameterizations, $\bar{C} = \{\varphi^1, \dots, \varphi^N\}$. The routine works by minimizing the error between the data obtained and the coincident boundaries created in simulation using the nonlocal model presented in [13]. The initial frame’s data, $\{\varphi_{t_0}^1, \dots, \varphi_{t_0}^N\}$ obtained from the traced parameterization is used as the initial spatial conditions for the model—defining the curves $\{\gamma_{t_0}^1, \dots, \gamma_{t_0}^N\}$ which are to evolve. The strength C and the length scale σ of the nonlocal kernels, f_i and g_{ij} used in the numerical simulation will be the major parameters of interest. However, prior to considering just two parameters, a minimalistic search for appropriate time scalar K (in addition to (C, σ)) was conducted. This time scaling is essential to the evolution of the model and K will remain constant throughout all experiments. Later, using greater-resolution curves, a search is conducted to more precisely find model parameters, (C, σ) , which minimize time-accumulated error E . Prior to addressing how the error is defined between C and $\bar{C}$ at time t_j , the parameterizations $\{\varphi_{t_j}^1, \dots, \varphi_{t_j}^N\}$ must be interpolated to length $\bar{L}$ to be compatible with simulated boundaries $\{\gamma_{t_j}^1, \dots, \gamma_{t_j}^N\}$. The function, ‘interppolygon’ was used to interpolate L -component curves $\varphi_{t_j}^i$, and returns $\bar{L}$ -component $\bar{\varphi}_{t_j}^i$. For simplicity, the interpolated curves will maintain their original notation, $\bar{\varphi}_{t_j}^i \equiv \varphi_{t_j}^i$, for the remainder of this study.

Now that simulated and observed curves are of the same length ($\bar{L}$), we define

the total error returned as follows:

$$E = \sum_{t_j \in \mathbf{t}} \sum_{i \in \mathbf{I}} (\chi_{t_j}^i)^2 \xi_{t_j}^i \quad , \quad (3.3)$$

where $\mathbf{t} = \{t_0, \dots, t_F\}$, $\mathbf{I} = \{1, \dots, N\}$, and,

$$\chi_{t_j}^i = \|\varphi_{t_j}^i(\mathbf{x}, \mathbf{y}) - \gamma_{t_j}^i(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})\|_{l^2} \quad ; \quad \xi_{t_j}^i = \|\varphi_{t_j}^i(\mathbf{x}, \mathbf{y}) - \varphi_{t_j}^i(\mathbf{x}', \mathbf{y}')\|_{l^2}. \quad (3.4)$$

Here, $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ denotes a reordering of $(\mathbf{x}, \mathbf{y})$ such that every observed point $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ corresponds to the nearest simulated point $\gamma_{t_j}^i(\tilde{x}, \tilde{y})$. Additionally, denote $(x'_l, y'_l) = (x_{l+1}, y_{l+1}) \forall l \in \{1, \dots, \bar{L}\}$ such that $\xi_{t_j}^i$ reflects the l^{th} arclength component of $\varphi_{t_j}^i(x, y)$. Notice that $\xi_{t_j}^i$ is used to scale $\chi_{t_j}^i$ so that the error is appropriately weighted throughout time.

Script ‘multicolony’—previously created for numerical simulation of the [13]—was adapted to include this error. This script simulates the evolution of curves $\gamma_{t_j}^i$ for $t = \{t_1, \dots, t_F\}$; as previously mentioned, the numerical simulation is formulated in accordance with (3.1). Using the initial frame $\gamma_{t_0}^i \equiv \varphi_{t_0}^i$, the simulation evolves these curves with time step size dt . The time step dt must be chosen appropriately such that $\frac{1}{dt} = z(K)$ for some $z \in \mathbb{Z}_+$, where $K \in \mathbb{Q}$ is the time scaling parameter. This way, the error between simulation $\gamma_{t_j}^i$ and observed $\varphi_{t_j}^i$ will be calculated whenever $t \bmod(K) = 0$, where t is overall time and K is the time scalar. After alterations, ‘multicolony’ now accepts input parameters C and σ to evolve the simulated curves, while simultaneously calculating the error between $\gamma_{t_j}^i$ and observed colony parameterizations $\varphi_{t_j}^i$. When the script has concluded, it returns the total accumulated error as formulated in (3.3).

Now that ‘multicolony’ is formulated appropriately, the built-in MATLAB

function ‘fminsearch’ (applied to ‘multicolony’) is used to find parameters (C^*, σ^*) which minimize the total accumulated error between simulation and observation. The function uses the Nelder–Mead simplex algorithm as formulated and described in [9]. Simplex methods do not numerically calculate gradients or derivatives which provides a unique flexibility in their implementation. The method first creates a simplex—essentially, a local expansion—around the initial guess, (C^0, σ^0) and orders the simplex entries with respect to their resultant error. The entry with the largest error is thrown out and a new entry is accepted for consideration. It is worth noting that in some instances, the simplex may change completely, but this is rare. Exact steps for how new simplex entries are created, as well as information regarding the exact numerical method are detailed in [9]. Ultimately, ‘fminsearch’ concludes when a certain stopping criteria is met, and then returns the optimal model parameters (C^*, σ^*) .

3.7 Materials and Methods

3.7.1 Bacterial Strain and Growth Media

The bacterial strain of interest is *Paenibacillus dendritiformis* (30A1). Nutrient deficient agar (NDA) medium was made from *NaCl* (5g/L), *K₂HPO₄* (5g/L), and Bacto-Peptone (2g/L). Difco-Agar (13g/L) was added to solidify the media. A total volume of 250mL of NDA was prepared, autoclaved for 20 minutes, and then the molten media (70°C) was poured into 12 Petri dishes—each with a volume of roughly 20mL. After ten minutes, NDA-plates were placed in a closed room-temperature container, along with a 50mL beaker of deionized water, and left to dry for 72 hours. The procedure outlined is based on methods and conditions used in [3].

3.7.2 LB and Plate Inoculation

The *P. dendritiformis* bacteria are stored at -80°C in Luria broth (*LB*) with 25% glycerol. A total volume of 500mL of Luria broth was prepared using *Fisher Bioreagents* LB solute 20g/L and deionized water. In a sterile 250mL flask, LB is inoculated using the frozen *Paenibacillus dendritiformis* stock. The inoculated culture is then grown for 18 hours within a calibrated incubator adjusted to 30°C with shaking of 150 revolutions per minute. After the 18hr growth period, the optical density of the inoculated broth was examined to ensure proper growth. To obtain ideal bacterial growth on the plates, an optical density (measured at wavelength 650nm) of $OD_{650nm} = 0.8 \pm 0.05$ proved desirable.

Now that the plates have dried for 72 hours and the bacterial culture has reached the desired optical density, the plate inoculations may commence. The six ‘best looking’ plates are arranged around a lit Bunsen burner which provides convective air flow, ensuring sterility of the samples. Approximately 15 seconds prior to inoculating a given plate, the bacterial culture is vortexed (*touch* setting - Thermolyne Type 37600 Mixer) for ten seconds. Once the culture is nicely homogenized, a micro-pipette is used to inoculate the plate with $2\mu\text{L}$ circular colonies. The vortexing and plating is repeated for all six NDA plates while exposing only one plate at a time. Once all colonies have been inoculated, plates are loaded into the imaging incubator. After ensuring all incubator systems are correctly functioning, the timelapse is started and the imaging door is closed. Though multiple modifications were made to the methods of growing *Paenibacillus dendritiformis* to the C-morphotype, the general procedure conducted in this section can be attributed to authors of [3] and [5].

3.7.3 Boundary Tracing Routine

The following instructions provide the exact details needed to obtain bacterial colony parameterizations evolving over time, given an input image sequence (*.jpg*, *.tiff*, *.png*). The scripts ‘PD_boundary_0’ and ‘PD_boundary_F’, as well as the following instructions provide a refined routine necessary to obtain the desired parameterizations with relative ease.

Obtaining Parameterizations $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$:

Before proceeding, ensure all images of interest, along with ‘PD_boundary_0.m’ and ‘PD_boundary_F.m’, are loaded into the current directory. This routine is compatible with MATLAB 2020b (or newer) and requires MATLAB’s Image Processing Toolbox.

In the MATLAB command window...

- 1: Type “ *close all ; clear all ; clc* ” .
- 2: Type “ *PD_boundary_0* ” .
- 3: Input the file name of the first image in the sequence as “ [‘ ###.tiff ’] ” .
- 4: Type “ ‘ *n* ’ ” .
- 5: Type “ ‘ *n* ’ ” .
- 6: Select desired cropping for all images, *double click* and exit the figure.
- 7: Type “ *PD_boundary_0* ” .
- 8: Again, input “ [‘ ###.tiff ’] ” .
- 9: Type “ ‘ *y* ’ ” .
- 10: Type “ ‘ *n* ’ ” .

* If the binary figure appropriately describes each colony boundary, proceed to Step 11. If not, exit the figures, adjust the parameter *TB* in ‘PD_boundary_0.m’ and restart from Step 7. *

11: In Figure 2, select (*click*) the starting points (on the red curve(s)) which will be used to trace each colony.

* When selecting starting points: *Click* on one point for each colony. Choose a point along a vertical portion of the right side of each colony tracing. If two red curves trace a colony, select a point from the larger (or outer) of the two.*

12: *Two-finger click* on any of the N selected points and *click* “ *Export Cursor Data to Workspace...* ” and then exit out of the figures. *PD_boundary_0* ”.

13: Type “ *PD_boundary_0* ” .

14: Again, input “ [‘ ###.jpg ’] ” .

15: Type “ ‘ y ’ ” .

16: Type “ ‘ y ’ ” .

17: Enter the first frame number.

18: Enter the last frame number.

19: Enter the number of colonies.

* Ensure Figure 4 has accurate green boundaries for each colony. If any colony has not been successfully traced, restart from Step 7. Otherwise continue to Step 17. *

20: Type “ *PD_boundary_F* ” .

Done - The parameterizations $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ for all time points and colonies are automatically saved to a designated directory. * The scripts *PD_boundary_0* and *PD_boundary_F* were written by me specifically for this project. *

3.7.4 Model Training Routine

The following instructions describe the exact steps needed to obtain optimal parameters C^* and σ^* given input colony parameterizations $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$. The routine has been adjusted to fit curves $\gamma_{t_j}^i(\mathbf{x}, \mathbf{y})$ to $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ by utilizing the Simplex method



Figure 3.2: Sample output from the Boundary Tracing Routine. Though only three timesteps are shown in the *above* figure, the routine returns *green (light gray)* parameterizations $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ for all colonies, at all time points.

to find optimal model parameters C^* and σ^* .

Obtaining C^* and σ^* : Before proceeding, the uncompressed package ‘PDESOLVE_2016_UPDATE20’ as well as the training curves $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ and respective timelapse images must be loaded into the current directory. The .zip package includes the scripts ‘conCAT’, ‘DERIVATIVE’, ‘ERR’, ‘INIT’, ‘interppolygon’, ‘iswhole’, ‘Multicolony’, ‘Multicolonycheck’, ‘PRIMITIVE’, ‘REFINE’, ‘RESAMPLE’ and ‘SMOOTH’, as well as sample images and sample curves which can be used as an example. Additionally, the following routine is compatible with MATLAB 2020a (or newer) and requires MATLAB’s Image Processing Toolbox.

In the script ‘*Multicolony.m*’...

- 1: Input the cell-array data $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ as “ D ” (*line 3*).
- 2: Input the beginning and end frame numbers (*lines 8,9*).
- 3: Input the image directory, experiment date and colony details as “ Pic ” , “ date ” and “ SF ” , respectively (*lines 10-12*).
- 4: Input the desired polygon dimension used to interpolate colony parameterizations as “ NN ” (*line 14*).

* Appropriate values for NN range from 6 to 80. $NN = 20, 50$ were used in this study. *

In the MATLAB command window...

5: Type “ *close all ; clear all ; clc* ” .

6: Type “ $X = [0.5 , 45]$ ” .

* Here, X is the initial guess used for parameters C and σ . The value of X above was found to be a good-enough guess for all experiments conducted in this paper, however, this can be adjusted to any value. *

7: Type “ $X1 = fminsearch(@Multicolony , X)$ ” .

* This could take quite a while depending on the value of NN . *

8: Save $X1$ appropriately.

* At this point, optimal parameters $X = (C^*, \sigma^*)$ have been obtained. If timelapse imagery of the scripts is desired, proceed to Step 9, otherwise the routine is complete. *

In the script ‘ *Multicolonycheck.m* ’...

9: Repeat Steps 1-4, this time in ‘Multicolonycheck’.

In the MATLAB command window...

10: Type “ *Multicolonycheck(X1)* ” .

Done - The generated image sequence will automatically be saved to a previously specified directory.

3.8 Results

After a dozen failed imaging experiments, the imaging system was finally perfected. Two imaging sessions in particular yielded useful data (10 timelapse parameterizations in total) used in training and testing the nonlocal model. All timelapses are still frame image sequences (captured every 30 minutes) of *Paenibacil-*

lus dendritiformis initially inoculated as 2 μ L circular colonies. The timelapse figures in the following section show four time-points (or still images) from used timelapses.

Three timelapses' parameterizations were used to establish the time scaling used in all further simulations by minimizing the error between observed and simulated parameterizations. For this portion of the analysis, 20-dimensional ($NN = 20$) polygons were used to parameterize the data. This smaller value of NN was used to speed up the search for the, at this point, three optimal parameters K , C^* , and σ^* . Later, after the optimal time scaling has been obtained, NN will be increased to 50 for greater precision, and parameters of interest are reduced to only C^* and σ^* .

Within figure 3.3, the three timelapses' observed and optimized simulated parameterizations are displayed. Here, the yellow curves represent the obtained data $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$, and the red curves represent the optimized simulated boundaries $\gamma_{t_j}^i(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, for all bacterial colonies. The first and second column of images show two different single colony timelapses, with obtained $\varphi_{t_j}^1(\mathbf{x}, \mathbf{y})$ and optimized $\gamma_{t_j}^1(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$. After running the model optimization routine on timelapses' data, optimal model parameters are obtained as $(K, C^*, \sigma^*) = (3.12, 0.48, 16.97)$ and $(K, C^*, \sigma^*) = (3.07, 0.59, 13.55)$, respectively. Similarly, the third column of images shows a two colony timelapse as well as $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ and respective optimized $\gamma_{t_j}^i(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ for both colonies. Once again the error minimization routine is run on this data and the optimized model parameters were returned as $(K, C^*, \sigma^*) = (2.96, 0.54, 38.53)$. From these three timelapses, K was extremely consistent and led to a fixed time scaling of $K = 3$ for the nonlocal model.

Now that $K = 3$ has been found to be the optimal time scalar, more precise colony parameterizations, with $NN = 50$, are obtained for four more timelapses. Obtained colony parameterizations $\varphi_{t_j}^i(\mathbf{x}, \mathbf{y})$ are once again used to obtain nonlocal model parameterizations $\gamma_{t_j}^i(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, after finding optimal model parameters, this

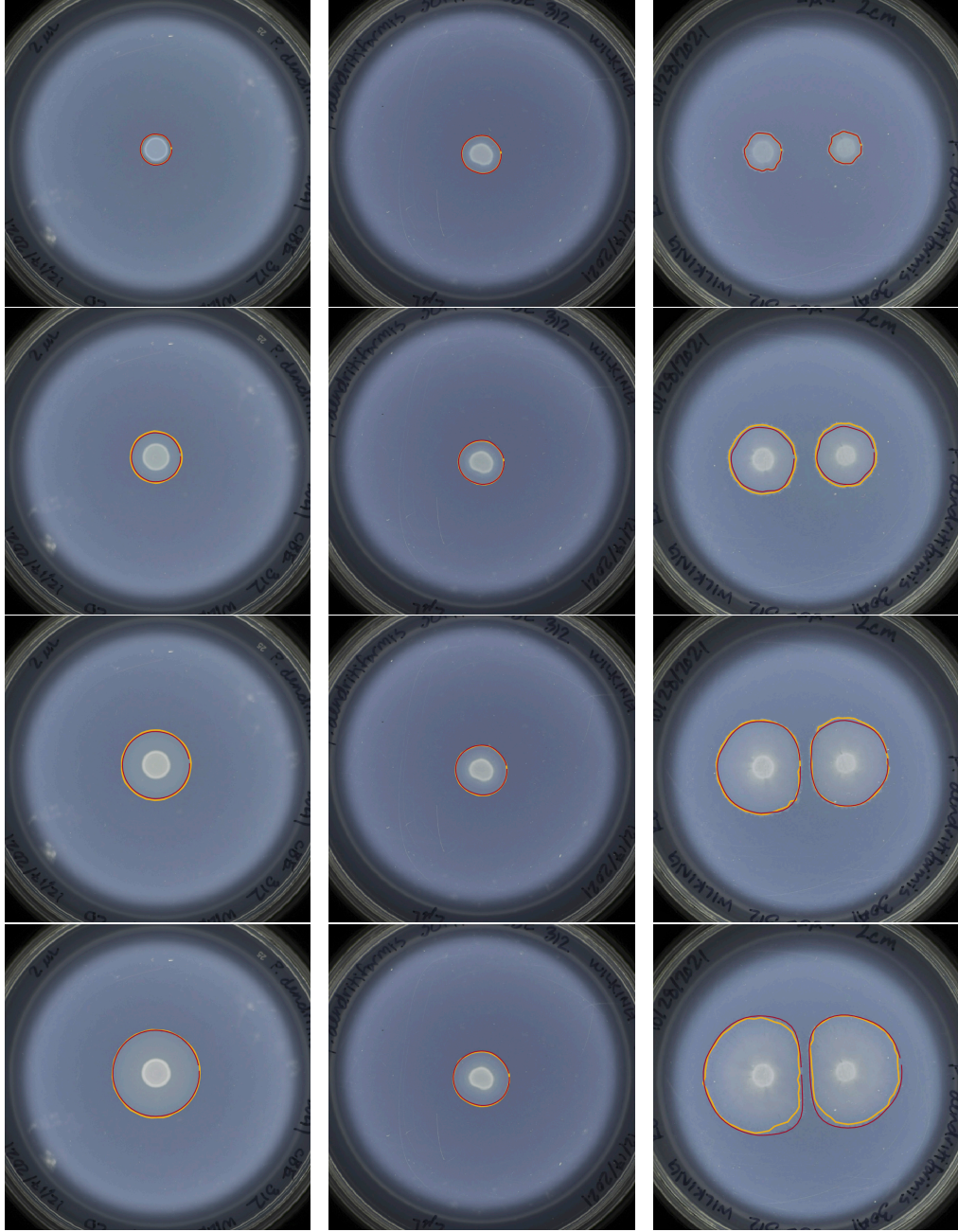


Figure 3.3: Timelapses of growing bacterial colonies, their obtained parameterizations $\varphi_{t_j}^i$ and their respective optimized nonlocal model boundaries $\gamma_{t_j}^i$. These timelapses were used to find time scalar K . The *yellow (light gray)* curves are obtained $\varphi_{t_j}^i$, the *red (dark gray)* curves are simulated $\gamma_{t_j}^i$ using optimal parameters (K, C^*, σ^*) . Timelapses are arranged as columns - *left*: one-colony timelapse ; *center*: one-colony timelapse ; *right*: two-colony timelapse with 2cm spacing.

time, C^* and σ^* . The observed colony parameterizations and the simulated model parameterizations from the four more precisely trained two-colony timelapses are displayed in figure 3.4 and figure 3.5.

The two columns in figure 3.4 show two two-colony timelapses with inoculation spacing of one and two centimeters, respectively. Optimal model parameterizations were obtained for the two-colony one centimeter experiment as $(C^*, \sigma^*) = (0.50, 24.24)$. Similarly, the optimal model parameters for the two-colony two centimeter timelapse were calculated to be $(C^*, \sigma^*) = (0.61, 40.37)$.

The two columns in figure 3.5 show two more two-colony timelapses, both with inoculation spacing of two centimeters. Optimal model parameterizations were obtained for the two two-colony experiments as values $(C^*, \sigma^*) = (0.5014, 31.5589)$ and $(C^*, \sigma^*) = (0.4793, 36.3407)$, respectively.

At this point in the study, seven timelapses have been successfully modeled through the error minimization routine developed in this study. Therefore, seven different pairs of optimal model parameters (C^*, σ^*) have been obtained for varying inoculation patterns. From the previous timelapses, model parameter data has been obtained for two one-colony timelapses, one two-colony timelapse with inoculation spacing of 1cm, and four two-colony timelapses with 2cm inoculation spacing. The goal for the remainder of this section is to accurately model additional colony growth timelapses by extrapolating from the data obtained for the seven trained simulations. Based on only the initial parameterizations (initial image frame) $\varphi_{t_0}^i$, the extrapolated parameters are used in the nonlocal model to generate simulated boundaries $\gamma_{t_j}^i$ of plated colonies, for full colony growth.

To extrapolate from known parameters, the parameters (C^*, σ^*) must be compared to the varying inoculation conditions for all obtained simulations. For the experiments conducted in this study, the number of colonies and the spacing between

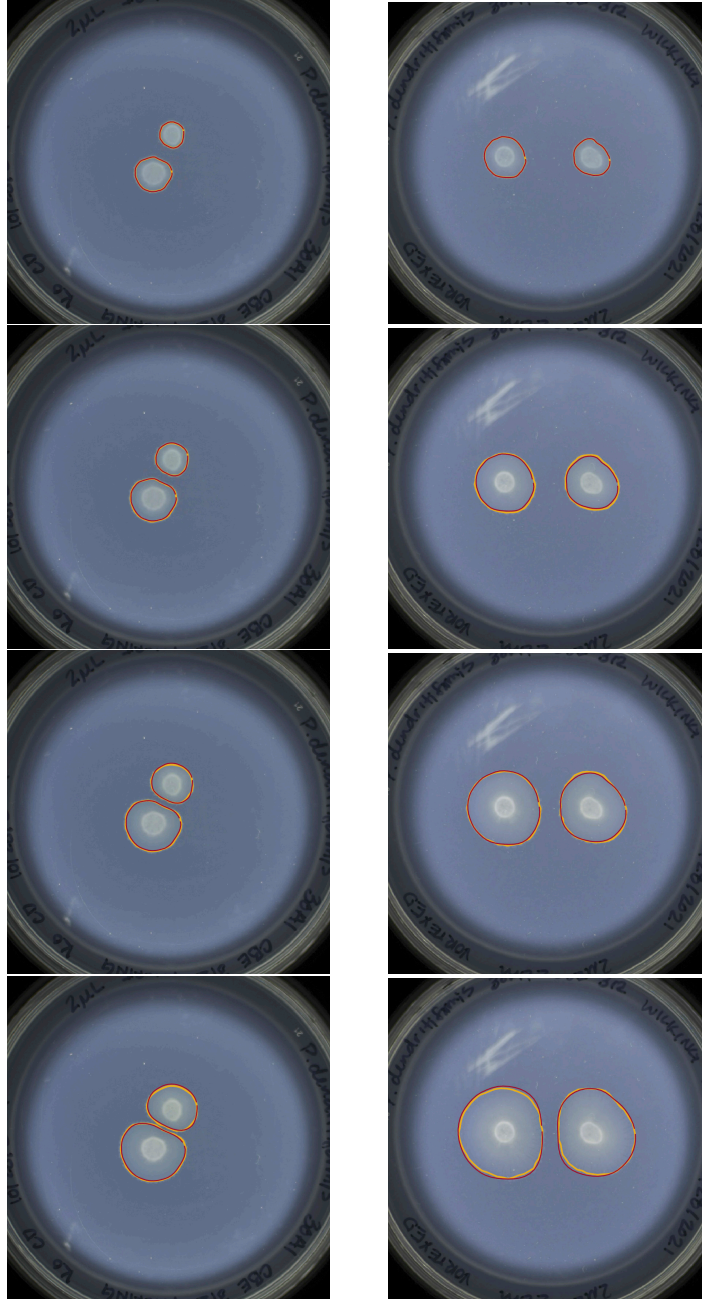


Figure 3.4: Timelapses of growing bacterial colonies, their obtained parameterizations $\varphi_{t_j}^i$ and their respective optimized nonlocal model boundaries $\gamma_{t_j}^i$. The *yellow* (*light gray*) curves are obtained $\varphi_{t_j}^i$, the *red* (*dark gray*) curves are simulated $\gamma_{t_j}^i$ using optimal parameters (C^*, σ^*) . Timelapses are arranged as columns - *left*: two-colony timelapse with 1cm inoculation spacing; *right*: two-colony timelapse with 2cm inoculation spacing.

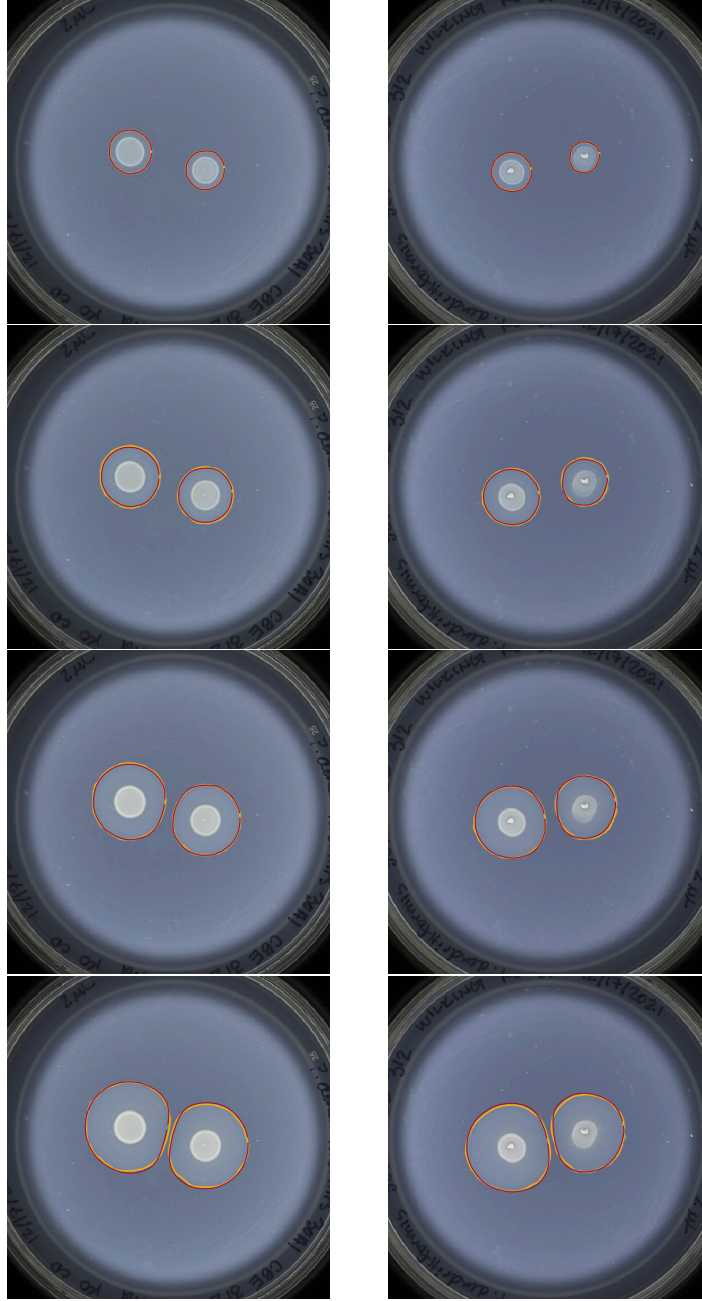


Figure 3.5: Timelapses of growing bacterial colonies, their obtained parameterizations $\varphi_{t_j}^i$ and their respective optimized nonlocal model boundaries $\gamma_{t_j}^i$. The *yellow (light gray)* curves are obtained $\varphi_{t_j}^i$, the *red (dark gray)* curves are simulated $\gamma_{t_j}^i$ using optimal parameters (C^*, σ^*) . Timelapses are arranged as columns - *left*: two-colony timelapse with 2cm inoculation spacing; *right*: additional two-colony timelapse with 2cm inoculation spacing.

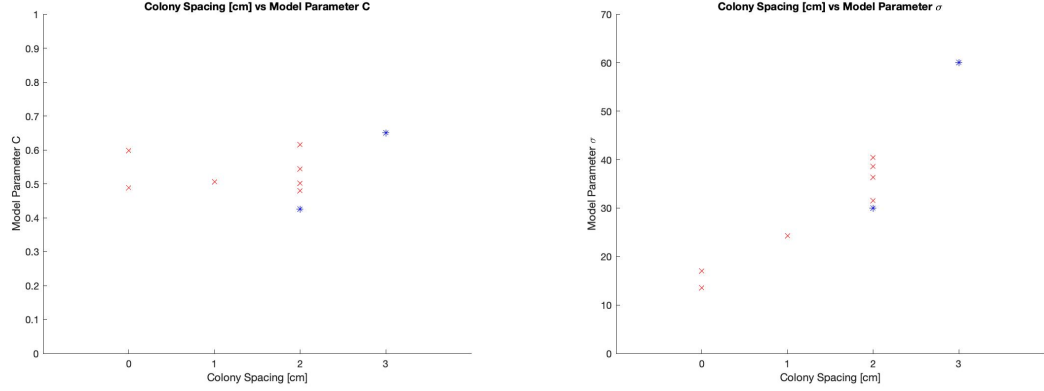


Figure 3.6: Parameters C (left) and σ (right) with respect to colony spacing. *Red (gray) 'x'*: learned (optimized) model parameters ; *Blue (dark gray) '*'*: extrapolated model parameters.

colonies are the only variables of consideration. However, as evidenced in [13], the model far exceeds these capabilities and is assumed to handle further variations of inoculation conditions, as well as imposed physical boundaries on the imaged plate itself.

In figure 3.6, seven pair of parameters, C^* and σ^* , obtained from the minimization routine, are plotted with respect to inoculated colony spacing as *red (gray) 'x'*. In these plots, single colony model parameters were plotted with colony spacing of 0. Similarly, in figure 3.7 the seven pair of parameters were plotted with respect to the number of colonies. Though there is extremely limited data to ‘learn’ from, extrapolated parameters were used to predict the qualitative growth of additional timelapses. Additionally, the best guesses for these extrapolated parameters have been plotted in both figure 3.6 and figure 3.7 as *blue (dark gray) '*'*.

Based on the qualitative trends present in figure 3.6, a series of eye-ball guesses were made for parameters used to describe an additional, two-colony three centimeter spaced timelapse. After a couple guesses, using extrapolated parameters $(C^E, \sigma^E) =$

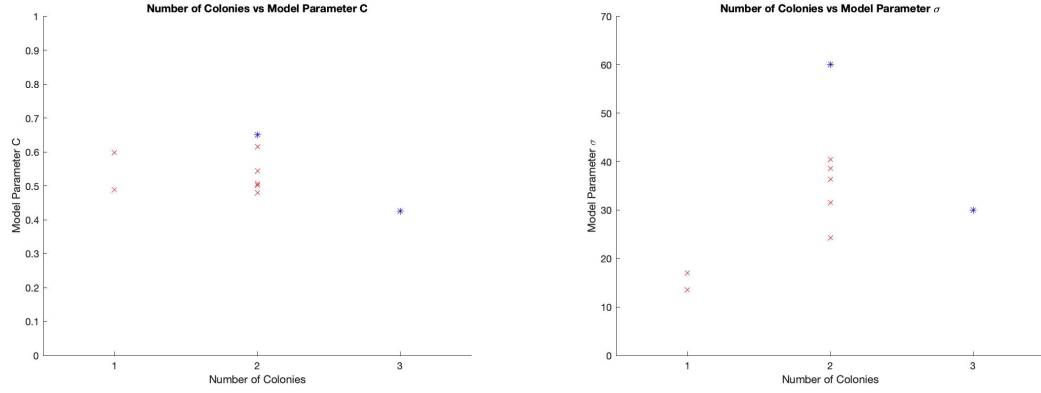


Figure 3.7: Parameters C (left) and σ (right) with respect to the number of colonies plated. As in previous, Red (gray) ‘x’: learned (optimized) model parameters ; Blue (dark gray) ‘*’: extrapolated model parameters.

(0.65, 60), the model appeared to accurately describe the qualitative growth of the new timelapse. The first column in figure 3.8 shows the initial *yellow* (light gray) obtained boundaries used in conjunction with extrapolated parameters, $(C^E, \sigma^E) = (0.65, 60)$, to generate the simulated *red* (dark gray) boundary parameterizations $\gamma_{t_j}^i$. For any reasonable initial inoculation pattern, a larger sample population for ‘trained’ timelapses would provide a much better prediction for model parameters (C, σ) .

Additionally, two three-colony two centimeter spaced timelapses were obtained within this same imaging period. These timelapses posed significant issues when attempting to accurately trace the bacterial boundaries (‘Boundary Tracing Routine’). The binary threshold used to distinguish colony from background was not uniform due to the lighting conditions present. Once the repulsion started in both of these timelapses, the ‘Boundary Tracing Routine’ failed at distinguishing between colonies - treating the three as one. Though accurate bacterial boundaries could not be obtained for all time points, the initial frame’s colonies were traced perfectly as $\varphi_{t_0}^i$. This time, extrapolated parameters are guessed based on the trends seen in figure 3.7.

After a couple of guesses, expected parameters $(C^E, \sigma^E) = (0.425, 30)$ proved to accurately describe the qualitative growth of the first three-colony 2 cm (evenly) spaced inoculation as seen in the central column of figure 3.8. The initial traced boundaries $\varphi_{t_0}^i$ are again plotted as *yellow (light gray)* curves. In conjunction with extrapolated (C^E, σ^E) , the initial tracings are used to obtain the simulated bacterial growth (*red (dark gray)* curves). After $(C^E, \sigma^E) = (0.425, 30)$ proved to accurately describe the qualitative growth of the first three-colony two centimeter spaced timelapse (central column of figure 3.8), the values were next tested on an additional timelapse with the same inoculation conditions. Once again, for this additional timelapse, the initial colony tracings $\varphi_{t_0}^i$ are obtained and used in conjunction with extrapolated model parameters $(C^E, \sigma^E) = (0.425, 30)$. The model again proves its capabilities in accurately capturing the bacterial growth and repulsion as seen in the *right* column of figure 3.8.

In this study, the imaging setup yielded only 10 usable timelapses. Though the sample population is extremely limited, the results obtained in this chapter show the methods conducted to be a viable means for training the nonlocal model. By minimizing the error between observed bacterial boundaries and the simulated boundaries (given the same initial frame) for multiple inoculation conditions, parameterization values are learned with respect to these inoculation conditions. Within the study, these conditions were limited to colony spacing and the number of colonies. Ideally, a much larger sample population of optimized timelapses could yield far more precise simulations. Additionally, statistical models could be used to extrapolate parameters to further inoculation conditions. Though extrapolations conducted in this paper took multiple guesses, the routines and methods conducted provide the framework needed in training such a nonlocal model with biological data.

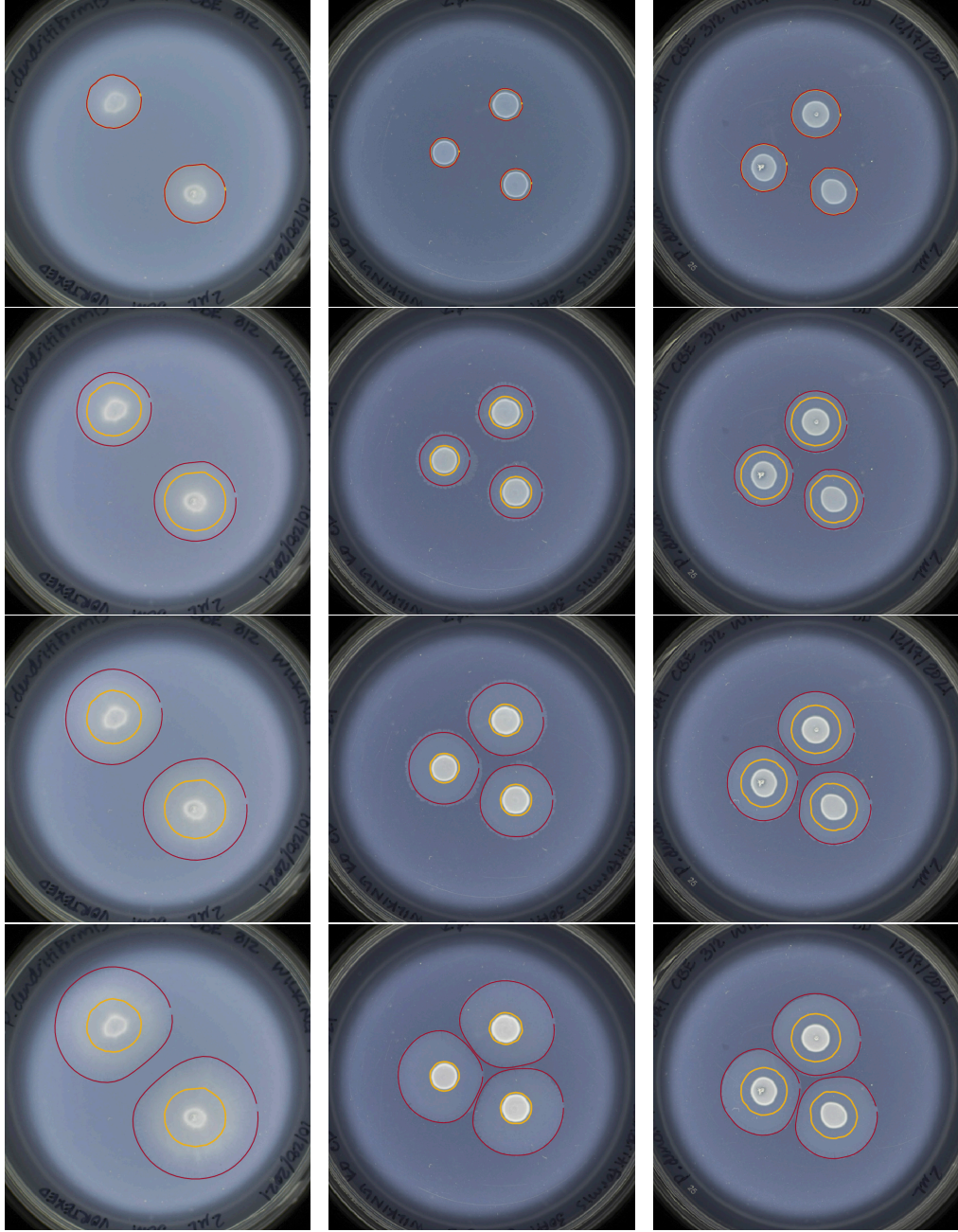


Figure 3.8: Initial colony (frame) tracings are obtained as *yellow (light gray)* $\varphi_{t_0}^i$ and used in conjunction with extrapolated parameters (C^E, σ^E) to obtain modeled *red (dark gray)* bacterial growth tracings, $\gamma_{t_j}^i$. Timelapses are arranged as columns - *left*: two-colony 3cm spaced timelapse using extrapolated $(C^E, \sigma^E) = (0.65, 60)$; *center*: three-colony 2cm spaced timelapse using extrapolated $(C^E, \sigma^E) = (0.425, 30)$; *right*: additional ‘test’ three-colony 2cm spaced timelapse using extrapolated $(C^E, \sigma^E) = (0.425, 30)$.

3.9 Conclusion

Through the development of a uniquely crafted imaging incubator, timelapse imagery was used to obtain boundary parameterizations. These tracings were then successfully used to ‘train’ a previously developed nonlocal model for colony front repulsion. By examining overall trends between inoculation patterns and optimized model parameter values, extrapolations were made to model additional qualitative imagery data. Future research involving robust statistical analysis and prediction of parameters would be extremely beneficial in extrapolating parameters for additional growth patterns. However, this study proves the methods conducted to be adequate in ‘training’ the numerical simulation [13] to approximate bacterial colony growth and repulsion as displayed by *Paenibacillus dendritiformis*.

CHAPTER FOUR

PARTICLE TRACKING AND MICRORHEOLOGY IN HUMAN GASTRIC
ORGANOIDS4.1 Introduction

Organoids are closed membrane mucosal sacs which are meant to represent an organ on a microscopic scale [16]. Organoids are believed to have a relatively homogeneous interior, primarily composed of mucus in addition to some dead-cell debris. Organoids, specifically those originating from gastric tissue, have been used to study the effects of bacterial infection on organoid tissue [2] and gastric cancer modeling [16]. However, little research has been conducted to observe the mechanical properties present within organoids. In this study, human gastric organoids are injected with fluorescent colloidal particles, imaged at various positions and time points, and then analysed to determine the viscoelastic modulus of the organoids' mucus. By examining the diffusive behavior—purely Brownian—of injected particles, the organoids' inner mucosal mechanical properties are extracted. The complex viscoelastic modulus—obtained from sampled particle tracks—reflect the fluid behavior mucus (within human gastric organoids) exhibits when perturbed at various frequencies.

Particle tracking is used to determine the positional coordinates for injected particles exhibiting Brownian motion. These particle tracks are then used to calculate mean squared displacements for each particle, which can then be used to quantify both the viscous and elastic, frequency-dependent moduli for the environment of embedding, mucus—a technique known as microrheology. This study is the first using microrheology to characterize and quantify the mechanical properties within an

organoid.

Though multiple particle tracking experiments were conducted for this study, the following chapter will provide the results from one specific experiment. In this experiment, around 70 organoids were grown and injected in preparation for imaging. However, only 8 organoids provided sufficient and viable tracking data. The imagery used for particle tracking purposes was obtained 24, 48, and 72 hours after the bead injections. At each of these time-points, 30 second videos were obtained at 6 ± 2 positions within each organoid. The experiment was set up to observe the heterogeneity existing within an organoid, as well as the variation between organoids. Multiple positions were imaged at multiple time-points to determine if any spatial or temporal trends exist. The results and procedure for this experiment will be outlined throughout subsequent sections.

4.2 Particle Tracking

Gastric glands are first isolated from a hospital patient’s gastric tissue, and are then seeded in Matrigel where they grow into human gastric organoids (HGOs). The HGOs are grown for several weeks until they reach a diameter of $300\mu\text{m}$, or larger, when they are deemed to be injectable. A selection of organoids were injected with $1\mu\text{m}$ fluorescent colloidal particles, diluted to a relative concentration of 10% (from the original volume). After all organoids were injected, a selection of the injected organoids were loaded onto glass slides in preparation for imaging. The organoid growth and imaging setup methodology are based on procedures from [15]. Confocal microscopy was next performed to image the injected human gastric organoids. It is important to note the organoids maintain a constant temperature of 310.15°K throughout their entire life—including while they are imaged on the confocal microscope.

The confocal scope was adjusted to $20\times$ objective zoom and an additional $4\times$ optical zoom which proved optimal for tracking purposes. An attempt was made to image a broader field of the organoid using a $10\times$ objective zoom. With this wide of frame, the $1\mu\text{m}$ fluorescent beads are only two to four pixels in diameter which is problematic when attempting to precisely track particles position. Similarly, an attempt was made to image particles with an objective zoom of $63\times$. Although the particle tracking was far more precise in this instance, only a few particles were viable for tracking in such a constrained domain. Objective zoom of $20\times$ proved sufficient in capturing the tracks of 15 ± 5 particles throughout a 30-second video.

The first image in figure 4.1 displays a brightfield view of a particular injected organoid, ‘Organoid 4’, in the study. The two smaller images were obtained using the confocal microscope with the optimal zoom $20\times$ objective, and additional $4\times$ optical zoom. The left image is the brightfield layer and the right is the fluorescent layer of a single region (position) imaged in ‘Organoid 4’ at the 24 hour time-point.

For a particle to be viable for tracking, it must be a singlet, and far enough away from others, so as to avoid hydrodynamic interactions. As long as particles were more than $3\mu\text{m}$ from another particle, a particles diffusion is assumed to be unaffected by hydrodynamic interactions. After injecting the colloidal beads into the organoid, some beads tend to disperse around the mucosal portion of the organoid, whereas others settle to the bottom. The particles which successfully embed themselves in the mucosal matrix sometimes aggregate, affecting their diffusion. For this reason, only singlet beads are tracked—doublets, triplets and larger aggregates are ignored.

After obtaining the 30 second fluorescent-field imagery, the videos were first converted from ‘.lif’ files to ‘.avi’ files and saved to the particle tracking directory. At this point, the particle tracking routine, detailed later and developed by [12], may commence. After the video is loaded into the ‘polyparticletracker’ guided user

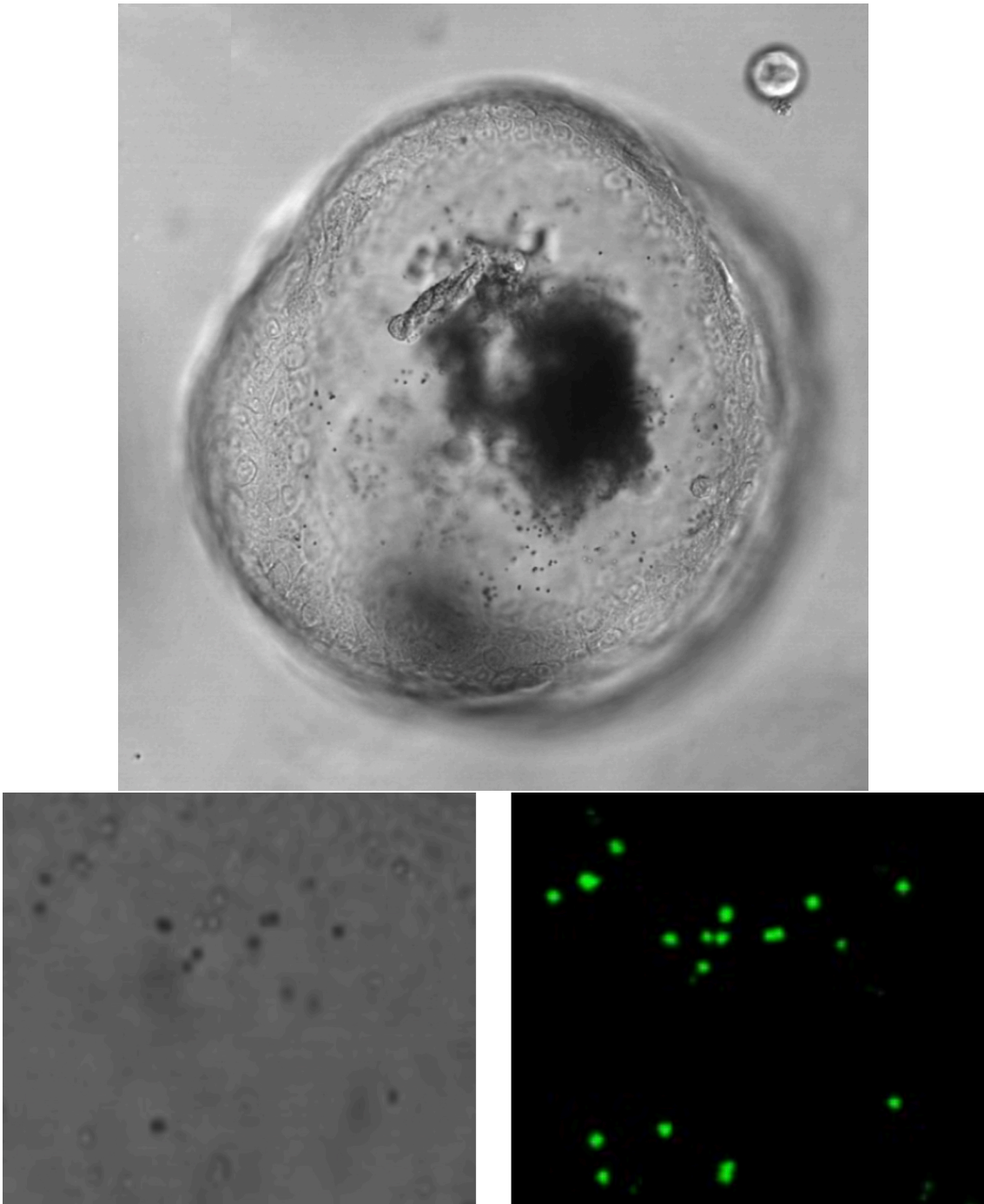


Figure 4.1: ‘Organoid 4’ injected with $1\mu\text{m}$ fluorescent beads - *top*: Large scale view of ‘Organoid 4’ captured in bright-field ; *bottom left*: Tracking scale ($20\times$ objective, additional $4\times$ optical zoom) bright-field view of ‘Organoid 4’ at particle tracking position 3 of 8 (24hr timepoint) ; *bottom right*: Corresponding fluorescent-field image (same region and timepoint as *Bottom left*).

interface, the video is examined to ensure the video provides sufficient trackable events. The region of interest is constrained if the trackable events occur in a specific subregion of the video.

Many parameters must be adjusted to capture the particles of interest for this study. First, a Gaussian filter with standard deviation of 2 pixels was applied to smoothen the appearance of the particles. Next, the minimum radius was set to 3 pixels and the maximum was set to 7 pixels. Additionally, the eccentricity threshold was adjusted to 0.65, which, along with radius bounds, allows the program to only track singlet beads and ignore any aggregates. A minimum track length requirement was added and set to 250 frames. Some particles dip in and out of the visible field and those tracks are not of interest—this parameter requires tracks be longer than 250 frames in order for the track to be recorded in the output data. The parameter ‘Search for new particles in N-frames’ is adjusted to 40, allowing the program to only search for new trackable particles every 40 frames. Lastly, the type of calculation used to approximate the center of each particle is set as ‘polynomial fit gaussian weight’. This essentially fits a two dimensional Gaussian curve over each particle (bright spot in the fluorescent image) and records the peak of the signal.

Particle tracks are stored as a single cell array at the end of the routine. Each of the entries is a particular particle’s track as a two column, n row matrix where the i^{th} row is the i^{th} $\mathbf{x} = (x, y)$ position for a particle’s track. In the following section, these particle tracks will be used to obtain mean squared displacement data and subsequently the viscoelastic properties of the organoid’s mucus. Though only colloidal particles are tracked throughout this study, the ‘polyparticletracker’ can also be used to track motile bacteria.

The fluorescent image on the left of figure 4.2 shows a particular position imaged within ‘Organoid 4’. The arrows point to six different $1\mu\text{m}$ fluorescent colloidal beads

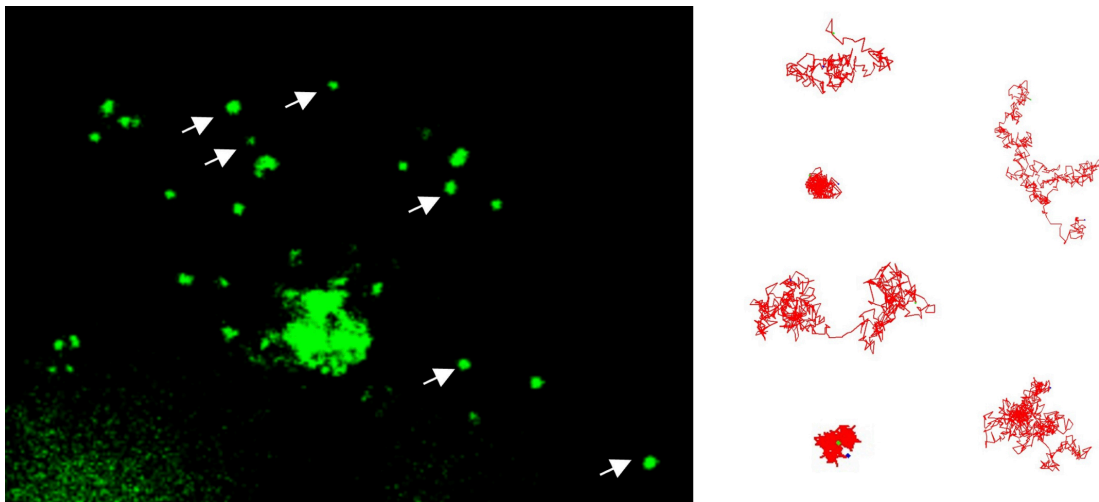


Figure 4.2: Example of beads tracked in ‘Organoid 4’ (position 3/8, 24hr timepoint) (*left*) and their respective plotted tracks (*right*). The tracks display two distinct behaviors: caged diffusion and free diffusion.

whose tracks are displayed in the right image of figure 4.2. These tracks display two behaviors, Brownian motion with elastic caging—where particles’ motion is restricted to an extremely local environment, as if they are bouncing around within a cage—and less restricted Brownian motion. One bead even exhibits this caging behavior for a while, before moving out of the first cage and into a second for the remainder of its tracked period. It is important to note the distinction between doublets, triplets, larger aggregates and the singlets of interest, as seen in the left image of figure 4.2. The singlets which are not labeled in the figure reflect particles which did not meet the particle tracking criteria.

4.3 Microrheology Experiments

Once the particle tracks are obtained, they are used to calculate the particle’s mean squared displacement (MSD) over time. The MSD is calculated as an ensemble

average using the equation:

$$\text{MSD}(t) = \langle |\mathbf{x}(\mathbf{t}) - \mathbf{x}_0|^2 \rangle = \frac{1}{N} \sum_{i=1}^N |\mathbf{x}^i(t) - \mathbf{x}^i(0)|^2. \quad (4.1)$$

where N is the total number of particles to be averaged, $\mathbf{x}_0^i$ is the reference (initial) position for the i^{th} particle and $\mathbf{x}^i(\mathbf{t})$ is the i^{th} particle position at time t .

The time dependent MSDs are then fit with an exponential model, where α represents the exponential slope of the MSD profile. α values provide a good measure for a particle's diffusive behavior. Particles which experience significant caging, or are diffusing through an elastic environment, will have a lower α value, whereas freely diffusing particles have an α value ≈ 1.0 . A particle which is fit with $\alpha > 1.0$ is considered to be exhibiting super-diffusive behavior. In [11], authors performed similar analysis and determined α values to be reliable for $0 < \alpha < 2.0$. The exponential model is formulated as follows:

$$\text{MSD}(t) \approx \beta t^\alpha. \quad (4.2)$$

At this point in the study, a series of control experiments were conducted to ensure the particle tracking routine was accurate and robust. $1\mu\text{m}$ fluorescent beads were injected into phosphate buffered saline (PBS - pH ≈ 7.4 buffer), as well as 15mg/mL and 30mg/mL concentrations of human gastric mucus. Particle tracking was conducted for all videos obtained from the three sample control media. Figure 4.3 reflects the ensemble time dependent MSD profiles for beads injected into the three media, and imaged with two different confocal zoom settings: 20 $\times$ objective with additional 4 $\times$ optical zoom and 10 $\times$ objective with an additional 4 $\times$ optical zoom. Though the 10 $\times$ -zoom MSD profiles are displayed in this plot, only the *bold* 4 $\times$ -zoom

MSD profiles were used as means for control.

The control MSD profiles were then fit with the exponential model (4.2), providing individual and ensemble α values for beads in each medium. PBS and HGM are primarily made of water, and therefore the beads should exhibit freely diffusive behavior when embedded in such media. From [11], purely viscous particle diffusion (free diffusion) corresponds to $\alpha \approx 1.0$, and purely elastic particle diffusion corresponds to $\alpha \approx 0.00$. Ensemble particle MSDs reveal values: $\alpha = 0.971(\pm 0.083)$ for PBS, $\alpha = 0.946(\pm 0.135)$ for 15mg/mL HGM, and $\alpha = 1.034(\pm 0.207)$ for 30mg/mL HGM. These values are extremely close to 1.0 (1.0 ± 0.2) and prove the tracking routine is viable in collecting positional information. An additional test experiment was conducted to further validate the routine’s capabilities. A sample video of beads in water was obtained from the authors of [19]. The beads in water once again proved the routine capable—with calculated $\alpha \approx 1.0$.

Mean squared displacements for all particles tracked are next run through the microrheology routine, where viscoelastic material properties are extracted. Though human gastric organoids appear, at first, to be relatively homogeneous (as seen in figure 4.1), no study (that could be found in documentation) has performed microrheology experiments within organoids to determine the viscoelastic properties within an organoid. The following section outlines the approach taken to extract frequency dependent viscoelastic properties within human gastric organoids. The routine was developed in accordance with the microrheology experiments conducted in [12] and [11]. Scripts obtained directly from the authors of [19] were adapted to perform the following analysis. MSDs are used to calculate the local viscoelastic spectrum $\tilde{G}(s)$ where s is the Laplace frequency. Subsequently, $\tilde{G}(s)$ was used to calculate the complex storage loss modulus $G^*(\omega) = G'(\omega) + iG''(\omega)$. Here, $G'(\omega)$ represents the storage (elastic) modulus, $G''(\omega)$ represents the loss (viscous) modulus,

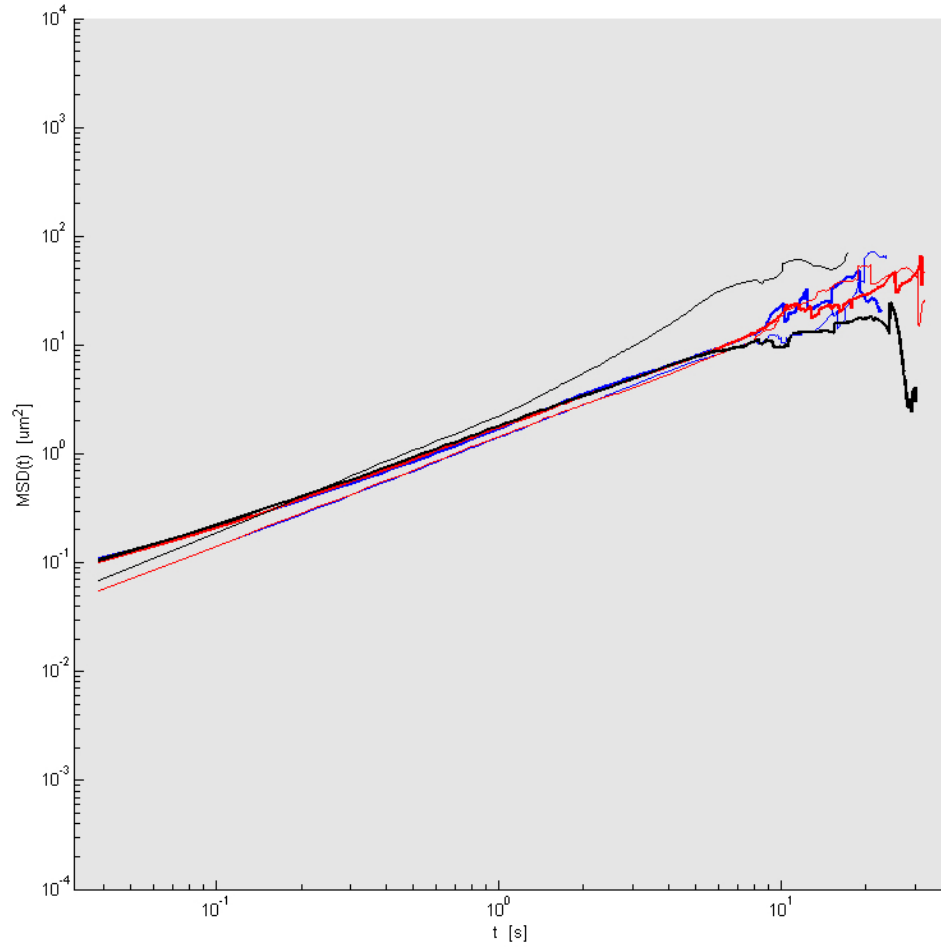


Figure 4.3: Time dependent ensemble MSD profiles of $1\mu\text{m}$ fluorescent beads diffusing through different control media. The *bold* profiles represent data obtained with the desired imaging conditions ($20\times$ objective, and additional $4\times$ optical zoom). Control MSDs in various media - *blue*: phosphate buffered saline (PBS) ; *black*: human gastric mucus (HGM) at 15mg/mL concentration ; *red*: human gastric mucus (HGM) at 30mg/mL concentration.

and frequency $\omega = \frac{2\pi}{t}$.

Assuming the local viscoelastic modulus around each particle is near the bulk viscoelastic modulus present within the organoid, then $\tilde{G}(s)$ can be calculated from MSD(t) [12]. This is a major assumption for this study as it is later determined local viscoelastic moduli to be quite heterogeneous throughout an organoid. However, given this assumption $\tilde{G}(s)$ is calculated from the Laplace transform of MSD(t) (denoted $\langle \Delta \tilde{r}^2(s) \rangle$) and using the generalized Stokes–Einstein equation (GSE) as proposed in [11]:

$$\tilde{G}(s) = \frac{k_B T}{\pi R s \langle \Delta \tilde{r}^2(s) \rangle}, \quad (4.3)$$

where k_B is Boltzmann’s constant, T is temperature (°K), R is the particle’s radius (μm)(in our study, $0.5\mu\text{m}$), and s is the Laplace frequency.

From [11], we know $\tilde{G}(\omega) \approx G^*(s)$ by letting $G^*(\omega) = \tilde{G}(s = i\omega)$, where $G^*(\omega)$ is an equivalent formulation of the GSE and $\omega = \frac{2\pi}{t}$ is standard frequency. Now, the complex storage loss modulus $G^*(\omega) = G'(\omega) + iG''(\omega)$ has been obtained as:

$$G^*(\omega) = G'(\omega) + iG''(\omega), \quad (4.4)$$

where $G'(\omega)$ represents the storage (elastic) modulus, $G''(\omega)$ represents the loss (viscous) modulus. At this point, the frequency dependent elastic and viscous moduli are graphed as in the right plot in figure 4.4. The left image in figure 4.4 shows individual and ensemble bead MSDs, with respect to time, which were used to calculate the ensemble frequency dependent complex storage-loss modulus (right plot in figure 4.4). It is important to note $G^*(\omega)$ cannot be trusted at the frequency extremes, as discussed in [11]. However, $G^*(\omega)$ is trusted for $\omega \in (10^{-1}, 10^1)$ as discussed in [11].

Numerically, the code provided from authors of [19] performs one interesting

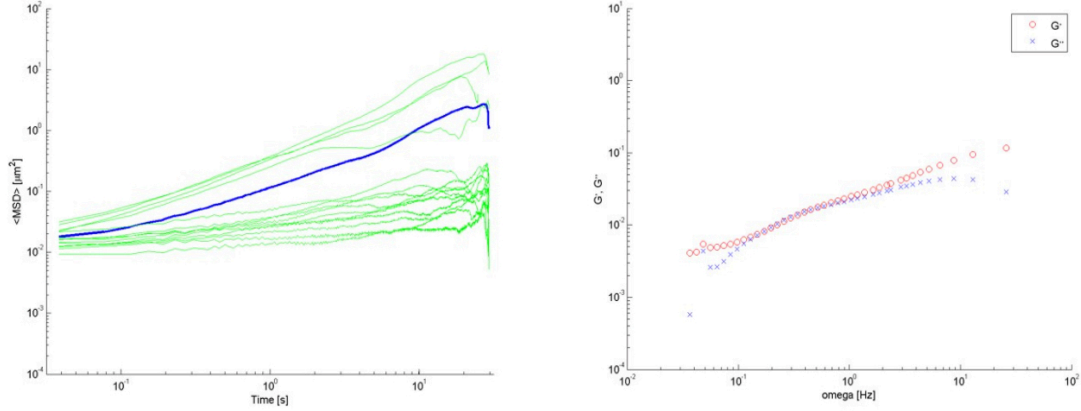


Figure 4.4: MSDs and resultant ensemble complex modulus from ‘Organoid 4’ 24hr after injection - *left*: individual (*green (light gray)*) and ensemble (*blue (dark gray)*) MSD profiles of all beads tracked ; *right*: the resultant elastic (*red (gray circles)*) and viscous (*blue (dark gray x’s)*) moduli.

calculation: calculating the numerical Laplace transform of $\text{MSD}(t)$ as $\langle \Delta \tilde{r}^2(s) \rangle$. The details for numerical methods used to ultimately obtain $G^*(\omega)$ are in [12].

4.4 Methods

4.4.1 Particle Tracking Routine

The following instructions detail the routine used to track particles’ movement throughout a video (*.avi* video, or *.tiff* image sequence). The imagery used for particle tracking in this study consists of 30 second videos. The videos were imaged using a confocal microscope with objective zoom of $20\times$ and an additional optical zoom of $4\times$. The following instructions use functions, and are adapted, from [14].

Obtaining Particle Tracks:

Before proceeding, the uncompressed package ‘polyparticletracker’, as well as the *.avi* file of interest (or, *.tiff* image sequence) must be loaded into the current

directory. The .zip package includes the scripts ‘polyfitweighted2.m’, ‘polyparticle-tracker.m’, ‘polyparticletracker_gui.fig’, ‘polyparticletracker_gui.m’ and ‘polyval2.m’. Additionally, the following routine is compatible with Matlab 2009b (or earlier) and will not work for more current versions of Matlab, without adaptation.

In the Matlab command window...

- 1: Type “ *data = polyparticletracker_gui* ” .

In the GUI...

- 2: Input the parameter ‘ *Filename* ’ as “ *example.avi* ” .
 * Following each input below, values which were used to analyse the videos in this study will be displayed as (*V*) . However, these values should be adjusted to track whichever particles are of interest. *
- 3: Input the parameter ‘ *Noise length* ’ as value (*2*) .
- 4: Input the parameter ‘ *Eccentricity threshold* ’ as value (*0.65*) .
- 5: Input the parameters ‘ *Maximum radius* ’ and ‘ *Minimum radius* ’ as values (*7*) and (*3*) , respectively.
- 6: Input the parameter ‘ *Minimum track length (frames)* ’ as value (*240*) .
- 7: Input the parameter ‘ *Search new particles in N* ’ as value (*40*) .
- 8: Select ‘ *Polynomial Fit Gaussian Weight* ’ (bottom right of GUI). * Here, changing the region of interest (by selecting ‘ *Change ROI* ’) can allow for faster processing and may be useful when tuning the previous parameters. *
- 9: Click on the ‘ *Start!* ’ command.
- 10: When the program no longer says ‘ *Running!* ’ in place of ‘ *Start!* ’ , select ‘ *Exit* ’ .

In the Matlab command window...

- 11: Type “ *data = data.tr* ” .
- 12: Save the particle track list by *two-finger clicking* on the variable ‘ *data* ’ and

clicking ‘ Save As... ’ .

Done - The saved ‘ *data* ’ is a cell-array with each entry a matrix containing (x, y) coordinate positions of a particular particle’s track.

4.4.2 Microrheology Routine

The following routine details the instructions needed to obtain ensemble and individual mean squared displacements (MSDs) of all particle tracks (and respective x). The MSDs are fit to an exponential model—where exponent α reflects the bead’s (bead ensemble’s) type of diffusivity—and are also used to calculate ensemble and individual frequency dependent viscoelastic modulus $G = G' + iG''$ (and respective ω) of all particle tracks. The script ‘ microrheologyanalysis_01_01_22.m ’ also calculates the ensemble and individual $G'(1\frac{rad}{s})$, $G''(1\frac{rad}{s})$, $\tan\delta(1\frac{rad}{s})$. The functions and Matlab scripts used in the following routine were adapted from scripts used in [19]. (G' is the elastic component and G'' is the viscous component)

Obtaining Microrheology Calculations:

Before proceeding, the uncompressed package ‘microrheologyanalysis_01_01_22’, as well as the saved particle tracking data must be loaded into the current directory. The .zip package includes the scripts ‘calc_G.m’, ‘logderive.m’, ‘Mean_SD_many_single_beads_E.m’, ‘Mean_SD_many_single_beads.m’, ‘making_logarithmically_spaced_msd_vs_tau.m’, ‘microrheologyanalysis.m’, ‘polyfitw.m’, ‘space125.m’ as well as example data ‘water.mat’. Additionally, the following routine is compatible with Matlab 2009b (or newer).

In the script ‘ *microrheologyanalysis_01_01_22.m* ’...

- 1: Input the actual pixel size (μm) (*line 7*).
- 2: Input the frame rate (f/s) (*line 12*).
- 3: Input the radius of the particles which were tracked (μm) (*line 24*).
- 4: Input the temperature of the environment when the particles were imaged ($^{\circ}\text{K}$)

(*line 26*).

In the Matlab command window...

5: Type “ *microrheologyanalysis_01_01_22* ” .

6: Type the filename for the tracking data as “ ‘*example_tracks_data.mat*’ ” .

4.5 Results

As previously stated, 8 organoids were injected, imaged and tracked. Previously, it was assumed organoids’ mechanical properties are relatively homogeneous, however analysis in this study will reveal this is not completely true. The following section will discuss the results obtained from executing the particle tracking and microrheology routines for varying locations and time-points throughout the organoid. The following analysis was obtained from running the particle tracking and microrheology routines on 30-second 2D confocal fluorescent field videos with 15 ± 5 trackable beads per video. There were 6 ± 2 positions imaged within each organoid, at 24, 48 and 72hr after the bead injection. Though temporal and spatial variation were included in this experiment, no viable spatial or temporal trends were established following the analysis.

First, α values were obtained from every trackable bead injected into an organoid. The α values were extremely variable, but centered around $\alpha \approx 0.38$ with a range $\alpha \in (0.01, 1.00)$ and standard deviation $\sigma = 0.21$. Given such a large standard deviation, it is extremely difficult to characterize the entire population of beads’ diffusive behavior. Though the ensemble value between all organoids is $\alpha = 0.38$, every particle does not diffuse in the same elastic manner as the ensemble value would suggest. Some organoids even had some beads which were elastically caged and other beads which freely diffused. These two groupings of behavior are evident in the MSD plot on the left of figure 4.4. From this figure, it is quite evident there

exists a bimodality of particle's diffusive behavior within 'Organoid 4' at the 24hr time-point. Particles in additional organoids and time-points displayed these two diffusive behaviors, whereas all particles in some organoids proved to diffuse elastically and appear to be caged. Additional statistical analysis may reveal some spatial and temporal variation for α values. However, this study reveals an immense heterogeneity between individual particles' α values in a particular organoid and ensemble α values between organoids, positions and time-points after injection.

After α values were calculated for all particles in all organoids, numerical methods for solving (4.3) are used to calculate the frequency dependent complex storage-loss modulus $G^*(\omega) = G'(\omega) + iG''(\omega)$, for all particles in all organoids, at all positions and time-points. The ensemble frequency dependent storage-loss modulus was plotted for each organoid at each time-point as in the right plot of figure 4.4. Here, the red circles represent the elastic component $G'(\omega)$ and the blue x's represent the viscous component $G''(\omega)$ at a particular frequency. Here, ω values represent the frequency at which the environment (surveyed by colloidal particles) are perturbed. Given a particular perturbation frequency, $G'(\omega)$ and $G''(\omega)$ values reveal the organoids viscoelastic properties. For the right plot in figure 4.1, when the elastic component is greater than the viscous component, the organoid's mucus deforms elastically. Similarly, when the viscous component is greater than the elastic component, the organoid's mucus deforms viscously, allowing for greater particle diffusion.

For the particular organoid plotted in figure 4.4 ('Organoid' 4 at 24hr time-point), microrheology of the bead ensemble reveals both elastic or viscous behavior for different perturbation frequencies. For the frequency range $\omega \in (10^{-0.9}, 10^{-4})$, the mucosal environment in 'Organoid 4' (at the 24hr time-point) appears to deform viscously. For all other frequencies, the mucosal environment appears to

deform elastically. At all time-points, the ensemble MSD profile of ‘Organoid 4’ reveals viscous behavior in some frequency regime. When the same analysis was performed on all other organoids, the viscous component $G''(\omega)$ never prevailed at any perturbation frequency. Instead, all other organoids, at all time-points, appear to deform elastically. That being said, there are some particles in each organoid which diffused freely throughout their environment; though the ensemble of particles showed elastic behavior. Though viscosity and elasticity is obtained for all frequency values, the following analysis will focus on the calculated elasticity and viscosity at $\omega = 1 \frac{rad}{s}$, $G'(\omega = 1 \frac{rad}{s})$ and $G''(\omega = 1 \frac{rad}{s})$, respectively.

The storage and loss component values at $\omega = 1$ provide an estimate for the standard elasticity and viscosity of the mucosal environment within an organoid. The ensemble MSD for particles in a particular video (position) were used to calculate $G'(\omega = 1 \frac{rad}{s})$ and $G''(\omega = 1 \frac{rad}{s})$ for all organoids at all time-points. Ensemble $G'(1 \frac{rad}{s})$ and $G''(1 \frac{rad}{s})$ data for all positions (videos) tracked (in all organoids, at all timepoints) is plotted as the first two plots in figure 4.5. Additionally, $\tan\delta(1 \frac{rad}{s}) = G''(1 \frac{rad}{s})/G'(1 \frac{rad}{s})$ was calculated and plotted as the bottom plot in figure 4.5. $\tan\delta(1 \frac{rad}{s})$ is of particular interest as it reveals if the mucosal environment is behaving viscously or elastically. In the $\tan\delta(1 \frac{rad}{s})$ plot of figure 4.5, for points below the *dashed* line, elasticity prevails. Similarly, for points above the *dashed* line, viscosity prevails. In all plots of figure 4.5, each *red* point is the ensemble average of beads tracked in one 30-second video for a particular organoid (‘Organoids 1–9’ are organized left to right at each time-point; There are only three discrete time-points). In this figure, each *black* point represents ensemble averages calculated from all beads tracked in all organoids at one particular time point.

As evident in the $\tan\delta(1 \frac{rad}{s})$ plot of figure 4.5, the majority of particle ensembles for different organoid positions reveal elastic behavior. Some *red* points are above

the *dashed* line, revealing some positions as viscous regions based on the ensemble behavior of particles within this position (video). Statistical analysis has not been utilized to establish any temporal or spatial trends of $G'(1\frac{rad}{s})$, $G''(1\frac{rad}{s})$ or $\tan\delta(1\frac{rad}{s})$. However, the analysis conducted in this study reveals immense heterogeneity of the mucosal mechanical properties present between organoid positions, and at all scales. Additionally, the microrheology analysis performed reveals heterogeneity in the general population of organoids sampled. Further analysis, and larger sample populations may reveal some trends of the mechanical properties of human gastric organoid's interior structure. However, this study is concluded with the observation that organoids are not as homogeneous as they are believed. This study reveals much greater heterogeneity and complexity of human gastric organoids than was previously assumed.

4.6 Conclusion

This study concludes with microrheology analysis as an acceptable means for sampling extremely local mucosal environments in human gastric organoids. Particle tracking and microrheology analysis of these organoids reveal immense heterogeneity throughout an organoid, spatially, and among organoid populations. Though further statistical analysis may reveal some spatial or temporal trends for the mucosal mechanical properties present within human gastric organoids, this study has concluded with an obvious viscoelastic diversity present within and between HGOs. By injecting colloidal particles within a small population of HGOs, the mechanical properties within the organoids were sampled in various regions, through means of particle tracking and microrheology. This study is the first of its kind and provides the framework for future research utilising microrheology to analyze organoids' viscoelastic characteristics.

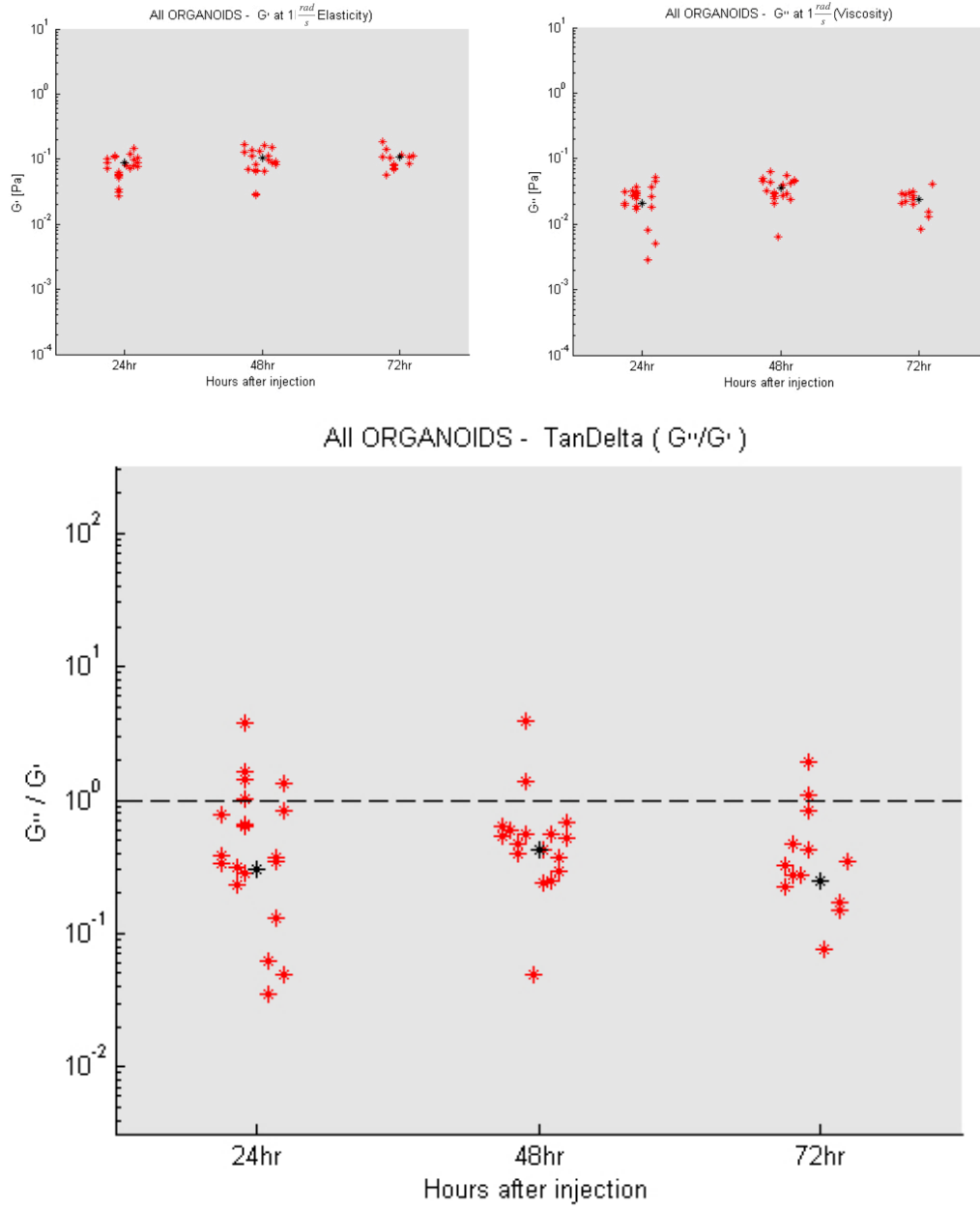


Figure 4.5: Calculated $G'(1 \frac{rad}{s})$, $G''(1 \frac{rad}{s})$ and $\tan\delta(1 \frac{rad}{s})$ values for all videos in all organoids at all timepoints following injection. In all plots, *red (light gray)* points represent ensemble values for one particular position imaged in one organoid. The *black* points represent all *red* points averaged at one timepoint after injection. *Top left:* $G'(1 \frac{rad}{s})$ (elasticity) ; *top right:* $G''(1 \frac{rad}{s})$ (viscosity) ; *bottom:* $\tan\delta(1 \frac{rad}{s})$ ($G''(1 \frac{rad}{s})/G'(1 \frac{rad}{s})$) values.

CHAPTER FIVE

CONCLUSION

Within this thesis, nonlocal models were first examined and studied as methods developed with aims in biological applications. The Cucker–Smale flocking model kicked off a stream of nonlocal models constructed to simulate various biological behaviors. Nonlocal models for Lévy flights in crime, aggregation and swarming models are next examined so as to provide background in the importance of nonlocality in biology. In Chapter 3, a very different type of nonlocal model is examined and utilized in analysing and predicting the behavior of competitive-inhibitory repulsive bacterial colony fronts. Image processing and analysis techniques were used to track boundary dynamics and to obtain optimal model parameters for use in bacterial simulation.

In Chapter 4, further image analysis techniques were used to track particles exhibiting Brownian motion throughout human gastric organoids. Microrheology techniques were used to extract viscoelastic properties present within organoids’ closed-cell membranes. Analysis of the viscoelastic moduli reveals an extremely heterogeneous mucosal structure which tends to behave elastically in the majority of local mucosal environments sampled. This thesis is concluded with a further understanding in utilizing imagery data to obtain observables of biologic systems. In two instances, obtained imagery data was used to make qualitative and quantitative conclusions of biological behavior present within our experimental systems—repulsive bacterial colonies and viscoelastic mucus within human gastric organoids.

REFERENCES CITED

- [1] Rama Bansil, Joseph Hardcastle, and Maira Constantino, *Microrheology of mucin: tracking particles and helicobacter pylori bacteria*, Épitöanyag (Online) (2015), no. 4, 150.
- [2] Sina Bartfeld, Tülay Bayram, Marc van de Wetering, Meritxell Huch, Harry Begthel, Pekka Kujala, Robert Vries, Peter J Peters, and Hans Clevers, *In vitro expansion of human gastric epithelial stem cells and their responses to bacterial infection*, Gastroenterology **148** (2015), no. 1, 126–136.
- [3] Avraham Be’Er, HP Zhang, E-L Florin, Shelley M Payne, Eshel Ben-Jacob, and Harry L Swinney, *Deadly competition between sibling bacterial colonies*, Proceedings of the National Academy of Sciences **106** (2009), no. 2, 428–433.
- [4] Andrea L Bertozzi, Theodore Kolokolnikov, Hui Sun, David Uminsky, and James Von Brecht, *Ring patterns and their bifurcations in a nonlocal model of biological swarms*, Communications in Mathematical Sciences **13** (2015), no. 4, 955–985.
- [5] Avraham Be’er, Gil Ariel, Oren Kalisman, Yael Helman, Alexandra Sirotamadi, HP Zhang, E-L Florin, Shelley M Payne, Eshel Ben-Jacob, and Harry L Swinney, *Lethal protein produced in response to competition between sibling bacterial colonies*, Proceedings of the National Academy of Sciences **107** (2010), no. 14, 6258–6263.
- [6] Sorathan Chaturapruek, Jonah Breslau, Daniel Yazdi, Theodore Kolokolnikov, and Scott G McCalla, *Crime modeling with Lévy flights*, SIAM Journal on Applied Mathematics **73** (2013), no. 4, 1703–1720.
- [7] Felipe Cucker and Steve Smale, *Emergent behavior in flocks*, IEEE Transactions on Automatic Control **52** (2007), no. 5, 852–862.
- [8] Theodore Kolokolnikov, Hui Sun, David Uminsky, and Andrea L Bertozzi, *Stability of ring patterns arising from two-dimensional particle interactions*, Physical Review E **84** (2011), no. 1, 015203.
- [9] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright, *Convergence properties of the Nelder–Mead simplex method in low dimensions*, SIAM Journal on Optimization **9** (1998), no. 1, 112–147.
- [10] Andrew J Majda and Andrea L Bertozzi, *Vorticity and incompressible flow*, vol. 27, Cambridge University Press, 2002.
- [11] Thomas G Mason, *Estimating the viscoelastic moduli of complex fluids using the generalized Stokes–Einstein equation*, Rheologica acta **39** (2000), no. 4, 371–378.
- [12] Thomas G Mason, Kavita Ganesan, J Harry van Zanten, Denis Wirtz, and Scot C Kuo, *Particle tracking microrheology of complex fluids*, Physical Review Letters **79** (1997), no. 17, 3282.

- [13] Scott G McCalla and James H von Brecht, *Fronts under arrest: Nonlocal boundary dynamics in biology*, Physical Review E **94** (2016), no. 6, 060401.
- [14] Salman S Rogers, Thomas A Waigh, Xiubo Zhao, and Jian R Lu, *Precise particle tracking against a complicated background: polynomial fitting with Gaussian weight*, Physical Biology **4** (2007), no. 3, 220.
- [15] T Andrew Sebrell, Barkan Sidar, Rachel Bruns, Royce A Wilkinson, Blake Wiedenheft, Paul J Taylor, Brian A Perrino, Linda C Samuelson, James N Wilking, and Diane Bimczok, *Live imaging analysis of human gastric epithelial spheroids reveals spontaneous rupture, rotation and fusion events*, Cell and Tissue Research **371** (2018), no. 2, 293–307.
- [16] Therese Seidlitz, Sebastian R Merker, Alexander Rothe, Falk Zakrzewski, Cläre von Neubeck, Konrad Grützmann, Ulrich Sommer, Christine Schweitzer, Sebastian Schölch, Heike Uhlemann, et al., *Human gastric cancer modelling using organoids*, Gut **68** (2019), no. 2, 207–217.
- [17] Martin B Short, Andrea L Bertozzi, and P Jeffrey Brantingham, *Nonlinear patterns in urban crime: Hotspots, bifurcations, and suppression*, SIAM Journal on Applied Dynamical Systems **9** (2010), no. 2, 462–483.
- [18] Martin B Short, Maria R D’orsogna, Virginia B Pasour, George E Tita, Paul J Brantingham, Andrea L Bertozzi, and Lincoln B Chayes, *A statistical model of criminal behavior*, Mathematical Models and Methods in Applied Sciences **18** (2008), no. supp01, 1249–1267.
- [19] C Su, K Bieniek, W Liao, MA Constantino, SM Decker, BS Turner, and R Bansil, *Comparison of motility of h. pylori in broth and mucin reveals the interplay of effect of acid on bacterium and the rheology of the medium it swims in*, bioRxiv (2020).
- [20] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet, *Novel type of phase transition in a system of self-driven particles*, Physical Review Letters **75** (1995), no. 6, 1226.