Models and Simulations of Collective Motion in Biomimetic Robots and Bacteria

Joanna Cohen
B.S. in Mechanical Engineering
California Institute of Technology, 2005

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2007

© Massachusetts Institute of Technology 2007. All rights reserved.

Author
Department of Mechanical Engineering
May 18, 2007

Certified by
Michael S. Triantafyllou
Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Certified by
Vick K-F Yue
Professor of Hydrodynamics and Ocean Engineering
Thesis Supervisor

Certified by
Jean-Jacques E. Slotine
Professor of Mechanical Engineering and Information Sciences
Thesis Supervisor

Accepted by
Lallit Anand
Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students
Models and Simulations of Collective Motion in Biomimetic Robots and Bacteria

by

Joanna Cohen

Submitted to the Department of Mechanical Engineering on May 18, 2007, in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Abstract

In nature, one finds many examples of collective motion, from flocking birds to swarming bees. Any one organism makes its decisions based solely on local information; either it can sense what its close neighbors are doing, or in the case of a single-celled organism, it can sense some local property of its environment. Yet complex global behaviors arise from these local interactions, and these large-scale patterns have neither a leader nor any other centralized control system. In this thesis, two specific cases of collective motion are studied: fish schooling and bacteria swimming across a surface.

When fish swim in schools, they swim in the same direction as each other at approximately the same speed. Previous studies of fish have discovered three primary behaviors that, together, lead to large-scale coordination and schooling in the animals. This thesis demonstrates that the same algorithms can be applied to a group of identical underwater robots. If the robots need to coordinate with each other, they can use biomimetic control laws and adopt the interaction algorithms used by fish. A series of simulations are run to see what possible group behaviors can come from these control laws.

At a smaller scale, prior experiments have revealed that bacteria and other small organisms also show collective motion. Unlike fish, bacteria cannot see their neighbors; the individual can only sense the bulk contribution of its neighbors to the flow at its location. The single-celled organisms are small and swim slowly, so they have very small Reynolds numbers. They are modeled in this work in a Stokes flow regime; the model is built bottom-up starting from the hydrodynamic field created by one organism and then superimposing these fields on top of each other. Different possible control policies are tested where each organism has an instantaneous desired direction based on some local property of the flow. While simulations of the current model
do not yield results that fully emulate real bacteria, they have some similarities and provide insight into the complex hydrodynamic interactions between low Reynolds number swimmers.

Thesis Supervisor: Michael S. Triantafyllou
Title: Professor of Mechanical and Ocean Engineering

Thesis Supervisor: Dick K-P Yue
Title: Professor of Hydrodynamics and Ocean Engineering

Thesis Supervisor: Jean-Jacques E. Slotine
Title: Professor of Mechanical Engineering and Information Sciences
Acknowledgments

This research and thesis would not have been completed without the support of many people. I would like to thank:

- My research advisors Professor Michael Triantafyllou, Professor Dick Yue, Professor Jean-Jacques Slotine, and Dr. Hover for supporting me throughout my time here. They managed my unusual research situation extremely well and helped me keep my focus as I worked on multiple projects at once. They constantly challenged me and this helped me achieve more than I thought possible.

- The members of the Nonlinear Systems Laboratory, the Towing Tank Laboratory, and the Vortical Flow Research Laboratory for providing me with interesting and insightful conversations, along with other students and colleagues throughout the Institute.

- The American Society of Engineering Education for funding me via an NDSEG fellowship.

- Professor Mitchel Resnick of the MIT Media Lab for letting me borrow Lego Mindstorms controllers, motors, and sensors. Even though I did not get to play with them as much as I would have liked, I am thankful for his generosity.

- My wonderful, supportive family and friends who kept me sane and always had encouraging words for me even when I could not find any of my own.
Contents

1 Introduction 17

2 Collective Motion of Fish Schools and Application to Biomimetic Robots 21

2.1 Motivation 21

2.2 Background 22

2.2.1 Models of Fish Schooling and Related Swarming Algorithms 22

2.2.2 Contraction Theory 23

2.3 Model Used in this Research 27

2.4 Theoretical Analysis 30

2.4.1 Definition of Synchronization in This Model 30

2.4.2 Synchronization Analysis in this Model Using Contraction Theory 31

2.5 Simulations and Results 34

2.5.1 Basic Model 34

2.5.2 Incorporating Individual Dynamics 38
2.5.3 Switching to Nearest-Neighbor Network .................. 40
2.5.4 Adding a Leader ............................................. 42
2.5.5 External Inputs .............................................. 44

2.6 Discussion ..................................................... 46

3 Collective Motion at Low Reynolds Number .............. 49

3.1 Background and Motivation .................................. 49
  3.1.1 Observed Phenomena and Models ......................... 49
  3.1.2 Motivation for New Proposed Model ..................... 57
  3.1.3 Stokes Flow .............................................. 58

3.2 Modeling and Simulation Details ......................... 68
  3.2.1 Model Characteristics .................................. 68
  3.2.2 Simulation Characteristics ............................... 81
  3.2.3 Self-Validation Tests and Internal Error ............... 83

3.3 Simulations and Results ...................................... 91
  3.3.1 Pattern Simulations ..................................... 92
  3.3.2 Randomized Simulations ................................. 103

3.4 Discussion .................................................. 113

4 Conclusions .................................................. 115

4.1 Contributions of this Research ............................. 115
4.1.1 Biomimetic Robots ............................................. 115
4.1.2 Low Reynolds Number Swimmers .............................. 116

4.2 Future Work ............................................................ 117

4.2.1 Biomimetic Robots ............................................. 117
4.2.2 Low Reynolds Number Swimmers .............................. 118
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Positions of Ten Elements at Different Points in Time</td>
<td>35</td>
</tr>
<tr>
<td>2-2</td>
<td>Velocities of Ten Elements as Time Elapses</td>
<td>36</td>
</tr>
<tr>
<td>2-3</td>
<td>Dependence of Convergence Time on Coupling Gains</td>
<td>37</td>
</tr>
<tr>
<td>2-4</td>
<td>Dependence of Convergence Time on Potential Function Shape</td>
<td>38</td>
</tr>
<tr>
<td>2-5</td>
<td>Velocities of Ten Elements as Time Elapses in Model with Individual Dynamics</td>
<td>40</td>
</tr>
<tr>
<td>2-6</td>
<td>System Evolution for Fifty Elements in Model with Individual Dynamics</td>
<td>41</td>
</tr>
<tr>
<td>2-7</td>
<td>Velocities of Ten Elements as Time Elapses in Model with Nearest-Neighbor Network</td>
<td>43</td>
</tr>
<tr>
<td>2-8</td>
<td>Velocities of Ten Elements as Time Elapses in Leader-Follower Model</td>
<td>44</td>
</tr>
<tr>
<td>2-9</td>
<td>Snapshots of Simulation with External Inputs</td>
<td>47</td>
</tr>
<tr>
<td>3-1</td>
<td>Velocity and Pressure Fields Around a Stokeslet</td>
<td>61</td>
</tr>
<tr>
<td>3-2</td>
<td>Velocity and Pressure Fields Around a Stresslet</td>
<td>64</td>
</tr>
<tr>
<td>3-3</td>
<td>Streamlines and Flow Speed Around a Rotlet</td>
<td>65</td>
</tr>
</tbody>
</table>
3-4 Streamlines and Magnitude of Velocity in Flow Due to Potential Singularities

3-5 Geometry of a Bacterium

3-6 Effect of Changing Regularizing Function Width on Velocity Near Singularity Center

3-7 Contributions of Singularities Placed at Center of Sphere

3-8 Reduction of Error Caused by Superimposing Potential Singularities onto Stresslet

3-9 Flow Field Produced by Two Potential Dipoles Superimposed on Stresslet

3-10 Flow Field Produced by Potential Dipole Superimposed on Stresslet as Used in Simulations

3-11 Initial Conditions for Test Cases 1 (left) and 2 (right)

3-12 Dependence of Speed and Error on L and n in Test Case 1

3-13 Dependence of Speed and Error on L and n in Test Case 2

3-14 Errors on Bacterium Surface Induced by Superposition of Flows

3-15 Fluid Motion Produced by Uniform Flow Over Different Body Shapes

3-16 Difference Between Uniform Flow Around Sphere and Ellipsoid

3-17 Paths of 6 Bacteria in Ring Configuration Swimming Up Pressure Gradients

3-18 Pressure Field and Gradients in Simulations with 6 Bacteria in Ring Configuration with $B = 0.4565$
3-19 Pressure Field and Gradients with 8 Bacteria in Disc Configuration
with \( B = 1 \) ......................................................... 96

3-20 Initial Straight Line Configuration ........................................ 97

3-21 Paths of 10 Bacteria Starting in Line Configuration with Drag-Reduction
Control Mechanism ....................................................... 98

3-22 Speeds and Orientations of 10 Bacteria Starting in Line Configuration
with Drag-Reduction Control Mechanism ........................................ 99

3-23 Changes in System Behavior Arising from Changes in B .......................... 100

3-24 Changes in System Behavior Arising from Changes in Initial Separation
Distance ................................................................. 102

3-25 Bacteria Paths with Even and Odd N ........................................ 103

3-26 Changes in System Behavior Arising from Changes in Number of Bac-
teria (N) ........................................................................ 104

3-27 Normalized Flow Velocity at \( t = 40 \) .................................................. 106

3-28 Logarithmic Plot of Mean-Square Distance of Tracers versus Time .... 108

3-29 Value of Alignment Parameter at \( t = 40 \) for Different Values of R .... 110

3-30 Average Area Fraction of Values of Alignment Parameter .................. 111

3-31 Average Speed of Bacteria with Different Control Policies ................ 112
List of Tables

2.1 Values of Constant Parameters Used in Simulations ............... 34
Chapter 1

Introduction

Many different life forms, from bacteria to insects to fish, exhibit collective behaviors. A large group of organisms appear to cooperate with each other to achieve some common goal that they could not achieve individually. In bees and ants, the collective behaviors can be extremely complex. Ants form colonies where each individual has a purpose and they work together to gather food and avoid predators, in effect creating their own society [52]. Bee swarms can contain up to ten thousand individuals, yet they are capable of making group decisions to choose nesting sites [37]. A small subset of the swarm searches for new sites, and when a scouting bee returns to the rest of the group, it performs a “waggle dance” in an attempt to recruit other bees to the site that it found. Behaviors such as this are rich topics of research that delve into the sociology of animal groups.

A subcategory of collective behavior is collective motion, which occurs when organisms move together or their motion forms some large-scale pattern and sociological aspects do not have much effect. In most examples of collective motion, the organisms are not assigned separate tasks from each other. This does not mean that they cannot take on different roles, but there is no global task coordination such as that seen in insects. An intriguing characteristic of observed collective motion phenomena
is that without a leader or a centralized control system, global coordinated motions can still occur. These motions can be simple; in the absence of external stimuli, fish and birds might swim or fly in formation in a straight line at a constant speed. They can also be complicated and have a random influence, such as in colony of bacteria displaying patterns that have a timescale of about a second [26].

In [39], Shimoyama et al. develop a general model that encompasses a number of different collective motion behaviors in a simple set of equations. By changing a few parameters, the system behavior can be altered drastically. When inertia is ignored, they find four general behaviors that their model can display:

1. Marching: organisms move together in a straight line with their relative positions staying constant

2. Oscillating: center of mass of system moves in a straight line but individual organisms change positions relative to each other, or center of mass moves in a circular path while the relative positions of individuals are constant

3. Wandering: center of mass moves in an irregular path while relative positions stay constant

4. Swarming: center of mass moves in an irregular path and organisms move around it irregularly, but general cluster persists

These behaviors are similar to behaviors observed in real organisms. For example, as mentioned above, fish and birds can move together as if in a marching state, while a swarm of bees traveling from one place to another exhibits characteristics if the swarming state in this model. The model is not intended to extract information about what each organism is actually thinking or how it makes decisions, but its results do capture the motions of a number of different species.

In this thesis, two specific examples of collective motion are examined in depth. Chapter 2 models a school of fish and then applies that model to a simulated group
of underwater robots. It discusses why the model is effective in different cases and possible ways it could be expanded. Chapter 3 models bacteria and other organisms swimming at low Reynolds numbers, a regime where viscous forces dominate inertial effects to the point where inertia can be neglected completely. While different types of bacteria exhibit slightly different spatial patterns, some characteristics are common across many species. Unlike fish, which can see each other from a distance, individual bacteria cannot sense their neighbors using sight, so their interactions must be based on hydrodynamic effects. The model is built from the bottom up starting with the hydrodynamic flow field produced by each organism, which then makes decisions based on the net contribution of the other organisms at its local position.
Chapter 2

Collective Motion of Fish Schools and Application to Biomimetic Robots

2.1 Motivation

Using limited sensors and primitive brains, fish such as tuna, mackerel, and herring are capable of complex behaviors within groups of varying sizes [42]. The general term for these behaviors is *schooling*. Many fish swim together in a closely-packed group and their proximity to each other provides a few advantages. It allows them to respond to threats more quickly; one fish can see a predator and change direction sharply, and the other fish will follow. Similarly, a fish that sees a source of food can direct other fish in that direction. Within a school, fish position themselves behind each other which allows them to take advantage of the vortices shed in their wakes. They use this hydrodynamic effect to swim more quickly [43]. Each element in this biological system has simple individual dynamics and local couplings, so overall group control is distributed and the complexity does not scale with increased group size.
Because fish have the advantage of millions of years of evolution, their behavior must provide optimal group performance. A system of underwater robots can be used to emulate the behaviors of school-forming fish. This biomimetic approach is beneficial because properties of fish schools such as distributed control and algorithm scalability embody the desired characteristics of robot swarms. Group cooperation is ideal for robotics applications because many simple, identical robots are easier to produce and control than one complex robot. The system is also more robust and reliable than using one robot; if an individual is destroyed or somehow gets lost from the swarm, the system quickly adapts and readjusts without losing functionality. Performing robotics tasks with a swarm allows complicated machinery to be replaced with robust control laws.

2.2 Background

2.2.1 Models of Fish Schooling and Related Swarming Algorithms

In studies of fish, three local behaviors have been identified that lead to a global schooling behavior. At a short range, fish swim away from each other to avoid collisions. At longer separation distances, they swim towards each other. Also, they adjust their orientations to align with their neighbors. A number of models have tried to determine the situations in which each of these behaviors dominate. Some divide the area around a fish into discrete concentric circular regions where a neighboring fish in a given region determines the original fish's behavior. Each fish exhibits one behavior at a time with respect to each of its neighbors, and the behavior is solely dependent on the separation between them [18]. Other models, such as Kunz and Hemelrijk’s, give each behavior a continuous distribution instead of a discrete region of influence [23]. The weight of each behavior depends on both the distance between fish and also their relative angular positions. This model uses a large attraction re-
gion where the attraction is strongest near the limits of the fish’s sensing radius, a small repulsion region that is strongest in front of the fish, and an alignment region that is concentrated on the sides of the fish. Taking another approach, Stöcker [42] and Tu and Terzopoulos [45] each set up flow charts to display their decision-making algorithms. In both cases, the dominant behavior is repulsion; if one fish gets too close to another, its primary objective will be to swim away. If the area immediately surrounding a fish is empty, then the fish will focus on the other behaviors of attraction and alignment.

Biomimetic robot swarming algorithms similar to the one presented later in this paper tend to incorporate the repulsion and attraction behaviors seen in schooling fish into a single control strategy, separating them into discrete regions. The repulsion region is a circle around the robot and the attraction region is a larger circle around the repulsion region. Sometimes the alignment region is considered to act throughout both of these regions [44] while other times it is ignored completely because the robots are trying to perform a task other than schooling [35],[47]. This paper focuses on the former approach.

2.2.2 Contraction Theory

A general, nonlinear system with state vector \( x \) where the evolution of the system also depends on the time \( t \) can be written:

\[
\dot{x} = f(x, t)
\]

(2.1)

This system is said to be contracting if initial conditions are forgotten exponentially fast, so all trajectories converge to a unique trajectory [25]. Contraction is analogous to stability in a linear system, but it is more general because linear stability only applies to a fixed point in space. To determine whether a system is contracting, one
can look at the difference $\delta x$ between two neighboring trajectories and how the square of its magnitude changes in time.

\[
||\delta x||^2 = (\delta x^T \delta x)
\] (2.2)

\[
\Rightarrow \frac{d}{dt} (||\delta x||^2) = \frac{d}{dt} (\delta x^T \delta x) = 2\delta x^T \delta \dot{x} = 2 \delta x^T \frac{\partial f}{\partial x} \delta x
\] (2.3)

A common result in linear algebra states that for any vector $y$ and any matrix $M$,

\[
y^T My^T \leq \lambda_{\text{max}} y^T y
\] (2.4)

where $\lambda_{\text{max}}$ is the largest eigenvalue of the symmetric part of $M$. Applying this to (2.2) gives:

\[
\frac{d}{dt} (\delta x^T \delta x) \leq 2\lambda_{\text{max}} (\delta x^T \delta x)
\] (2.5)

Here, the matrix $M$ is the Jacobian of the system in (2.1), so if $J = \frac{\partial f}{\partial x}$, then $\lambda_{\text{max}}$ is the largest eigenvalue of $J_s = \frac{1}{2} (J + J^T)$. Looking at this differential equation, one can see that if $\lambda_{\text{max}} < 0$, $(\delta x^T \delta x) \to 0$ exponentially, so $||\delta x|| \to 0$ exponentially as well. If this is true for all values of $x$, then all trajectories will exponentially converge to each other. Thus a sufficient condition for contraction is that all eigenvalues of $J_s$ are negative or, written another way,

\[
\left( \frac{\partial f}{\partial x} \right)_s \text{ negative definite}
\] (2.6)
Partial Contraction

As proven in Theorem 1 of [50], a nonlinear system like (2.1) can be written as

\[ \dot{x} = f(x, x, t) \quad (2.7) \]

which allows the creation of an auxiliary system of the form

\[ \dot{y} = f(y, x, t) \quad (2.8) \]

If the auxiliary system is contracting in \( y \), then the original system is said to be partially contracting. This has a powerful implication for synchronization. Start with two identical systems with states \( x_1 \) and \( x_2 \) that are coupled to each other where each subsystem is not fully contracting on its own. If a contracting auxiliary system can be formed that has particular solutions \( y = x_1 \) and \( y = x_2 \), then the two subsystems will synchronize to each other. Since each is a trajectory of (2.8) and all trajectories of (2.8) approach each other exponentially, \( x_1 \to x_2 \) exponentially.

Application of Contraction: Concurrent Synchronization

Starting from (2.1) again, assume that there exists a flow-invariant linear subspace \( \mathcal{M} \) so that any trajectory starting in \( \mathcal{M} \) remains in \( \mathcal{M} \) for all time [31]. Let \( n = \text{dim}(x) \) and \( p = \text{dim}(\mathcal{M}) \). Then, find a set of orthonormal basis vectors \( \{\hat{e}_1, \ldots, \hat{e}_n\} \) with the first \( p \) vectors forming a basis of \( \mathcal{M} \). The remaining \( n - p \) vectors will form a basis of the subspace \( \mathcal{M}^\perp \) that lies perpendicular to \( \mathcal{M} \). These vectors can be put into a matrix \( V \) so that the rows of \( V \) are \( \hat{e}_{p+1}^T, \ldots, \hat{e}_n^T \); \( V \) can be thought of as a projection matrix mapping \( x \) onto \( \mathcal{M}^\perp \). As described in [31], a sufficient condition for exponential convergence to \( \mathcal{M} \) is

25
As a simple example, consider an autonomous system with four scalar subsystems. Subsystems 1 and 2 are identical to each other and connected via diffusive couplings, as are subsystems 3 and 4; system 1 is also coupled to system 2, and system 3 to system 4. The full dynamics are:

$$\mathbf{x} = \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} f_1(x_1) + f_{c1}(x_3) + k(x_2 - x_1) \\ f_1(x_2) + f_{c1}(x_4) + k(x_3 - x_2) \\ f_2(x_3) + f_{c2}(x_1) + k(x_2 - x_3) \\ f_2(x_4) + f_{c2}(x_2) + k(x_3 - x_4) \end{bmatrix}$$

(2.10)

with scalar functions $f_1$, $f_2$, $f_{c1}$, and $f_{c2}$. The invariant subspace can be defined by $\mathcal{M} = \{ \mathbf{x} \mid x_1 = x_2, x_3 = x_4 \}$ so

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

(2.11)

defines the projection of $\mathbf{x}$ onto $\mathcal{M}^\perp$. Some algebra yields that the matrix from (2.9) is:

$$V \left( \frac{\partial f}{\partial \mathbf{x}} \right)_s V^T = \begin{bmatrix} f'_1 - 2k & \frac{1}{2} (f'_c + f'_{c2}) \\ \frac{1}{2} (f'_c + f'_{c2}) & f'_2 - 2k \end{bmatrix}$$

(2.12)

so if this matrix is negative definite, then the system will exponentially converge to the linear subspace $\mathcal{M}$ where $x_1 = x_2$ and $x_3 = x_4$. 

26
Nonlinear Manifolds

In [30], Pham and Slotine present some preliminary results regarding the situation where $\mathcal{M}$ is a nonlinear manifold with codimension 1, as opposed to the linear manifold described in the previous section. First, they assume that there is a vector $\mathbf{u}$ that is never tangent to $\mathcal{M}$. Take $p(x)$ as the projection of $x$ on $\mathcal{M}$ and $r(x)$ as the distance from $x$ to $p(x)$ along $\mathbf{u}$. Then $x$ can be written as $x = p(x) + r(x)\mathbf{u}$. The vector $\mathbf{J}_r(x)$ is the Jacobian of the system with respect to $r(x)$; it is orthogonal to $\mathcal{M}$ everywhere and its magnitude is always constant in the direction of $\mathbf{u}$. The matrix $\mathbf{J}_f$ is the standard Jacobian of the system with respect to $f(x)$. With this notation, a sufficient condition for exponential convergence to $\mathcal{M}$ is:

$$\mathbf{J}_r(x)\mathbf{J}_f(x, t)\mathbf{u} \text{ negative definite} \quad (2.13)$$

This can be extended to nonlinear manifolds of higher codimension by finding $p$ independent vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_p\}$ that are never tangent to $\mathcal{M}$. The unit vector $\mathbf{u}$ is replaced by the matrix $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_p]$ which represents an orthonormal basis. The projection of $x$ onto $\mathcal{M}$ is defined by the vector $\mathbf{p}(x)$, and the vector $\mathbf{R}(x)$ replaces the scalar distance $r(x)$. Each component $R_i(x)$ of $\mathbf{R}(x)$ describes the distance from $\mathbf{p}(x)$ to $x$ in the $\mathbf{u}_i$ direction.

2.3 Model Used in this Research

The basic model used here is a slightly modified version of the one used by Tanner et al. in [44]. While the algorithm that the model employs comes from observations of fish and other schooling or flocking organisms, the analysis here refers to the objects modeled as “elements.” This abstract term could refer to a living organism or to a robot depending on the context and application.
Each element is represented as a point, and the elements have identical second-order dynamics. An individual one is defined by its position and its velocity, and its control terms are introduced as forces. The dynamic equations for element $i$ (where $i = \{1, \ldots, N\}$ and $N$ is the total number of elements) are written

\[
\frac{d}{dt} \begin{bmatrix}
  r^i \\
  v^i
\end{bmatrix} = \begin{bmatrix}
  v^i \\
  F^i_{c1} + F^i_{c2}
\end{bmatrix}
\]  

(2.14)

where $r^i$ is the position of element $i$, $v^i$ is the velocity of element $i$, and $F^i_{c1}$ and $F^i_{c2}$ are coupling forces.

Specifically, $F^i_{c1}$ is a “velocity-matching” force of the form

\[
F^i_{c1} = \sum_{j \in N_i} k(v^j - v^i)
\]  

(2.15)

for some constant $k > 0$. The parameter $N_i$ denotes the set of neighbors of element $i$. If the group of elements is thought of as a network where each element is a node, then $N_i = \{j \mid i, j$ connected by edge\}. This coupling term causes the elements to align with each other and it also helps with the cohesion of the group. It forces stragglers to speed up and elements that are going faster than the rest of the group to slow down. The second coupling term $F^i_{c2}$ takes the form

\[
F^i_{c2} = \sum_{j \in N_i} \nabla_{r^i} U(r_{ij})
\]  

(2.16)

where $U$ is a scalar potential function. Since the coupling force is proportional to the gradient of this potential function, element $i$ feels a force pushing it towards the local minimum of $U$. The value of the potential function depends on the distance between elements $i$ and $j$, denoted by $r_{ij} = ||r^j - r^i||$, and the function has the following properties:
1. $U(r_{ij})$ is differentiable and radially unbounded

2. $U(r_{ij}) \to \infty$ as $r_{ij} \to 0$

3. $U(r_{ij})$ has a unique minimum at $r_{ij} = r_d = \text{desired spacing between elements}$

The first condition ensures that the gradient of the potential function is always a well-defined quantity, while the second and third provide a means of implementing the attractive and repulsive couplings between elements. When $r_{ij} < r_d$, elements $i$ and $j$ feel a repulsive force and are pushed away from each other. Since the magnitude of the force goes to infinity as the distance between elements gets infinitely small, collisions are inherently prevented. When $r_{ij} > r_d$, elements $i$ and $j$ feel an attractive force that pushes them closer together and encourages group cohesion. Thus $F_{c1}$ and $F_{c2}$ together account for the three characteristic behaviors of schooling fish: alignment, long-range attraction/cohesion, and short-range repulsion.

The specific choice of $U$ is taken from studies of intermolecular potential functions which automatically satisfy the conditions listed above. The $n$-$m$ potential defined in [6] has the form

$$U(r_{ij}) = U_0 \left[ \frac{1}{m} \left( \frac{r_d}{r_{ij}} \right)^m - \frac{1}{n} \left( \frac{r_d}{r_{ij}} \right)^n \right]$$

where $U_0$ is a positive constant, $m$ and $n$ are positive integers with $m > n \geq 0$, and $r_d$ is the distance at which $U$ reaches its global minimum.


2.4 Theoretical Analysis

2.4.1 Definition of Synchronization in This Model

A group of elements is said to display synchronized behavior when it resembles a school of fish. There are two requirements for synchronization:

1. all velocities are equal \( \mathbf{v}^1 = \cdots = \mathbf{v}^N \)

2. the elements are closely packed

As discussed in [44], the elements will be closely packed if

\[
\sum_{j \in N_i} \nabla_x U(r_{ij}) \to 0 \quad \forall \ i
\]  

(2.18)

Since the simulation tracks the velocity of each element and updates it throughout time, it is easy to observe when the first condition is satisfied. The gradients of the local potential functions are not stored, however, so some analysis is required to determine when the second condition is met. Since each element calculates its coupling forces by summing the contributions from its neighbors, the edges between neighboring elements form a network whose graph can be characterized by an incidence matrix \( \mathcal{B} \) of dimension \( N \times m \) (where \( m \) is the total number of connections). If edge \( k \) connects nodes \( i \) and \( j \) and \( i < j \), then

\[
\mathcal{B}_{ik} = -1
\]

(2.19)

\[
\mathcal{B}_{jk} = +1
\]

(2.20)

\[
\mathcal{B}_{lk} = 0 \quad \forall \ l \neq i, j
\]

(2.21)
If the velocities of the elements are equal to each other, the coupling terms $F_{c1}$ all equal zero, so that for each element, $\dot{v}^i = F_f^i$. Letting $v_{(i)} = (v^1, \ldots, v^N)^T$, the resulting dynamics are:

$$\dot{v}_{(i)} = \dot{v}^1 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = B \begin{bmatrix} \vdots \\ F_{c2}^{ij} \\ \vdots \end{bmatrix} \quad (2.22)$$

For a connected graph, the matrix $L = BB^T$ has a zero eigenvalue of multiplicity one corresponding to the eigenvector $(1, \ldots, 1)^T$, so this eigenvector lies in the null space of $B$. From (2.22) it follows that

$$\dot{v}^i = 0 \Rightarrow F_{c2}^{ij} \rightarrow 0 \ \forall \ i, j \quad (2.23)$$

so condition (2.18) is satisfied. When the elements' velocities are equal, they will be tightly packed, and the velocity information is sufficient to show both conditions for synchronization.

\section*{2.4.2 Synchronization Analysis in this Model Using Contraction Theory}

The simplest version of this model comes from setting $N = 2$ and analyzing how two coupled elements interact with each other. In two dimensions, the state vector $x$ has eight components.
The nonlinear manifold $\mathcal{M}$ is defined by three constraints. Two come from the velocity-matching term ($v^1_x = v^2_x$ and $v^1_y = v^2_y$) and the third comes from the cohesion coupling term ($\|r^2 - r^1\| = r_d$). When the third constraint is satisfied, each element lies at the minimum of the other’s potential function, so the gradient will be zero. Because of this, $\mathcal{M}$ represents an invariant set with codimension 3. Three unit vectors that lie orthogonal to it are:

$$\begin{align*}
\hat{u}_1 &= \frac{1}{\sqrt{2}}(0,0,1,0,0,0,-1,0)^\top \\
\hat{u}_2 &= \frac{1}{\sqrt{2}}(0,0,0,1,0,0,0,-1)^\top \\
\hat{u}_3 &= \frac{1}{2\|r^2 - r^1\|^2}(r^1_x - r^2_x, r^1_y - r^2_y, 0, 0, r^2_x - r^1_x, r^2_y - r^1_y, 0, 0)^\top
\end{align*}$$

(Note that in (2.27), the norm written before the vector is squared; all other superscripts identify the element and are not exponents.) The vector $R(x)$ is defined by:
\[
R(x) = \begin{bmatrix}
R_1(x) \\
R_2(x) \\
R_3(x)
\end{bmatrix} = \begin{bmatrix}
v_x^1 - v_x^2 \\
v_y^1 - v_y^2 \\
||r^2 - r^1|| - r_d
\end{bmatrix}
\] (2.28)

When \(\mathcal{M}\) has codimension 1, \(J_r\) is a vector; with a higher-dimensional nonlinear subspace, \(J_R\) is a matrix. Specifically, in this case,

\[
J_R(x) = \begin{bmatrix}
0 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 \\
r_1^2 - r_2^2 & r_1^2 - r_2^2 & 0 & 0 & r_2^2 - r_1^2 & r_2^2 - r_1^2 & 0 & 0
\end{bmatrix}
\] (2.29)

The Jacobian \(J_r\) is found by taking the derivatives of

\[
f(x) = \begin{bmatrix}
v_x^1 \\
v_x^2 \\
k(v_x^2 - v_x^1) + U_0 \left( \frac{r_y^m}{||r^2 - r^1||^{n+2}} - \frac{r_x^m}{||r^2 - r^1||^{m+2}} \right) r_x^1 \\
k(v_y^2 - v_y^1) + U_0 \left( \frac{r_x^m}{||r^2 - r^1||^{n+2}} - \frac{r_y^m}{||r^2 - r^1||^{m+2}} \right) r_y^1 \\
k(v_x^1 - v_x^2) + U_0 \left( \frac{r_y^m}{||r^2 - r^1||^{n+2}} - \frac{r_x^m}{||r^2 - r^1||^{m+2}} \right) r_x^2 \\
k(v_y^1 - v_y^2) + U_0 \left( \frac{r_x^m}{||r^2 - r^1||^{n+2}} - \frac{r_y^m}{||r^2 - r^1||^{m+2}} \right) r_y^2
\end{bmatrix}
\] (2.30)

In the interest of space, the Jacobian is not printed here. To test whether the system synchronizes, one needs to find the eigenvalues of \(J_R J_f U\) (where \(U = [u_1 \ u_2 \ u_3]\)). If all eigenvalues are negative, then the state of the two-element system exponentially approaches \(\mathcal{M}\) and the elements synchronize. However, despite numerous simulations suggesting that this should be the case, the eigenvalues do not seem to all be less than zero. The introduction of a metric as defined in [25] might lead to a proof of synchronization, but such a metric has not yet been found.
2.5 Simulations and Results

2.5.1 Basic Model

For the first simulations, the model described in §2.3 is implemented in two dimensions using Matlab. The differential equation solver ode45 is used with its default conditions to integrate (2.14) through time using the values shown in table 2.5.1 chosen for the various constants in the equations. When the simulation is initialized, the positions of the elements are set randomly within a circle of radius $R_0 = 5r_d$. The velocities are set randomly as well with $v_x, v_y \in [-1, 1]$.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_d$</td>
<td>2</td>
</tr>
<tr>
<td>$k$</td>
<td>1</td>
</tr>
<tr>
<td>$U_0$</td>
<td>10</td>
</tr>
<tr>
<td>$m$</td>
<td>2</td>
</tr>
<tr>
<td>$n$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.1: Values of Constant Parameters Used in Simulations

The results at different time steps are shown in figure 2-1. Although the elements sometimes appear to overlap with each other, they are represented as infinitely small points in the simulation, the overlap is merely an artifact of them being plotted with a finite size. The white line in each circle represents the velocity of that element normalized with respect to the maximum speed at the given time step. Figure 2-2 shows the evolution of each element’s speed and orientation through time. By $t = 50$, the velocities of the elements have converged to the same value, as can be seen in figure 2-1(f).

To quantify the system convergence, the convergence time $t_c$ is defined as the time it takes for the system to reach a synchronized state. Here, synchronization is defined as a state where the largest discrepancy in speeds is less than 1% of the maximum speed at that time step and the largest discrepancy in orientations is less than 1% of $2\pi$. So if each element moves at speed $v^i$ at direction $\phi^i$ with respect to the $x$-axis,
Figure 2-1: Positions of Ten Elements at Different Points in Time
then

\[ t_c = \min(t) \text{ such that } \left\{ \begin{array}{l}
\max_i(v^i) - \min_i(v^i) < 0.01 \max_i(v^i), \text{ and } \\
\max_i(\phi^i) - \min_i(\phi^i) < 0.02\pi
\end{array} \right. \]  

(2.31)

Note that the condition \( t > t_c \) implies that the system is approaching a fully-synchronized state, so by (2.23), the elements will cluster together as the system converges. The convergence time \( t_c \) depends most consistently on the gain factors \( k \) and \( U_0 \) as shown in figure 2-3. The contour lines mark divisions in \( \ln(t_c) \).

The coupling term \( F_{cl} \) acts like a proportional controller that drives the velocities of the elements closer to each other, with \( k \) being the proportional gain. High values of \( k \) push the system harder towards synchronization but there is no derivative control,
so the overshoot is larger and the system takes longer to reach the synchronized state. From the data it follows that when $k > 0.25$ and $U_0$ is held constant, $t_c \sim k$. When $k < 0.25$, the system does still seem to synchronize, but the convergence time increases because the strength of $F_{ci}$ is lowered. If the value of $k$ is held constant, then $t_c \sim U_0^{-1}$.

![Graph](image)

(a) $t_c$ versus $k$

(b) $t_c$ versus $U_0$

![Graph](image)

(c) Contours of $t_c$ plotted against $k$ and $U_0$

Figure 2-3: Dependence of $t_c$ on $k$ and $U_0$

Even though they are not coupling gains per se, the constants $m$ and $n$ affect the shape of the potential function so they also affect $t_c$. The dependence of $t_c$ on these parameters is shown in figure 2-4. (Half of the plot appears white because of the
requirement that \( m > n \).) The apparent relationship between the parameters is highly nonlinear and complex. For simplicity, future simulations will be run with \( m = 2 \) and \( n = 1 \) instead of trying to deduce the optimal values of the parameters which would require a large number of trials.

### 2.5.2 Incorporating Individual Dynamics

The basic model assumes that the only forces acting on each element are the coupling forces given as inputs. The coupling forces lead to synchronization between the elements but they have no influence on the steady-state velocity of the group. Additional control forces can be added to set the group speed. These forces are referred to as “individual dynamics” because the force on element \( i \) depends only on the state...
of that element. The individual force $F_i$ takes the form

$$ F_i = -\gamma v + T \frac{v}{||v||} \tag{2.32} $$

with constants $\gamma$ and $T$ set to be identical for every element. $F_i$ acts as a proportional-derivative control input for the speed $||v||$ by pushing it towards $||v|| = T/\gamma$; when $||v|| = T/\gamma$, $F_i = 0$. The individual force also has a physical interpretation. It represents the force felt by an element that produces a constant thrust in its direction of motion, and also feels a drag force proportional to its velocity that acts in the opposite direction. In this interpretation, $T$ is the thrust coefficient and $\gamma$ is the drag coefficient. The full dynamics of the system become:

$$ \begin{bmatrix} \dot{r}^i \\ \dot{v}^i \end{bmatrix} = \begin{bmatrix} v^i \\ F_{c1}^i + F_{c2}^i + F_i^i \end{bmatrix} = \begin{bmatrix} v^i \\ \sum_{j \in N_i} [k(v^j - v^i) + \nabla_r U(r_{ij})] - \gamma v^i + T \frac{v^i}{||v||} \end{bmatrix} \tag{2.33} $$

The velocity of each element with this model is plotted in figure 2-5. The initial conditions and constants are the same as in the simulation shown in figure 2-2. For the additional parameters, $T = 3$ and $\gamma = 1$, which implies that the “desired” velocity is $||v|| = T/\gamma = 3$.

To demonstrate the scalability of this model, a similar simulation is run with $N = 50$. The velocity of the elements along with the initial and final configurations are shown in figure 2-6. The system is not fully synchronized at $t = 100$ but one can see that the group is growing more cohesive and the velocities are approaching a synchronized state.
Figure 2-5: Velocities of Ten Elements as Time Elapses in Model with Individual Dynamics

2.5.3 Switching to Nearest-Neighbor Network

The results presented so far assume that each element is connected to every other element. This network topology, where $\mathcal{N}_i = \{j \mid j \neq i\}$, is called an all-to-all network. However, this topology is not physically realistic because, whether in a biological system or a robotic one, each element can only sense its neighbors (e.g. within a sensing radius $R$). This presents two complications: firstly, it removes a lot of edges from the network, and secondly, it means that the topology of the network changes with time (called a “switching network”). The new network is a nearest-neighbor network and it is defined by

$$\mathcal{N}_i = \mathcal{N}_i(t) = \{j \mid r_{ij}(t) \leq R\} \quad (2.35)$$
Figure 2-6: System Evolution for Fifty Elements in Model with Individual Dynamics
As described in [50], if a system synchronizes with an all-to-all network configuration, then it should synchronize with a switching network topology as well, provided that the network is connected ∀t. Simulations show that appropriate initial conditions and choice of R let the network stay connected at all times. The initial conditions should be chosen such that every element \( i \) starts in the network by ensuring that \( \min_j(r_{ij}) < R \). Additionally, elements should not start too close to each other, because the strong repulsive force can push one out of range of all the other elements. To satisfy both of these constraints, the elements are initialized in a grid pattern with random noise in their positions.

The velocities of ten elements with a nearest-neighbor network and \( R = 3r_d \) are shown in figure 2-7.

The restrictions placed on the initial conditions might seem unrealistic or infeasible for a physical system. However, robots could work around the restriction in a similar way to how fish work around it. Since each element can sense its neighbors with the sensing radius \( R \), it can also sense when it has lost the group because its sensing area will be empty. At this point the element would change its behavior and start a searching algorithm. This paper does not develop such an algorithm.

### 2.5.4 Adding a Leader

If the constants \( T \) and \( \gamma \) in the individual control force terms are adjusted, the steady-state group speed can be tuned to a desired value. However, the direction the group travels in is still determined by the initial conditions. Specifically, because of the structure of the velocity matching term, the final direction is the average of the initial directions.

\[
\phi_{\text{steady-state}} = \frac{1}{N} \sum_{i=1}^{N} \phi_i(0)
\]  

(2.36)
The direction of the group can be controlled by letting one of the elements “lead” the others. Element 1 can become the leader by simply turning off its coupling forces. This makes the leader blind, since ignores its neighbors and does not make any decisions based on their states. The leader’s dynamics are:

\[
\frac{d}{dt} \begin{bmatrix} r^1 \\ v^1 \end{bmatrix} = \begin{bmatrix} v^1 \\ F^1_l \end{bmatrix} = \begin{bmatrix} v^1 \\ -\gamma v^1 + T \frac{v^1}{||v^1||} \end{bmatrix} \tag{2.37}
\]

Since the leader feels no coupling forces, it will continue moving in its initial direction for all time. As the system synchronizes, the other elements will converge to the velocity of the leader, meaning that the whole group will move in the leader’s initial direction as well. In a sense, adding a leader in this way is like using a centralized
control law; however, the only global input is the orientation of the single leader element. The other elements (termed "followers") are not controlled in any centralized or global way. They do not know which element is the leader, and there is no guarantee that they are all coupled to it since it might lie outside of their sensing radius. Figure 2-8 shows the velocities in a simulation with ten elements using the same parameters as the previous simulation. The leader's initial orientation is $\phi^1(0) = \pi/4$ and it can be seen that all elements synchronize to this orientation.

2.5.5 External Inputs

All the models and simulations discussed so far have focused on the elements' interactions with each other. Another interesting area of analysis is to look at the elements'
interactions with their environment. For fish, this could involve swimming towards a food source or away from a predator. A similar scenario for robots could involve a group searching an area for something of interest. It could be a thermal vent that scientists are trying to collect data on or an enemy ship in a harbor. In simulation, the area of interest (or “target area”) is shown as a dark gray circle. The model assumes that the elements have previously determined that dark gray circles have some significance and should be explored more closely. Although each element has “target sensors,” the model does not require that the elements all see the target area independently of each other; even if only one element senses the target, it can move in that direction and the other elements will follow it.

The elements need a new set of control laws to determine their behavior when they see a target area. Their sensors are assumed to work within a distance $R_{\text{target}}$ which essentially gives the target area an “influence region” centered at $r_{\text{target}}$ with a radius of $R_{\text{target}}$. When an element enters the influence region, it turns to move towards the target. The control input is an artificial spring-damper force

$$F_{\text{target}}^i = -K_p(||r^i - r_{\text{target}}||) - K_d v^i$$

with constant proportional gain $K_p$ and derivative gain $K_d$. Since the primary objective of an element that has sensed a target area is to move towards that target, the coupling gains of that element are lowered by a factor of 1000. The coupling gains are not turned off completely because the potential function goes to infinity at zero separation between elements regardless of the coupling strength, so even small-magnitude coupling forces will prevent collisions. The dynamics of an element within the influence region of a target are:

$$\frac{d}{dt} \begin{bmatrix} r^i \\ v^i \end{bmatrix} = \begin{bmatrix} v^i \\ F_{\text{target}}^i + 0.001 (F_{c1}^i + F_{c2}^i) + F_l^i \end{bmatrix}$$

(2.39)
Once an element is inside of the target area, it reverts back to the control laws described by (2.34), except with a smaller desired separation distance $r_d$ so that the elements form a tighter cluster. If the element reaches an edge of the target area and travels a small distance outside of it, it immediately switches back to the control laws described by (2.39) and returns to the area.

The simulation begins with the elements in random positions to the left of a field of circles. The leader element moves to the right towards the circles and the other elements follow it. Once they sense the target, they turn towards it, and soon all elements reach it. Snapshots of this sequence for six elements are shown in figure 2-9.

By $t = 20$, the elements have entered a synchronized state. The leader, marked with a white dot, is initialized moving to the right, so the group now moves in that direction. The field of circles is on the right and the one dark gray area is the target. None of the elements have entered the target's influence region. By $t = 40$, the lower two have sensed the target. (The large light gray circle represents the influence region.) At $t = 50$, the two elements that sensed the target first have entered the target area. Meanwhile, the remaining elements have been pulled into the influence region. This does not happen because they sense the target; they only turn to move downwards because they are coupled to the elements that do sense the target. Once they are pulled into the influence region by the coupling forces, they sense the target themselves and move towards it. By $t = 80$, all the elements have entered the target area and they are exploring it in a small cluster formation.

2.6 Discussion

The model presented here builds on previous models to show elements displaying schooling behavior using a realistic network structure with local, distributed control laws. It takes a biomimetic approach by making the general behavior of the elements mimic that of schooling fish. By combining individual control laws with coupling
Figure 2-9: Snapshots of Simulation with External Inputs (details in text)
terms and making one element "blind," the direction and speed of the school can be determined. This essentially allows the entire system behavior to be adjusted by changing one or two parameters. In addition to the internal control algorithms, some preliminary results are shown regarding the elements' response to external inputs. Future work could focus on this aspect of the model and also on the theoretical analysis of the system.
Chapter 3

Collective Motion at Low Reynolds Number

3.1 Background and Motivation

3.1.1 Observed Phenomena and Models

Experimental Results and Observations

Organisms such as protozoan zooplankton and bacteria, which are small and complex, swim through water at low Reynolds number ($Re$). Since these single-celled organisms do not have any means of sensing something at a distance from them (such as a visual or auditory system), they cannot distinguish between their neighbors. They can only “communicate” with each other by sensing the flow around them and reacting to it, thus modifying the flow sensed by their neighbors. Despite this limitation, patterns of collective motion have been observed in quite a few species. Genin et al. find evidence of zooplankton aggregation where the organisms gather at a specific depth in the ocean, even in the presence of strong upwelling or downwelling.
currents [15]. However, the authors do not determine whether the behavior is due to some cooperation between the organisms or if it is merely a result of each organism responding in the same way to some external stimulus.

In [14], experiments on the zooplankton species *Daphnia* show a combination of individual and cooperative behavior that lead to coordinated group motion. *Daphnia* are phototactic organisms, meaning that they are attracted to sources of light. When a low concentration of them are placed in an aquarium with a vertical shaft of light shining through it, they swim towards the light and then circle around it. Some swim clockwise while others swim counterclockwise, and each organism changes directions every so often. (In one example, an organism circled the light seven times and then reversed its direction of rotation.) At higher concentrations, the zooplankton continue to swim in circular paths, but they also begin to cooperate and swim in the same direction as each other. The group moves with unified circular motion as a combination of the individual behavior (swimming in a circle around a light source) and the collective behavior (aligning with their neighbors).

In a sessile drop of water, the chemotactic bacteria *Bacillus subtilis* swim up oxygen gradients towards the top of the drop and after some transient behavior, they reach a steady-state circulation pattern [13]. This is an example of bioconvection, a phenomenon that arises from the interplay of physics and biology when chemotactic bacteria are in constant motion [10]. First, the bacteria swim upwards towards the water-air interface at the top of the drop. The bacteria are denser than water so they form a heavy top layer, which results in a Rayleigh-Taylor instability [19]. This layer starts to wrinkle and the creases fall downwards creating plumes of bacteria. At the same time, near the edges of the drop, the heavy top layer slides down towards the water-air-solid interface and rolls up into a vortex near the contact line. Similar experiments are performed on a suspended drop, and pictures of the flow field on the bottom (at the water-air interface) show groups of bacteria swimming in vortices and other patterns. The bioconvection patterns are another example of the complex group motions that can arise from combination of individual and collective behaviors.
Another example of collective motion is seen in the spermatozoa of *Strongylocentrotus droebachiensis*, which swim in small rings along a planar surface in the absence of an external stimulus [36]. The behavior is believed to arise from hydrodynamic interactions between their flagella. Each ring, or vortex, contains approximately 10 sperm with their flagella beating in synchrony. The vortices all rotate clockwise, probably due to hydrodynamic effects as well, and they are arranged into a locally hexagonal array. Riedel et al. find that this pattern only appears when the concentration of sperm cells is above some critical value just as Erdmann et al. observed in *Daphnia*.

In experiments with *Proteus mirabilis* and *Bacillus subtilis* swimming across a flat surface, similar behaviors are observed, except the vortices resemble rotating disks instead of rings. *P. mirabilis* form patterns of targets and spirals [38] and *B. subtilis* show collective motion in the form of “whirls” and jets [26]. The chemotactic nature of these organisms is the primary factor contributing to the bioconvection patterns described in [13]. Here, however, each bacterium has similar oxygen access to its neighbors because they are confined to a thin layer along an agar plate. Mendelson et al. use this two-dimensional experimental domain in part because it makes video observation easier. While analyzing the video data, they observe patterns on three different length scales. First, on the smallest scale, individual bacteria tend to align with their neighbors. Second, groups of bacteria organize into jets and whirls (or vortices). Third, the vortices arrange themselves roughly into a locally hexagonal pattern.

While the general positions of the vortices stay constant over time, the behavior of each vortex changes over a timescale of about one second. A typical vortex might start in a clockwise direction, break up and form two opposing parallel jets, and then re-form into a counter-clockwise vortex. A vortex that breaks up and re-forms is more likely to switch to the opposite direction, but it can also rotate in the same direction as before. At any given time, clockwise and counter-clockwise vortices occur with about the same frequency, though neighboring vortices usually rotate in opposite directions from each other. This suggests that the bacteria are interacting
with each other and not just the surface they swim on, because individual organisms with eukaryotic flagella are much more likely to swim across a surface clockwise than counter-clockwise in the absence of bacterial interactions [12]. One reason why the organisms might swim in these patterns is to increase their speed and thus access nutrients more quickly; Mendelson et al. find that the bacteria in jets can swim over twice as fast as the individual bacteria swim on their own.

Three separate experiments run on *Escheria coli* provide evidence of collective motion by tracking passive particles in a solution with the bacteria instead of following the motion of the organisms themselves. Soni et al. use an optical trap to capture a single bead within a bacterial bath containing various concentrations of organisms [41]. The bead has a diameter of 3 \( \mu \text{m} \), slightly larger than the cell body of each organism. With no bacteria present, the bead’s position fluctuates a small amount due to Brownian motion. However, as the concentration of bacteria increases, the magnitude of these fluctuations increases to a point where they cannot be attributed solely to Brownian motion. The fluctuations show distinct timescales that suggest the bacteria are moving collectively instead of colliding with the bead randomly.

In [21], Kim and Breuer look at the mixing rate of two streams of fluid, one containing fluorescent particles and the other having the same concentration of non-fluorescent particles, as they combine in a Y-shaped chamber. The particles used are Dextran which have diameters on the order of 10 nm (much smaller than *E. coli*). In the absence of bacteria, the two streams mix at a rate consistent with standard diffusion as they move down the channel. When bacteria are placed in one of the streams, however, the mixing rate increases. Kim and Breuer calculate an apparent diffusion coefficient that increases linearly with the concentration of bacteria, implying that the presence of the swimming organisms causes the particles to superdiffuse and generally increases mixing.

Wu and Libchaber reach a similar conclusion in [53]. They look at a thin soap film with a high concentration of *E. coli* constricted to a quasi-two-dimensional domain.
The soap film also contains 10 μm-diameter tracer beads. In short timescales the authors find superdiffusive behavior of the beads, while in longer timescales the beads exhibit normal diffusive behavior. This indicates a critical timescale at the transition between diffusion and superdiffusion that describes the lifetime of transient formations of the bacteria. The authors also notice the appearance of vortices and jets in the bacterial bath similar to the patterns observed in [26]. There are no significant temperature, gravitational, or chemical gradients present in their experiment so the patterns must be solely a result of the hydrodynamic interactions between organisms.

Existing Models

A few models have been developed that try to explain the results described in section §3.1.1. Tuval et al. develop a mathematical model of bioconvection in [46] to explain the patterns observed in that paper and also in [13], which are driven by the chemotactic behavior of \textit{B. subtilis}. The domain modeled is the cross-section of a small water droplet on a solid surface. The governing equations for the model consist of three coupled PDEs for the oxygen concentration throughout the drop, the cell density, and the fluid velocity. At the water-surface interface, there is no flux of oxygen or bacteria; the fluid has no-slip and no-penetration boundary conditions. At the water-air interface, the concentration of oxygen is equal to its saturation value in air, there is no flux of bacteria, and the fluid flow field is stress-free. Numerical finite-element simulations of these equations produce images of the cell density that closely match the experimental results. In fact, the model is qualitatively so similar to the observed patterns that it can be used to provide additional information about the flow field and the O\textsubscript{2} concentration, quantities that cannot be observed as easily as cell density.

The strength of this model is its similarity to the experiments. By modeling cell concentration as a parameter of the entire field, however, it does not provide any insight to the individual's motivation. The bacteria try to swim up oxygen gradients
to maximize their uptake of this nutrient but the model cannot determine if every organism swims up the gradients in the same way; perhaps some feel a stronger influence and become leaders, while the other bacteria merely follow them. Also, it is unclear how the model would perform in the absence of large-scale oxygen gradients. Yet in the situation it describes, this model successfully matches the experimental results.

In [24], Lega and Passot take a different approach to modeling the swimming behavior of \textit{B. subtilis}. The domain they use is a thin layer of water spread over a nutrient-rich agar surface. The bacteria do not have much motility in the vertical direction, so they will not exhibit the three-dimensional patterns seen in bioconvection results such as [13]. Instead, the bacteria density stays relatively constant in the fluid, so Lega and Passot model the highly-concentrated bacteria and the water that they swim in as a single complex fluid. The governing equation for the velocity of the complex fluid is the Navier-Stokes equation. As in the previous model, the chemotactic behavior of the bacteria is incorporated into the calculations. Nutrients enter the fluid from the top and bottom, through both the water-air and water-agar interfaces. The authors use the reaction-diffusion equation and the Navier-Stokes equation to model the behavior of a bacterial colony in its center and at its edges where it spreads along the agar plate.

Numerical simulations of this model show how powerful it is in capturing the behavior of the organisms at multiple length scales. On a large scale, the model shows colony growth in finger-like protrusions, as observed in experiments. With the inclusion of a randomized local forcing term, the model shows vortex-like regions of rotating fluid on a small scale. The regions rotate in random directions as seen in [26]. The model suffers from the same problem as the previous model in that it does not capture the behavior of individual organisms, but it provides great insight into the behavior of bacteria swimming along an agar surface in an expanding colony.

Kitsunezaki looks at the behavior of \textit{P. mirabilis} in [22]. He notes that the collective
motion phenomena observed in these organisms are patterns of local directional order, not patterns of density. To capture this behavior, his two-dimensional model tracks the number of bacteria swimming with different orientations. One motivating factor behind the bacteria's orientation is that each organism wants to move away from areas of high cell density, presumably for more equal distribution of nutrients. The organisms also move via the diffusion-like effect of collisions between individuals. When the average concentration of organisms is increased past some critical value, his simulation shows clean spiral patterns in the steady-state equilibrium case. A parameter can be adjusted that allows only clockwise (or counter-clockwise) spirals to match experiments with this species. Kitsunezaki’s model gives insight into the mechanism behind the formation of such spirals, but the images obtained from the simulation are much more organized than any experimental results.

In [9], Czirók et al. build on Vicsek’s model from [48] by incorporating features that apply specifically to bacteria. Interestingly, while other papers describe bacteria swimming along a nutrient-rich agar plate, here, the bacteria are said to be in a “hostile environment” on a “hard agar substrate.” The fundamental control algorithm in this model, as in [48], is that each organism tries to align with its neighbors (subject to noise) while swimming at a constant speed. Building on this algorithm, the authors impose a vortical-type flow on the bacteria by changing the domain to a donut shape with reflective circular walls. Then, to force a circular flow without a circular boundary, the authors introduce two more layers of complexity. They abandon the assumption of constant speed and replace it with constant thrust. The speed is then determined by balancing the propulsive force with a drag term dependent on the local cell density. Then a form of chemotaxis is introduced; instead of following global oxygen gradients, however, each organism emits an attractant that other bacteria swim towards. The simulation is carried out on a hexagonal lattice and shows the formation of multiple vortices in opposite directions.

The advantages of this model are that its final results are similar to the experimental results presented in [26] and that each individual organism is tracked and accounted
for, unlike previously-discussed models which only track cell density. However, the model is put together in a rather piece-meal fashion. No physical explanation is given for some of the behaviors; for example, the drag force used is not determined by the hydrodynamic force. The hexagonal lattice is an unnatural constraint, and even the local alignment on which the whole system is based is not physically realistic for bacteria. Each organism might try to align itself with the local flow, but it cannot sense its neighbors or their swimming direction to find the local average orientation.

Hernandez-Ortiz et al. take an approach in [17] that yields the closest existing model to the one presented in this thesis. They model small flagellar bacteria such as *E. coli* or *B. subtilis* and create a simple model for each organism, then simulate the flow field through time. The domain they use is a thin square with solid walls on the top and the bottom. The sides have periodic boundary conditions and the bacteria are allowed to move in three dimensions. Each organism is represented as a dumbbell composed of two identical spheres connected by a massless rod; the orientation of the rod determines the angle of the thrust, which has a constant magnitude (equal to twice the drag on one sphere) and is provided by a “phantom flagellum” at the back of the organism. As the simulation progresses through time, the bacteria’s positions change based on their thrust, drag, and advection from the flow created by the swimming motions of the other organisms. To calculate the advection, the thrust from the flagellum and the drag on each sphere of the other bacteria are modeled as point forces or Stokeslets.

The simulation also tracks passive tracer particles that move purely via advection. At high bacteria concentrations and short timescales, the tracers exhibit superdiffusive behavior like that seen in [21] and [53]. The model presented by Hernandez-Ortiz et al. also finds a critical concentration above which the bacteria show collective motion. However, there is little other discussion of their results, even qualitatively. The authors do not give a physical justification for using two spheres to model one cell, and they do not explore what would happen if the bacteria had any decision-making capacity. The interaction between organisms is derived directly from the
hydrodynamic forces, which makes it more realistic than other models described so far.

Additional models examine other species that exhibit collective motion characterized by groups of organisms swimming in the same direction along a circular path. Erdmann et al. model swarms of the phototactic zooplankton *Daphnia* [14] and Riedel et al. model small rings formed by the spermatozoa of sea urchins on a flat surface. [36] While both of these models are insightful, neither starts from hydrodynamic interactions so they take a different approach from the model proposed here.

### 3.1.2 Motivation for New Proposed Model

The goal of the model proposed here is to simulate the collective motion behaviors observed in small organisms. A common pattern seen in a variety of experiments and species is that of vortices, whirls, or spirals spinning in different directions. In between these vortices, jets of bacteria swim together in a straight line at a fast speed. Using a minimal model of each individual organism, the basic hydrodynamic interactions between them can be calculated analytically. These organisms swim in a low Reynolds number regime, so the governing Navier-Stokes equations can be simplified to the linear equations of Stokes flow. Most of the experiments described restrict the bacteria to a two-dimensional domain for ease of observation and analysis, so this model takes the same approach. Each bacterium is assumed to swim in a straight line on its own; in the presence of an external flow field or other swimming bacteria, the bacterium is advected and rotated by the flow. The observed “run-and-tumble” behavior of *E. coli* and *B. subtilis* is not modeled, but experiments show that bacteria tumble less often when restricted to two-dimensional motions [53]. However, each organism can reorient itself based on some parameter of the flow, and a few possible reorientation schemes are proposed and tested. Their feasibility depends on the sensing capabilities of the bacteria which are not explored here. Testing different control policies can give insight into how the organisms would behave if such policies
3.1.3 Stokes Flow

Governing Equations

To model a hydrodynamic system in water, the basic governing equations used are the continuity equation

\[ \nabla \cdot \mathbf{u} = 0 \quad (3.1) \]

and the Navier-Stokes equation

\[ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u} + \mathbf{b} \quad (3.2) \]

where \( \mathbf{u}(\mathbf{x}, t) \) is the velocity at a position \( \mathbf{x} \) and time \( t \), \( P(\mathbf{x}, t) \) is the relative pressure at \( \mathbf{x} \) and \( t \), \( \mathbf{b} \) is a body force (usually gravity), \( \rho \) is the density of the fluid, and \( \nu \) is its kinematic viscosity [2]. The continuity equation represents conservation of mass, while the Navier-Stokes equation represents conservation of momentum. (Note that the incompressible forms of the equations are given because the fluid in question is water.) As described in [33], the pressure \( P \) can be replaced by a modified pressure term \( p = P - \mathbf{b} \cdot \mathbf{x} \). The distinction is only relevant when considering surface forces on the boundary, and the substitution cancels the body force term in (3.2).

With characteristic time, length, and velocity scales defined by \( T, L, \) and \( U \) respectively, the Navier-Stokes equations can be nondimensionalized with the dimensionless parameters \( \text{Re} = \frac{UL}{\nu} \) (Reynolds number) and \( \text{St} = \frac{L}{U_T} \) (Strouhal number) as follows:
The Strouhal number is relevant in situations where there is a characteristic frequency independent of the length and velocity scales; otherwise, $T = \frac{L}{U}$ and $St = 1$.

The bacteria modeled have characteristic lengths on the order of a few microns, and their characteristic velocity is about 10 body lengths per second [34]. In water, this gives them a Reynolds number of

$$\text{Re} \approx \frac{5 \mu m \times 50 \mu m}{10^{-6} \text{m}^2/\text{s}} = 2.5 \times 10^{-4}$$

so $\text{Re} \ll 1$. In such a low Reynolds number regime (called Stokes flow or creeping flow), the inertial terms on the left-hand side of (3.3) can be neglected. The governing equations of Stokes flow are the continuity equation (3.1) and the Stokes equation:

$$\nabla^2 \mathbf{u} - \frac{1}{\mu} \nabla p = 0$$

These equations are linear in $\mathbf{u}$ and $p$. Since the inertial terms are neglected, the velocity and pressure are functions of $x$ alone; there is no time dependence, and the flow properties are transmitted instantaneously. The pressure and vorticity $\omega = \nabla \times \mathbf{u}$ are both harmonic functions.

**Fundamental Solutions of Stokes Flow**

**Solutions Associated with Forces Acting on the Fluid** In the case of a single bacterium swimming at speed $U$, the organism experiences a drag force acting along the swimming direction. It must propel itself (via its flagella) with a thrust equal to the drag exerted on it by the fluid. This means the bacterium can be treated as a
“force-free” particle [49] since the forces that arise from its swimming motion sum to zero.

Before finding the Stokes flow solution for a force-free particle, the Green’s function for a point force must be found. The flow produced by a force \( \mathbf{F} \) acting at a point \( \mathbf{x}_0 \) solves

\[
\nabla^2 \mathbf{u} - \frac{1}{\mu} \nabla p = \mathbf{F} \delta(\mathbf{x} - \mathbf{x}_0)
\]

(3.6)

where \( \delta \) is the 3D Dirac delta function. Assuming that the fluid is still except for the influence of the point force, the boundary conditions require that \( ||\mathbf{u}|| \to 0 \) as \( r \to \infty \). Without loss of generality, take \( \mathbf{F} = F\hat{x} \) and \( \mathbf{x}_0 = 0 \), and write \( r = ||\mathbf{x}|| = \sqrt{x^2 + y^2 + z^2} \). The solution, called a *Stokeslet*, is the most fundamental solution of the Stokes equations. As calculated in [16], the velocity and pressure fields associated with a Stokeslet are

\[
\mathbf{u}_{\text{stok}} = \frac{F}{8\pi \mu} \left[ \left( \frac{1 + x^2}{r^3} \right) \hat{x} + \left( \frac{xy}{r^3} \right) \hat{y} + \left( \frac{xz}{r^3} \right) \hat{z} \right]
\]

(3.7)

\[
p_{\text{stok}} = \frac{F}{4\pi} \frac{x}{r^3}
\]

(3.8)

as shown in figure 3.1.3. Switching to spherical coordinates (where \( \theta \) represents the zenith angle and \( \phi \) represents the azimuthal angle with respect to the \( x \)-axis gives

\[
\mathbf{u}_{\text{stok}} = \frac{F}{8\pi \mu} \left[ \left( \frac{2 \cos \theta}{r} \right) \hat{r} - \left( \frac{\sin \theta}{r} \right) \hat{\theta} + \hat{\phi} \right]
\]

(3.9)

\[
p_{\text{stok}} = \frac{F}{4\pi} \frac{\cos \theta}{r^2}
\]

(3.10)
(a) Streamlines and Flow Speed

(b) Pressure and Contours of Pressure Gradient

Figure 3-1: Velocity and Pressure Fields Around a Stokeslet
The Stokeslet is axisymmetric with respect to the $x$-axis since $u_0 = \frac{\partial u_r}{\partial \phi} = \frac{\partial u_\phi}{\partial \phi} = 0$.

The second fundamental solution of the Stokes equations is called a *Stokes dipole* [54] and it is calculated by taking the dot product of a vector $\alpha$ (where $||\alpha||$ is small) with the gradient of the Stokeslet. The Stokes dipole is often separated into two components, one produced by the symmetric part of the gradient (called a *stresslet*) and the other produced by the antisymmetric part (called a *rotlet*).

\[
\mathbf{u}_d = \alpha \cdot \nabla \mathbf{u}_{\text{stok}} = \mathbf{u}_{\text{str}} + \mathbf{u}_{\text{rot}} \quad (3.11)
\]

\[
\Rightarrow \quad \mathbf{u}_{\text{str}} = \alpha \cdot \frac{1}{2} \left[ \nabla \mathbf{u}_{\text{stok}} + (\nabla \mathbf{u}_{\text{stok}})^T \right] \quad (3.12)
\]

\[
\mathbf{u}_{\text{rot}} = \alpha \cdot \frac{1}{2} \left[ \nabla \mathbf{u}_{\text{stok}} - (\nabla \mathbf{u}_{\text{stok}})^T \right] \quad (3.13)
\]

A stresslet induces a flow field that contains only a radial component, while a rotlet induces a flow field characterized by purely rotational flow.

\[
\mathbf{u}_{\text{str}} = \frac{r}{8\pi\mu} \left[ -\frac{\mathbf{F} \cdot \alpha}{r^3} + \frac{3 (\mathbf{F} \cdot \mathbf{r}) (\mathbf{\alpha} \cdot \mathbf{r})}{r^5} \right] \hat{r} \quad (3.14)
\]

\[
\mathbf{u}_{\text{rot}} = \frac{1}{8\pi\mu} \left( \frac{\alpha \times \mathbf{F}}{r^2} \right) \times \hat{r} \quad (3.15)
\]

When a bacterium swims in a straight line, the drag on its body and the thrust produced by its flagella act along the same axis as each other in opposite directions. The vector separating the points of concentration of these forces also lies along the same axis, so $\alpha$ and $\mathbf{F}$ are parallel and $\alpha \times \mathbf{F} = 0$. Thus $\mathbf{u}_{\text{rot}}$ must be 0 everywhere, so this force configuration only produces a stresslet. Taking $\alpha = \alpha \hat{x}$ and $\mathbf{F} = F \hat{x}$, the velocity and pressure associated with a stresslet at the origin are
\[
\mathbf{u}_{\text{str}} = \frac{F_\alpha}{8\pi\mu r^2} (3\cos^2 \theta - 1) \mathbf{\hat{r}}
\]
\[
\rho_{\text{str}} = \frac{F_\alpha}{4\pi r^3} (1 - 3\cos^2 \theta)
\]

as shown in figure 3-2.

When the bacterium rotates about its center of mass due to an internal torque, it produces a rotlet. The torque it applies to the fluid can be written \( \mathbf{M} = \alpha \times \mathbf{F} \) and this produces the flow

\[
\mathbf{u}_{\text{rot}} = \frac{1}{8\pi\mu} \left( \frac{\mathbf{M}}{r^2} \right) \times \mathbf{\hat{r}}
\]
\[
\rho_{\text{rot}} = 0
\]

as calculated in [4] and shown in figure 3-3.

**Potential Flow Solutions** In potential flow, the fluid is assumed to be inviscid and the flow is irrotational (i.e. the vorticity is zero everywhere). Since

\[
\mathbf{\omega} = \nabla \times \mathbf{u} = 0
\]

and

\[
\nabla \times (\nabla \phi) = 0
\]

for any scalar function \( \phi \), the assignment \( \mathbf{u} = \nabla \phi \) can be made. Inserting this into the
Figure 3-2: Velocity and Pressure Fields Around a Stresslet

(a) Streamlines and Flow Speed

(b) Pressure and Contours of Pressure Gradient
continuity equation (3.1) yields $\nabla^2 \phi = 0$, which implies that any harmonic function $\phi$ corresponds to a potential flow solution. These solutions are useful in Stokes flow because they do not affect the pressure, forces, or torques on the fluid. Specifically, the most useful solutions for the conditions described in this paper are the solutions that are singular at the origin and satisfy the boundary condition $||\mathbf{u}|| \to 0$ as $r \to \infty$. The singularity of the lowest order is the potential dipole, or doublet, which induces the velocity field

$$
\mathbf{u}_{pd} = \frac{c_{pd}}{r^3} \left( 2 \cos \theta \hat{\mathbf{r}} + \sin \theta \hat{\theta} \right)
$$

where $c_{pd}$ is a constant that determines the strength of the doublet. Given the flow produced by one potential singularity, the flow produced by the subsequent one (in
increasing powers of $\frac{1}{r}$ can be found by taking the gradient and multiplying by a tensor. The flow produced by an organism swimming in a straight line is axisymmetric so the only terms of each singularity that are relevant are the terms that arise from taking derivatives in the $x$-direction (where $\theta = \phi = 0$). This gives:

\begin{align*}
\text{dipole: } u_1 &= u_{pd} = \frac{c_1}{r^3} \left(2 \cos \theta \hat{\mathbf{r}} + \sin \theta \hat{\theta}\right) \\
\text{quadrupole: } u_2 &= u_{pq} = \frac{c_2}{r^4} \left\{\frac{1}{2} \left[1 + 3 \cos(2\theta)\right] \hat{\mathbf{r}} + \sin(2\theta) \hat{\theta}\right\} \\
\text{octupole: } u_3 &= u_{po} = \frac{c_3}{r^5} \left\{3 \cos \theta + 5 \cos(3\theta) \hat{\mathbf{r}} + \frac{3}{4} \left[\sin \theta + 5 \sin(3\theta)\right] \hat{\theta}\right\}
\end{align*}

\begin{align*}
\text{16-pole: } u_4 &= u_{p16} = \frac{c_4}{r^6} \left\{\frac{1}{4} \left[9 + 20 \cos(2\theta) + 35 \cos(4\theta)\right] \hat{\mathbf{r}} \\
&\quad + \left[2 \sin(2\theta) + 7 \sin(4\theta)\right] \hat{\theta}\right\} \\
\text{32-pole: } u_5 &= u_{p32} = \frac{c_5}{r^7} \left\{\cos \theta \left[29 - 28 \cos(2\theta) + 63 \cos(4\theta)\right] \hat{\mathbf{r}} \\
&\quad + \frac{5}{4} \left[2 \sin \theta + 7 \sin(3\theta) + 21 \sin(5\theta)\right] \hat{\theta}\right\}
\end{align*}

\begin{align*}
\text{64-pole: } u_6 &= u_{p64} = \frac{c_6}{r^8} \left\{\frac{1}{2} \left[50 + 105 \cos(2\theta) + 126 \cos(4\theta) + 231 \cos(6\theta)\right] \hat{\mathbf{r}} \\
&\quad + 3 \left[5 \sin(2\theta) + 12 \sin(4\theta) + 33 \sin(6\theta)\right] \hat{\theta}\right\}
\end{align*}

The flow fields induced by these six singularities are shown in figure 3-4.

66
Figure 3-4: Streamlines and Magnitude of Velocity in Flow Due to Potential Singularities
3.2 Modeling and Simulation Details

3.2.1 Model Characteristics

Model for Individual Bacterium

**Bacteria Geometry** Although different types of bacteria have shown collective motion behaviors, the most-studied species (*E. coli* and *B. subtilis*) are morphologically similar [51]. *P. mirabilis* resembles these species as well. Each bacterium has a rod-shaped cell body about 4 μm long and slightly less than 1 μm in diameter. It is propelled by multiple flagella which bundle together into a single rotating helix as the organism swims forward. The flagella range in length from 10 to 15 μm[11].

![Figure 3-5: Geometry of a Bacterium](image)

**Far-field Flow** In the far field, the flow produced by the bacterium looks like a stresslet. The force separation value $\alpha$ is determined by the geometry of the bacterium itself (figure 3-5). The flagella are very thin filaments, so the drag force on them is negligible compared with the drag on the body and the drag force is concentrated at the center of mass of the body. Since the flagella rotate as a rigid helix at a constant angular speed, the thrust comes from a force distribution along their length $L_f$; as an approximation, the total thrust is taken to be concentrated at the center of the distribution, or halfway along the length of the flagella. If the body is modeled as an ellipse with major axis $a$ and minor axis $b$ and the flagella have length $L_f$, then
\[ \alpha = a + \frac{L_f}{2} \simeq 4a \] (3.29)

For simplicity, the body is modeled as a sphere instead of an ellipsoid without changing the center of mass or the volume.

\[
\text{sphere volume: } V_{\text{sph}} = \frac{4}{3} \pi R^3 \\
\text{ellipsoid volume: } V_{\text{ell}} = \frac{4}{3} \pi ab^2
\]

\[ V_{\text{sph}} = V_{\text{ell}} \Rightarrow a = \sqrt[3]{4R} \] (3.30)
\[ \Rightarrow \alpha = 4\sqrt[3]{4R} \] (3.31)

When a sphere of radius \( R \) moves through fluid at velocity \( \mathbf{U} \) in a low \( \text{Re} \) regime, the drag force exerted on it equals \(-6\pi \mu RU\). In the absence of inertia, a self-propelled sphere must provide a thrust force of \(+6\pi \mu RU\) to move forward at velocity \( \mathbf{U} \). Plugging \( F = 6\pi \mu R||\mathbf{U}|| \) and \( \alpha = 4\sqrt[3]{4R} \) into (3.16) and (3.17) gives the far-field flow for a sphere at the origin moving along the \( x \)-axis at speed \( U = ||\mathbf{U}|| \).

\[
\mathbf{u}_f(x) = 3\sqrt[3]{4} \ U \ \frac{R^2}{r'^2} \ (3 \cos^2 \theta' - 1) \ \hat{r}' \\
p_{f}(x) = 6\sqrt[3]{4} \ \mu \ U \ \frac{R^2}{r'^3} \ (1 - 3 \cos^2 \theta')
\] (3.32) (3.33)

Here, \( r' \) and \( \theta' \) are the distance and relative angle to \( x \) from the center of the stresslet \( \mathbf{x}_c = -2\sqrt[3]{4} \mathbf{x} \) and \( \hat{r}' \) is the unit vector pointing from \( \mathbf{x}_c \) to \( x \).
Regularization  Since the center of the stresslet at $x_c$ is outside the surface of the sphere, a singularity is created at this point with $||u_f(x_c)|| \to \infty$. Cortez introduces the method of regularized Stokeslets to handle the similar situation of singularities placed directly on a boundary [7]. Instead of deriving the Stokeslet flow by starting with a Dirac-delta point force, a "blob" force is used. Its distribution is described by the blob function $\phi^\varepsilon(x)$, a radially-symmetric function that reaches a finite maximum at $x = 0$. Like the Dirac-delta function, $\phi^\varepsilon$ is normalized so that the area (or volume in three dimensions) under the curve equals 1. The small-valued parameter $\varepsilon$ controls the spread of $\phi^\varepsilon$ as if the force is distributed over a ball of radius $\varepsilon$. Also, the blob function is defined such that $\lim_{\varepsilon \to 0} \phi^\varepsilon(x) = \delta(x)$. The new governing Stokeslet equation (replacing (3.6)) is

$$\nabla^2 u - \frac{1}{\mu} \nabla p = F \phi^\varepsilon(x - x_0) \quad (3.34)$$

where

$$\phi^\varepsilon(x) = \frac{15 \varepsilon^4}{8\pi (||x||^2 + \varepsilon^2)^{7/2}} \quad (3.35)$$

as used in [8]. Note that other choices of $\phi^\varepsilon(x)$ are equally valid, but this form is chosen for ease of computation. If the solution to (3.34) is labelled $u^\varepsilon_{stok}$ and the stresslet flow is derived in the same way as before,

$$u^\varepsilon_{str} = \alpha \cdot \frac{1}{2} \left[ \nabla u^\varepsilon_{stok} + (\nabla u^\varepsilon_{stok})^T \right] \quad (3.36)$$

replaces (3.12). The regularized stresslet solution is
\[ u_{str}^r(x) = \frac{F\alpha}{8\pi\mu} \left\{ \left[ \frac{r}{(r^2 + \epsilon^2)^{3/2}} (3\cos^2 \theta - 1) \right] \hat{r} \right. \]
\[ \left. - \left[ \frac{3\epsilon^2 r}{(r^2 + \epsilon^2)^{5/2}} \sin \theta \cos \theta \right] \hat{\theta} \right\} \quad (3.37) \]

\[ p_{str}^r(x) = \frac{F\alpha}{8\pi} \left[ \frac{2r^2 (2\cos^2 \theta + 1) + 5\epsilon^2}{(r^2 + \epsilon^2)^{5/2}} - \frac{5r^2 \cos^2 \theta (2r^2 + 5\epsilon^2)}{(r^2 + \epsilon^2)^{7/2}} \right] \quad (3.38) \]

so the regularized far-field flow for a bacterium is

\[ u_{\text{ff}}^r(x) = 3\sqrt{4} \, UR^2 \left\{ \left[ \frac{r'}{(r'^2 + \epsilon^2)^{3/2}} (3\cos^2 \theta' - 1) \right] \hat{r}' \right. \]
\[ \left. - \left[ \frac{3\epsilon^2 r'}{(r'^2 + \epsilon^2)^{5/2}} \sin \theta' \cos \theta' \right] \hat{\theta}' \right\} \quad (3.39) \]

\[ p_{\text{ff}}^r(x) = 3\sqrt{4} \, \mu \, UR^2 \left[ \frac{2r'^2 (2\cos^2 \theta' + 1) + 5\epsilon^2}{(r'^2 + \epsilon^2)^{5/2}} - \frac{5r'^2 \cos^2 \theta' (2r'^2 + 5\epsilon^2)}{(r'^2 + \epsilon^2)^{7/2}} \right] \quad (3.40) \]

with \( r' \) and \( \theta' \) defined as in (3.32) and (3.33). Note that taking the limit of (3.39) and (3.40) as \( \epsilon \to 0 \) recovers (3.32) and (3.33). When \( \epsilon > 0 \), some errors are introduced into the flow field near the singularity, but magnitude of these errors is very small. Figure 3-6 shows the effect of different values of \( \epsilon \) on the magnitude of the velocity. Small values of \( \epsilon \) result in a field \( u_{\text{ff}} \) that matches the singular flow field more closely, but also has higher maximum velocities near \( x_c \). Slightly larger values of \( \epsilon \) yield lower maximum velocities, but the flow diverges more from the singular field at small distances from \( x_c \). For the purposes of simulation, \( \epsilon \) is taken to be 0.1.

**Near-field Flow**  In the near field, the model must account for the boundary conditions on the bacterium surface. Fluid cannot pass through the surface due to the no-penetration condition, and since the flow is viscous, fluid also cannot move tan-
(a) Contours of Constant Flow Speed

(b) Value of Flow Speed Along x-axis

Figure 3-6: Effect of Changing $\epsilon$ on Value of $||u||$ Near $x_c$
gentially along the surface due to the no-slip condition [16]. This means that at every point near the surface, the fluid must have the same velocity as the surface itself. The bacterium is moving forward with a speed $U$, so on the surface defined by $r = R$, the prescribed boundary conditions are

$$u_p(\theta) = (U \cos \theta) \mathbf{\hat{r}} + (-U \sin \theta) \mathbf{\hat{\theta}}$$  \hspace{1cm} (3.41)$$

Calculating the flow on the surface given by the far-field stresslet model reveals that this condition is not satisfied by the stresslet alone. To quantify the discrepancy between the prescribed flow $u_p$ on the boundary and the flow $u_s$ induced on the boundary by the far-field stresslet plus any correction terms, a scalar error variable $\tilde{u}_1$ is introduced. It is defined as follows:

$$\tilde{u}_1 = \frac{\int_0^{2\pi} ||u_p - u_s||_r=R d\theta}{\int_0^{2\pi} ||u_p|| d\theta}$$  \hspace{1cm} (3.42)$$

The normalization term in the denominator is the “error” that would arise if $u_s = 0$ on the surface. Using the stresslet alone (3.32) yields $\tilde{u}_1 = 0.4159$. To lower $\tilde{u}_1$, potential singularities are added, which is possible because the potential flow solutions do not change the overall characteristics of the flow. Adding $N$ potential singularities of increasing order gives

$$u_s(x) = u_s(x) + \sum_{i=1}^{N} u_i(x|x_i)$$  \hspace{1cm} (3.43)$$

where $u_i$ is defined as in §3.1.3 and $x_i$ is the position of the singularity along the $x$-axis. By summing up the singularities as such, the flow field can be “tuned” by changing the set of singularity strengths $\{c_i\}$ (with $i = 1, 2, \ldots, 6$) and/or the set of singularity positions $\{x_i\}$. The goal is to find the constants that minimize the
error $\tilde{u}_1$ as calculated in (3.42). A secondary goal is to minimize the error using the smallest necessary number of singularities; if an additional term will only reduce the error by a small amount, then it should be neglected to reduce computational complexity. Using Matlab’s optimization toolbox, a few different combinations of singularities are tested.

First, one singularity of each order is placed at the center of the bacterium’s body, so that $x_i = 0 \forall i$. These additional terms allow $\tilde{u}_1$ to be minimized over the set $\{c_i\}$ with

$$\mathbf{u}_s(x) = \mathbf{u}_{\tilde{u}}(x) + \sum_{i=1}^{N} \mathbf{u}_i(x|0)$$  \hspace{1cm} (3.44)

for $N = 1, \ldots, 6$. The values of $\{c_i\}$ that minimize $\tilde{u}_1$ are shown in figure 3-7.

For the next set of trials, only potential dipoles are used, but instead of their positions being fixed at the center of the sphere, the positions are optimized as well. With $N$ dipoles, $\tilde{u}_1$ is minimized over $2N$ parameters defined by the set $\{c_i\} \cup \{x_j\}$. The flow on the surface is:

$$\mathbf{u}_s(x) = \mathbf{u}_{\tilde{u}}(x) + \sum_{j=1}^{N} \mathbf{u}_i(x|x_j)$$  \hspace{1cm} (3.45)

Two more sets of trials are run, one using only potential quadrupoles with

$$\mathbf{u}_s(x) = \mathbf{u}_{\tilde{u}}(x) + \sum_{k=1}^{N} \mathbf{u}_2(x|x_k)$$  \hspace{1cm} (3.46)

and the other using both dipoles and quadrupoles, with

$$\mathbf{u}_s(x) = \mathbf{u}_{\tilde{u}}(x) + \sum_{m=1}^{N/2} \mathbf{u}_1(x|x_m) + \sum_{n=1}^{N/2} \mathbf{u}_2(x|x_n)$$  \hspace{1cm} (3.47)
The resulting values of \( \tilde{u}_1 \) for all the trials are shown in figure 3-8.

The first observation made based on this data is that in the cases where more singularities of the same type are added, the error does not decrease significantly (if it decreases at all). Since the positions and strengths of the singularities are optimized independently from one another, so at times the "optimal" solution involves two or more dipoles (or quadrupoles) at the same position with strengths that sum to 0. Another observation is that the smallest value of \( \tilde{u}_1 \) found with this method is 0.3730, which is only a 10% drop from the baseline value with no potential singularities. The optimization methods used are probably getting trapped in local minima instead of finding the global minima.

Through trial-and-error experimentation, a singularity configuration is found with two
potential dipoles that reduce $\tilde{u}_1$ to 0.0858. The dipoles have strengths $c_1 = -42.3$ and $c_2 = 137$ and positions $x_1 = -7690$ and $x_2 = 8.62$. The flow field produced when these dipoles are superimposed on the stresslet is shown in figure 3-9. However, while this solution might be optimal for matching the prescribed boundary conditions, the region of fast-moving fluid that it creates in front of the bacterium is not physically feasible.

Instead, the model chosen for use in the simulations is the superposition of the stresslet and a single potential dipole at its optimized position within the cell body ($c_1 = 0.102$, $x_1 = 0.225$). In this case, $\tilde{u}_1 = 0.3748$, which is a 9.8% reduction relative to the value with no potential singularities and is only slightly higher than the minimal error that arises from using more singularities but keeping them confined to lie within the body.
Figure 3-9: Flow Field Produced by Two Potential Dipoles Superimposed on Stresslet ($c_1^1 = -42.3$, $c_2^1 = 137$, $x_1^1 = -7690$, and $x_1^2 = 8.62 \Rightarrow \bar{u}_1 = 0.0858$)

This flow field, shown in figure 3-10, is

$$u_{nf}(x) = u_{fl}(x) + u_1(x)|_{x1=0.225}|_{c1=0.102}$$  \hspace{1cm} (3.48)

**Rotation**  As mentioned in §3.1.3, a rotating bacterium produces a rotlet. As an organism modeled as a sphere of radius $R$ rotates in the $r\theta$-plane with an angular velocity $\Omega = \Omega \hat{\phi}$, it creates the flow

$$u_{rot}(x) = \Omega \frac{R^3}{r^2} \hat{\theta}$$  \hspace{1cm} (3.49)
Figure 3-10: Flow Field Produced by Potential Dipole Superimposed on Stresslet as Used in Simulations ($c_1 = 0.102$ and $x_1 = 0.225 \Rightarrow \vec{u}_1 = 0.3748$)

where $r$ and $\theta$ are measured relative to the origin. It should be noted that this equation automatically satisfies the boundary conditions of rigid-body rotation because $u_{rot}|_{r=R} = R (\Omega \times r) = R\Omega \hat{\theta}$.

**Bacteria Behavior and Interactions**

As a bacterium swims through still water, it creates a flow field that advects any objects in the fluid, whether they are passive tracer particles or other bacteria. A two-dimensional “slice” of fluid containing many swimming organisms will thus exhibit a complicated flow pattern, because the flow produced by each organism advects every other organism. The Stokes equations are linear so the net flow at a given point $x$ can be calculated by summing the contributions from all the bacteria. For the purpose
of modeling, the flow is assumed to be locally uniform near each bacterium, with a
velocity equal to the velocity that the fluid would have at the bacterium’s center if
that bacterium were not there. The advective induced flow for organism \( i \) is

\[
\mathbf{u}_{\text{ind}}^{i}(\mathbf{x}^{i}) = \sum_{j \neq i} \mathbf{u}_{nf}^{j}(\mathbf{x}^{i})
\]  

(3.50)

where \( \mathbf{u}_{nf} \) is defined as in (3.48). This superposition neglects the effects of composite
boundary conditions, introducing an error which is calculated in §3.2.3. Each organism
swims continuously with a constant magnitude thrust \( F_T = 6\pi \mu R U_T \). By the
Stokes force law [54], this implies that each organism swims at a constant speed \( U_T \)
relative to the fluid.

The forces acting on each organism are thrust and drag; they must sum to zero
because inertia is neglected in Stokes flow and acceleration is not possible. Balancing
the forces for organism \( i \) gives:

\[
\mathbf{F}_T^i + \mathbf{F}_D^i = 0 \Rightarrow 6\pi \mu R \mathbf{U}_T^i - 6\pi \mu R \mathbf{U}_D^i = 0
\]

\[
\Rightarrow \mathbf{U}_T^i = \mathbf{U}_D^i
\]

(3.51)

where \( \mathbf{U}_D^i \) is the velocity of the organism in its own reference frame. If \( \mathbf{x}^{i} \) is the or-
ganism’s position in the lab reference frame, then a change of coordinates for velocity
is accomplished by taking:

\[
\dot{\mathbf{x}}^i = \mathbf{U}_T^i + \mathbf{u}_{\text{ind}}^i
\]

(3.52)

This equation can be integrated to find the trajectory of each bacterium through
time. However, first, the direction of \( \mathbf{U}_T^i \) must be specified. Just as the forces on each
bacterium must sum to zero, the net torque must equal zero as well. The viscosity of the fluid exerts a torque $L_v$ on the cell body which competes with the bacterium's desire to swim in a certain direction. As stated in [29], "such physiological orientation processes can be characterized as equivalent to an external torque which can be added to any other external torques." If $U^i_T = U_T \hat{p}^i$ and the bacterium wants to swim in direction $\hat{k}^i$, the simplest model for its internal couple is $L = L_0 \hat{p} \times \hat{k}$. Setting $L + L_v = 0$ and simplifying gives the reorientation equation:

$$\frac{d\hat{p}}{dt} = \frac{1}{2B} \left[ \hat{k} - \left( \hat{k} \cdot \hat{p} \right) \hat{p} \right] + \frac{1}{2} \omega \times \hat{p} \tag{3.53}$$

Note that in [29], the body is assumed to be an ellipsoid, so additional terms from [20] are included. By modeling the body as a sphere, these terms drop out. The constant $B$ is a timescale of reorientation. If $B$ is large, $\hat{p}$ will approach $\hat{k}$ more slowly than if $B$ is small. Experiments show that for gyrotactic organisms, $B \sim 1$ sec [28]; for the purposes of this model, $B$ is considered a tuneable parameter, where possible values are chosen to keep $B \sim O(1)$. The vector $\omega$ is the local induced vorticity and the only non-zero component of the vorticity points in the $\hat{\phi}$ direction. Because of the linear property of the Stokes equations, the local induced vorticity is:

$$\omega = \omega \hat{\phi} = \nabla \times u_{\text{ind}} \tag{3.54}$$

The only information that bacteria have to base their decisions on is the properties of the flow where they are, since they do not have any distance sensors. Within this framework, seven possible control schemes are proposed for the choice of $\hat{k}$.

1. $\hat{k}_1 = \hat{p}$ : The bacteria continue swimming in the same direction (subject to the effects of the viscous couple).

2. $\hat{k}_2 = \frac{u_{\text{ind}}}{|u_{\text{ind}}|} :$ The bacteria try to align with the direction of the local induced flow.
3. \( \hat{k}_3 = \pm \frac{\mathbf{u}_{\text{ind}}}{||\mathbf{u}_{\text{ind}}||} \): The bacteria try to align with the local induced flow, but they will swim either with the flow or against it by choosing the direction closest to their current orientation. (In the full ellipsoid model, this could be a method of reducing drag.)

4. \( \hat{k}_4 = \frac{\nabla p}{||\nabla p||} \): The bacteria sense the local pressure gradient and swim up it.

5. \( \hat{k}_5 = -\frac{\nabla p}{||\nabla p||} \): The bacteria sense the local pressure gradient and swim down it.

6. \( \hat{k}_6 = \frac{\nabla \omega_\theta}{||\nabla \omega_\theta||} \): The bacteria sense the local vorticity gradient and swim up it.

7. \( \hat{k}_7 = -\frac{\nabla \omega_\theta}{||\nabla \omega_\theta||} \): The bacteria sense the local vorticity gradient and swim down it.

Let \( \phi \) be the angle of the unit vector \( \mathbf{p} \). The angular velocity \( \Omega = \dot{\phi} \) can be calculated from (3.53). As bacterium \( i \) rotates at an instantaneous angular velocity \( \Omega^i \), it creates a rotlet (3.49). This flow must be included in the summation in (3.50) so the total induced flow is written

\[
\mathbf{u}_{\text{ind}}^i(x^i) = \sum_{j \neq i} \left[ \mathbf{u}_{\text{nf}}^j(x^i) + \Omega^j \frac{R^3}{r_{ij}^2} \hat{\theta} \right] \quad (3.55)
\]

with \( r_{ij} = ||x^j - x^i|| \) and \( \hat{\theta} \) in the direction of the line connecting \( x^i \) and \( x^j \).

### 3.2.2 Simulation Characteristics

To find the trajectories of a group of \( N \) bacteria as they swim through a two-dimensional domain, equations (3.52) and (3.53) must be integrated through time. All variables and parameters are nondimensionalized with respect to the bacterium radius \( R \) and the forward velocity \( U_T \), so for example the time is multiplied by \( U_T / R \) to make it dimensionless. Organism \( i \) (where \( i = 1, 2, \ldots, N \) ) has two state variables, \( x^i \) and \( \mathbf{p}^i \). The initial values of all the states are defined randomly. The simulation
starts at \( t = 0 \) and runs until \( t = t_{\text{final}} \) while being updated in discrete time steps defined by the parameter \( \delta t \). At each time step, the time derivatives of \( x^i \) and \( \hat{p}^i \) are calculated as defined earlier, and then the states are updated.

\[
x^i(t) = x^i(t-1) + \dot{x}^i \delta t
\]
\[
= x^i(t-1) + \left( u^i_{\text{ind}} + U_T \hat{p}^i \right) \delta t
\]  
(3.56)

\[
\hat{p}^i(t) = \hat{p}^i(t-1) + \frac{d\hat{p}^i}{dt} \delta t
\]
\[
= \hat{p}^i(t-1) + \left\{ \frac{1}{2B} \left[ \hat{k}^i - \left( \hat{k}^i \cdot \hat{p}^i \right) \hat{p}^i \right] + \frac{1}{2} \omega^i \times \hat{p}^i \right\} \delta t
\]  
(3.57)

To track \( \hat{p}^i \), the simulation finds the angle \( \varphi^i \) it makes with the \( x \)-axis; similarly, \( \Psi^i \) denotes the angle between the “desired” orientation vector \( \hat{k}^i \) and the \( x \)-axis. Through algebraic manipulation of (3.57), the update equation for \( \varphi^i \) is found.

\[
\varphi^i(t) = \tan^{-1}
\left\{ \frac{2Bc\varphi^i(t-1) + [c\Psi^i - c(\Psi^i - \varphi^i(t-1))]c\varphi^i(t-1) - B\omega^i s \varphi^i(t-1)}{2Bs\varphi^i(t-1) + [s\Psi^i - c(\Psi^i - \varphi^i(t-1))]s \varphi^i(t-1) + B\omega^i c \varphi^i(t-1)} \right\}
\]  
(3.58)

In the interest of space, \( c \) and \( s \) represent \( \cos \) and \( \sin \) respectively. At \( t = t_{\text{final}} \), the simulation stops running and stores the values of \( x \) and \( \varphi \) for all bacteria at all times so the data can be plotted and analyzed.
3.2.3 Self-Validation Tests and Internal Error

A lot of simplifications and assumptions have been made in the modeling process, so its accuracy must be brought into question and measured by a series of tests. The methodology is to compare one set of simulation results to another to derive measures of relative internal error. By changing settings and adjusting parameters, this error can be quantified, and the optimal values can be chosen to strike a balance between minimizing both the error and the computational complexity.

**Number of Tiles**

To keep the concentration of bacteria constant as the organisms swim around in a square domain with side length \( L \), a periodic domain is implemented. The exact solution requires an infinite number of repeats in each direction, but this is not feasible to simulate, so the first parameter tested is the number of "tiles" in the full domain. Two simple cases are examined to provide insight into how changing the number of tiles affects the accuracy of the results.

![Figure 3-11: Initial Conditions for Test Cases 1 (left) and 2 (right)](image-url)
Test case #1: One Bacterium One bacterium is initialized at the left of the domain moving to the right with a speed $U_T$ (figure 3-11). In the absence of any other bacteria, it would continue to move at this speed. However, when neighboring bacteria are introduced through the addition of extra tiles to the periodic domain, the bacterium slows down (in the reference frame of the lab) and instantaneously reaches a steady-state speed $U_{ss}$. By using numerical curve-fitting techniques, the dependence of $U_{ss}$ on $n$ and $L$ can be quantified by the following equation:

$$\frac{U_{ss}}{U_T} = 1 + \left(\frac{7.45}{L}\right)^3 \left(n^{-0.675} - 1\right)$$

The $R^2$ value for each fit is 0.99 and the results are shown in figure 3-12(a). By taking the limit of (3.59) as $n \to \infty$, the steady-state speed with an infinite number of tiles ($U_{inf}$) is extrapolated.

$$\frac{U_{inf}}{U_T} = \lim_{n \to \infty} \frac{U_{ss}}{U_T} = 1 - \left(\frac{7.45}{L}\right)^3$$

The error accrued by using finite values of $n$ is calculated by comparing $U_{ss}$ to $U_{inf}$ (figure 3-12(b)).

Test case #2: Two Bacteria In case #1, because of the symmetrical arrangement and the structure of the flow fields surrounding each bacterium, the contributions of the stresslet components cancel each other out. This means that the observed variations in $\frac{U_{ss}}{U_T}$ are entirely due to the potential dipole components of the flow fields, so an additional, slightly more complicated case is needed to test the effect of varying $n$. For the second case, two bacteria are used to break the symmetry. Both start on the left of the domain moving to the right at a speed $U_T$; one is initialized at a distance $L/4$ from the bottom, and the other at a distance $3L/4$ (figure 3-11). With an infinite number of tiles, the bacteria would continue to swim horizontally because of symmetry. However, with finite $n$, they deflect vertically as they travel across the
(a) Dependence of Normalized Steady-State Speed on Domain Parameters

(b) Dependence of Error in Steady-State Speed on Domain Parameters

Figure 3-12: Dependence of Speed and Error on $L$ and $n$ in Case #1
domain. This deflection $|\Delta y|$ is measured over the time it takes to cross the domain. Values of $L < 60$ are omitted from these trials because the bacteria tend to deflect far enough from the straight path that they hit the top or bottom of the domain before reaching the opposite side.

Since $\lim_{n \to \infty} |\Delta y| = 0$, $|\Delta y|$ itself can be used as a measure of the error that arises from using a finite number of tiles, and its dependence on $L$ and $n$ is shown in figure 3-13.

![Figure 3-13: Dependence of Vertical Deflection on $L$ and $n$ in Case #2](image)

**Choice of Minimum Domain Length and Number of Tiles** Using the error values displayed in figures 3-12(b) and 3-13, the parameters chosen for use in simulations with low bacteria concentrations are $L \geq 100$ and $n = 25$, giving maximum errors of $5.76 \times 10^{-5}$ in case 1 and $0.0127$ in case 2. For higher concentrations, a
smaller domain is used with more tiles, letting $L = 60$ and $n = 49$.

Superposition: Effect on Boundary Conditions

The second self-validation test is intended to examine the assumption of pure linearity in superimposing the flow fields produced by each organism. This superposition ignores the boundary conditions on their surfaces. Solving the full problem here proves to be extremely difficult, even with two organisms, so the true answer for the flow field surrounding two swimming bacteria is not calculated. However, the correct boundary conditions are prescribed by the problem, so the flow calculated at the surface of each organism can be compared with the conditions that would be prescribed in the full solution. Specifically, the flow field produced by one bacterium should have zero magnitude on the surface of all other bacteria. To test this, one organism is placed at the origin aligned with the $x$-axis. A second organism is placed at various positions in the plane and $u_2$ is calculated on its surface as:

$$\tilde{u}_2 = \frac{1}{2\pi} \int_0^{2\pi} \left| u_{|r=R|} \right| d\theta$$

(3.61)

where $r$ and $\theta$ are relative to the center of the second bacterium and $2\pi$ is used as a normalization constant because it is the value of the denominator in (3.42). The error $\tilde{u}_2$ is quantified in a similar manner to $u_1$ in §3.2.1. The value of $\tilde{u}_2$ as it depends on the position of the center of the second bacterium is shown in figure 3-14.

The error is only calculated when the separation between the two centers is larger than $2R$; otherwise the bacteria would be in a collision state. As expected, the largest errors arise when one bacterium swims directly behind another; this is not likely to occur in real bacteria because the trailing organism would collide with the other’s flagella. Along the $y$-axis, $\tilde{u}_2 < 0.5$, and it is of the same order of magnitude as $\tilde{u}_1$. 87
Bacterium Shape

In the model presented here, each bacterium is assumed to be a sphere, but the real organisms are elongated rods or ellipsoids. The change in shape does not affect the far-field flow, but the significance of the shape assumption on the near-field flow must be determined. The near-field flow caused by a stresslet with an ellipsoidal surface surrounding one of the forces is a complicated problem so before attempting to solve it, the simpler case of an ellipsoid in uniform flow is examined. The dominant component of this flow is a Stokeslet which decays like \( \frac{1}{r} \), whereas the stresslet decays like \( \frac{1}{r^2} \), so the “shape” effect of an ellipsoid will be even smaller in the force-free case than in the point force case.

Uniform flow over a sphere is characterized by a point force and a potential dipole; flow over an ellipsoid is characterized by a distribution of point forces and potential dipoles spread over the line connecting the foci of the body [5]. If the background flow is written \( \mathbf{U} = U_x \hat{x} + U_y \hat{y} \), then the flow field around a sphere at the origin is

\[
\mathbf{u}_{\text{sphere}}(\mathbf{x}) = U_x \hat{x} + U_y \hat{y} - [\alpha_1 \mathbf{u}_{\text{str}}(\mathbf{x} | \hat{x}) + \alpha_2 \mathbf{u}_{\text{str}}(\mathbf{x} | \hat{y})] \\
+ [\beta_1 \mathbf{u}_{\text{pd}}(\mathbf{x} | \hat{x}) + \beta_2 \mathbf{u}_{\text{pd}}(\mathbf{x} | \hat{y})]
\]  

(3.62)
with constants $\alpha_1$, $\alpha_2$, $\beta_1$, and $\beta_2$ determined by boundary conditions. The second argument of each singularity velocity term is a unit vector defining the orientation of the singularity. For a prolate ellipsoid with its major axis aligned with the $x$-axis and its foci at $\pm c\hat{x}$, the flow field around it is

$$u_{\text{ellipsoid}}(x) = U_x \hat{x} + U_y \hat{y} - \int_{-c}^{c} \left[ \alpha_1 u_{\text{str}}(x - \xi \hat{x}|\hat{x}) + \alpha_2 u_{\text{str}}(x - \xi \hat{x}|\hat{y}) \right] d\xi$$

$$+ \int_{-c}^{c} \left( c^2 - \xi^2 \right) \left[ \beta_1 u_{pd}(x - \xi \hat{x}|\hat{x}) + \beta_2 u_{pd}(x - \xi \hat{x}|\hat{y}) \right] d\xi \quad (3.63)$$

where the constants are again determined by the boundary conditions. The average aspect ratio of the bacteria described in §3.2.1 is $\frac{a}{b} = 4$. To compare an ellipsoid with this aspect ratio to a sphere of equal volume (as in (3.30)) and radius 1,

$$c = a \sqrt{1 - \left( \frac{b}{a} \right)^2} = \sqrt{4 \left( 1 \right)} \sqrt{1 - 0.25^2} = 1.537 \quad (3.64)$$

Two cases are tested: uniform flow in the $x$-direction ($U_y = 0$) and uniform flow in the $y$-direction ($U_x = 0$). The flow fields produced in these 2 cases are shown in figure 3-15.

To find the differences between the flows around the sphere and the ellipsoid, cross-sections are taken along the $x$- and $y$-axes, and the “error” that arises from using the spherical model is calculated as

$$\tilde{u}_3 = \frac{\|u_{\text{sphere}}\| - \|u_{\text{ellipsoid}}\|}{\|U\|} \quad (3.65)$$

and is plotted in figure 3-16. When the flow is calculated at a distance $3R$ from the center of an organism (which corresponds to a separation of only $1R$ between two
organisms), the maximum value of $\tilde{u}_3$ is 0.2. This maximum only occurs if the flow is exactly perpendicular to the major axis of the bacterium; if it is at an angle, $\tilde{u}_3$ drops significantly. Also, as the separation distance increases, $\tilde{u}_3$ falls off rapidly. Since the velocity field produced by a body in uniform flow is characterized by slower decay rates with respect to distance than the field produced by a stresslet, $\tilde{u}_3$ would be even smaller in the force-free case. This means that the line distributions of singularities can be approximated as point singularities by modeling each bacterium as a sphere without inducing significant errors.
3.3 Simulations and Results

The full simulation presented here has three parameters which could change the system behavior significantly: the control scheme determined by $\hat{k}$, the timescale of reorientation $B$, and the concentration of bacteria which is related to the size of the domain (set by $L$) and the number of bacteria $N$. Many of the experimental results discussed earlier find that collective motion starts once the bacterial concentration is increased above some critical value; few papers quantify the critical value, but most find evidence of collective behavior when the volume fraction is at least 0.01 or 1%. The simulation will thus concentrate on combinations of $N$ and $L$ that yield volume fractions above this value.
Specifically, if the total domain is considered to be a square sheet with nondimensionalized thickness $2$, it has a volume $V_{\text{tot}} = 2L^2$. A group of $N$ bacteria have the volume $V_{\text{bact}} = \frac{4}{3}\pi N$. This gives the volume ratio

$$v = \frac{2\pi N}{3L^2}$$  \hspace{1cm} (3.66)

so if $v > 0.01$, then the condition on the number of bacteria is

$$N > 0.00477L^2$$  \hspace{1cm} (3.67)

to ensure that the simulation operates in the range where collective motion is observed. If $L = 60$, $N \geq 18$; if $L = 100$, $N \geq 48$.

### 3.3.1 Pattern Simulations

**Motivation**

Before running simulations with such large numbers of bacteria, simple “pattern” cases are tested to give insight into how the model behaves under different conditions. The patterns are chosen based on the observed experimental behaviors at low Reynolds numbers.

**Pattern 1: Ring Configuration**

The first pattern involves bacteria starting in evenly-spaced positions around a circular ring. Their orientations are initialized so that their velocity vectors are tangent to the circle. In [36], Riedel et al. find spermatozoa swimming in rings; this configuration is also the simplest case of the general circular motion observed in many
other circumstances. The goal of running this set of simulations is to see whether any combinations of $B$ and $\hat{k}$ lead to continuous circular motion. The most promising results come from setting $\hat{k} = \hat{k}_4$, where the bacteria try to swim up local pressure gradients. The bacteria follow spiral or circular paths until the radius of the circle reaches a certain value dependent on $N$ and $B$; sometimes they continue circling at this radius for a few complete cycles before the circle breaks up, while other times they only complete a partial cycle. With $N = 6$, the simulation is run with different values of $B$ and the results are shown in figure 3-17. Smaller values of $B$ lead to random bacterial motions throughout the trial; larger values of $B$ force the bacteria to swim away from each other before they can be entrapped by their neighbors into a circular path.

The simulation trial run with $B = 0.4365 = 10^{-0.36}$ is examined in more detail in figure 3-18. Here, the pressure field is plotted at various points in time. These images reveal that the pressure gradient lines are not circular, as one might expect given the resulting behavior. Instead, each bacterium swims towards the high-pressure region of the flow field produced by the bacterium in front of it, and when a group of them swim in this way at the same time, it leads to circular motion. The speeds of each organism in a global reference frame are greater than 1; depending on other parameters, the speeds range from 1.05 to 1.5. While these results are encouraging, they are highly dependent on the symmetry of the problem. As soon as the symmetry is broken, the bacteria stop swimming in circular paths as seen in figure 3-18(d).

**Pattern 2: Simple Disc Configuration**

The bacteria *B. subtilis* and *E. coli* are not observed in rings; instead they are seen in large vortices that resemble discs of organisms rotating as a group. The third pattern tested in simulation is a simple approximation of such a disc. Eight bacteria are used with half of them placed in each of two concentric rings. From the center, the bacteria in the inner ring are placed at angles $\pi/4$, $3\pi/4$, $5\pi/4$, and $7\pi/4$; the bacteria in the
Figure 3-17: Paths of 6 Bacteria in Ring Configuration Swimming Up Pressure Gradients
outer ring are placed at angles $\pi/2$, $\pi$, $3\pi/2$, and $2\pi$. As with the ring configuration, the initial velocities of each bacterium lie tangent to the circle at their position, and the goal of the disc simulations is to see if any combination of parameters will lead to collective circular motion.

Again, the pressure gradients come closest to producing circular paths. When the bacteria swim up pressure gradients ($\hat{k} = \hat{k}_4$), the inner bacteria stay in a ring formation, while the outer bacteria swim away from each other and the small ring. However, when the bacteria swim down pressure gradients ($\hat{k} = \hat{k}_5$), they hold the disc formation (given certain values of $B$). The disc grows and spreads out as it...
rotates, but if many rotating discs sit close to each other, their growth might be restricted. Figure 3-19 shows snapshots in time of the bacteria's positions as well as the pressure field and gradients.

Figure 3-19: Pressure Field and Gradients with 8 Bacteria in Disc Configuration with Pattern 3: Straight Line Configuration

Next, a group of bacteria are initialized in a vertical line, spaced an equal distance from each other. Their initial orientations are all pointing to the right (figure 3-20). This is meant to be similar to the jets seen in [26]. The simulation is run from this initial configuration with various values of $B$ and the seven proposed control policies.
for \( \hat{k} \). The most interesting results from this pattern arise from setting \( \hat{k} = \hat{k}_3 \), where each organism wants to align itself with the flow either upstream or downstream to reduce drag. The bacteria in this case do not continue to swim to the right. Instead, they separate into two even groups. The top group turns left and the bottom group turns right until both groups reach a steady-state configuration where one bacterium swims directly behind the one in front of it (figure 3-21).

![Figure 3-20: Initial Straight Line Configuration](image)

The alignment suggests that this is a feasible control policy. The only concern is that the speeds of the organisms in the global reference frame are mostly lower than if each one swam on its own. The “lead” bacterium on each side swims at a dimensionless speed greater than 1, but all the “follower” bacteria swim slower than 1. Speed and orientation results for these trials are shown in figure 3-22. Here, \( N = 10 \), the separation distance between neighboring bacteria \( d_s = 20 \), and \( B = 0.75 \) or \( B = 1 \). The system behavior can be analyzed in more detail by varying these parameters.

First, the effect of changing \( B \) is examined. As can be seen from the first two runs, higher values of \( B \) lead to the bacteria paths curving more. This is quantified by the parameter \( \theta_c \), defined as the angle of the orientation of the top group of bacteria with
Figure 3-21: Paths of 10 Bacteria Starting in Line Configuration with Drag-Reduction Control Mechanism

respect to the x-axis. The convergence time $t_c$ is defined as the time it takes for all the bacteria to align with each other within some small threshold, so

$$ t_c = \min(t) \text{ such that } \max_i(\phi_i) - \min_i(\phi_i) < \epsilon $$

(3.68)

with $\epsilon = 0.02$ in these trials. Another way of evaluating the system behavior is to look at the average speed $<U>$ of the bacteria after they align with each other. Plots showing how $\theta_c$, $t_c$, and $<U>$ change with variation in $B$ are shown in figure 3-23. These trials are run with $d_s = 20$ and $N = 10$.

The general trend shows that as $B$ increases, the bacteria take longer to reach their alignment angle, which is an intuitive result because $B$ sets the timescale of reorientation and rotation. The bacteria sweep out curving paths of larger radii and so the alignment angle $\theta_c$ also increases with higher $B$. An interesting result is that once the bacteria do align, their final average speed increases with $B$ even though it takes longer to reach this state.

Next, the dependence of the system behavior on the initial separation distance is...
Figure 3-22: Speeds and Orientations of 10 Bacteria Starting in Line Configuration with Drag-Reduction Control Mechanism
Figure 3-23: Changes in System Behavior Arising from Changes in $B$
explored. These trials are run with $N = 10$ and $B = 1$, and the results are shown in figure 3-24.

When the bacteria start closer together, they take longer to align with each other, and their final angle is larger. Again, the variation in average speed proves interesting; there appears to be a minimum speed when $d_s = 17$. When the bacteria start closer together, their final speed increases a slight amount. However, when $d_s < 15$, the system does not show alignment (which could be an inherent property of the model or could merely be a result of approximations within the simulation). For larger separation values, the average speed generally increases as the separation increases.

Lastly, the system behavior is tested for differently-sized groups of bacteria. When the group size is odd, the behavior is similar to when it is even, but the center bacterium swims straight to the right. The $\frac{N-1}{2}$ bacteria above it and below it branch off and align with each other as in the even case. Samples of this are shown in figure 3-25.

Because of the slight differences in behavior, cases where $N$ is odd and cases where it is even are analyzed separately from each other. The results of trials with varying values of $N$ are shown in figure 3-26. In these trials, $B = 1$ and $d_s = 20$.

As $N$ increases, so do $t_c$ and $\theta_c$; on the other hand, $< U >$ decreases with increases in $N$. The more bacteria are placed in line with each other, the slower their average speed. This suggests that this approach probably does not describe how bacteria actually swim. Experiments typically show increased bacteria speeds when the organisms are in a collective motion regime and that does not happen in this setup. However, the results could have another application. Biomedical researchers are trying to create microscopic robots, called microbots or cytobots, that could swim through fluid in the human body [3, 32]. These robots would be small enough to be considered low Reynolds number swimmers and so the models presented here could apply to them as well. If a human is designing the algorithms for a group of microbots and decides that alignment is more important than maximizing speed, the results described here would be relevant and useful.
Figure 3-24: Changes in System Behavior Arising from Changes in Initial Separation Distance
Figure 3-25: Bacteria Paths with Even and Odd N

While they system behavior with an initial line configuration is analyzed here in a number of situations, all trials presented so far have started with perfectly symmetric initial conditions. A further set of tests should be performed to see how the behavior changes when noise is introduced into the system. There could be some uncertainty in the initial positions, in the initial orientations, or in both, as any real system is likely to have noise in both position and orientation. Another interesting test case would have noise introduced into half of the bacteria and then mirrored. For example, if this is done with the bacteria’s positions, the spacing between them would not be constant but the system would still have a form of symmetry. The noisy positions of half the bacteria would be reflected over a symmetry line running perpendicular to the line of initial positions and through its center.

### 3.3.2 Randomized Simulations

**Model Parameters**

While the pattern configurations yield interesting results, they do not provide much insight into how real bacteria swim. The behaviors that resemble collective motion
Figure 3-26: Changes in System Behavior Arising from Changes in Number of Bacteria (N)
patterns observed in experiments appear to rely on symmetry; as soon as the symmetry is broken in the disc and ring cases, the circular motion stops. Since the ring case is completely symmetric initially, in a perfect simulation the bacteria would stay in a symmetric formation for all time. However, the singularities in the flow lead to very high speeds and forces, and the simulation has a finite time step, so the symmetry breaks when one organism gets too close to a singularity. This is an artifact of the implementation in Matlab but it reveals that a slight deviation from a symmetric configuration does not lead to circular motion.

While each of the patterns yield useful results within a different range of values for $B$, $B = 0.75$ lies in this range in each case, so this value is used in the trials presented here. They use 60 bacteria in a domain with $L = 60$ which yields a volume fraction $v = 0.035$. While this is a lower concentration than what is presented in some experimental results [26, 53] it is still high enough that it could display collective behaviors [41]. The periodic domain calculations are performed using a 7x7 grid with 49 tiles. Initially, the bacteria are randomly distributed over the domain and their orientations are set randomly as well. The same random configuration is used for each choice of $k$. The simulation runs for 100 time steps ($t_{\text{final}} = 100$) and the internal time step parameter $\delta t$ is set to 0.1.

Results

With these settings, the bacteria do not qualitatively appear to exhibit collective behaviors for any choice of $\hat{k}$. This could be due to a variety of factors that will be explored later. The behavior of the organisms can still be analyzed in a few ways. Calculations of the flow field, vorticity field, and pressure field are difficult to analyze because of the singularity produced by each organism. Even with regularization, the singularities produce regions of fluid that move significantly faster than the bacteria’s swimming speed. Instead, the direction of the flow can be analyzed. Figure 3-27 shows a snapshot of the velocity orientation vectors from the case where $\hat{k} = \hat{k}_1$ and
Each bacterium wants to swim straight. Each bacterium is represented as a black circle with a white line indicating its instantaneous orientation. The shading in the image represents the curl of the normalized velocity field, so if the shading parameter is called \( \sigma \) and the velocity of the fluid is \( \mathbf{u} \), then

\[
\sigma = \nabla \times \left( \frac{\mathbf{u}}{\| \mathbf{u} \|} \right)
\]  

(3.69)

The figure shows that, while the bacteria are not swimming in vortices, neighboring regions of fluid that are rotating in opposite directions. This suggests that this model captures at least some aspect of the bacteria’s interactions that lead to rotating vortices, even though it does not capture the behavior itself.
Qualitatively, the snapshots taken from simulations with other choices of the control parameter $k$ look very similar to the one shown here for $k = k_1$.

**Tracer Particles** One way to analyze the results is to follow the path of tracers in the flow. Evidence of superdiffusion of tracer particles has been found when the particles are much larger than the individual organisms [53] and also when the particles are much smaller relatively [21], so the superdiffusion must be a property of the fluid motion itself and be independent of particle size. The simulations here do not model the tracers as particles with a finite size because a finite-sized particle will have some effect on the fluid. Instead, each tracer is a point in the fluid. At each time step, the induced velocity at that point is calculated and used to find the point’s position at the next time step.

The simulations are seeded with $N_T = 20$ tracers in random initial positions. The position of tracer $i$ at time $t$ (where $i = \{1, \cdots, N_T\}$) is denoted $y_i(t)$. The mean-square distance traveled by the tracers is then calculated by averaging over time and also over all the particles as follows:

$$<\Delta y(t)^2> = \frac{1}{N_T} \sum_{i=1}^{N_T} <[y_i(t + \tau) - y_i(\tau)]^2>_\tau$$

(3.70)

The average over $\tau$ is found by using $\tau$ as a sliding parameter ranging from $\tau = 0$ to $\tau = t_{\text{final}} - t$ [53]. The dependence of $<\Delta y(t)^2>$ on $t$ can be approximated by $<\Delta y(t)^2> \sim t^\alpha$. When $\alpha = 2$, the distance traveled is proportional to time, as in ballistic motion or motion at a constant velocity. In a diffusive regime, $\alpha = 1$ and distance is proportional to the square root of time. The randomness of Brownian motion in diffusion processes leads to the shorter distances. The regime in between is called the *superdiffusive* regime, so superdiffusion occurs when $1 < \alpha < 2$. Figure 3-28 shows the results for $<\Delta y(t)^2>$ with all the different possible control policies.

Experiments find evidence of superdiffusion at short timescales, but that does not
Figure 3-28: Logarithmic Plot of Mean-Square Distance of Tracers versus Time

seem to be the case in simulation. The lines of $< \Delta y(t)^2 >$ versus time all have approximately constant slope, and on a logarithmic scale, they lie parallel to the line $< \Delta y(t)^2 > = t$ which has $\alpha = 1$. This means that the tracers move diffusively and not superdiffusively, though with different apparent diffusion coefficients. The absence of superdiffusion in simulation could be because tracer particles observed in experiments move due to both Brownian motion and advection from the fluid (where the fluid motion is caused by the bacteria swimming). However, in simulation, Brownian motion is ignored and the tracers’ motion is based solely on the flow advection. Alternatively, it could be an indicator of the lack of collective motion in this model.

**Alignment Parameter** Another measure of how the bacteria’s motion affects the fluid is the directional coherence within the flow. This can be found by first dividing the domain into an even grid. (The results presented here use 40 divisions in each direction for 160 total grid points.) The fluid velocity can be calculated at each of
these points for each time step. Then, the cosine of the angle between any two velocity vectors $u_1$ (at point $p$) and $u_2$ (at point $q$) is found by:

$$\kappa_{p,q} = \cos \theta_{p,q} = \frac{\|u_1 \cdot u_2\|}{\|u_1\| \|u_2\|}$$

(3.71)

The alignment parameter $\Theta_R(t, p)$ measures directional coherence by averaging $\kappa_{p,q}$ over the grid points $\{q\}$ within some radius $R$ of $p$ at time $t$. It is defined as

$$\Theta_R(t, p) = \frac{1}{N_R} \sum_{(q \mid \|p-q\|<R)} \kappa_{p,q}$$

(3.72)

where $N_R$ is the number of points that lie within the circle prescribed by $R$. (This method comes from liquid crystal theory as described in [27] and uses the specific alignment parameter of [1].) The units of $R$ are relative to the grid spacing, so for example $R = 1$ implies that the points directly above, below, to the left, and to the right of $p$ will be included in the calculation of $\Theta_R$ regardless of the grid spacing. Examples of the alignment parameter calculations are shown in figure 3-29; the simulation trial and time step are the same as the one shown in figure 3-27.

In regions where the fluid moves uniformly in one direction, the flow is coherent and $\Theta_R$ will be close to 1. If the flow is completely disorganized, then $\Theta_R \approx 0$, and if $\Theta_R$ is calculated in a region with opposing streamlines, it will approach -1. Changing the value of $R$ provides insight into the length scales of the coherence. For each radius, $\Theta_R$ is divided into “bins” and the area fraction of each bin is calculated. The fractions are then averaged over 200 time steps and the results are plotted in figure 3-30.

For small $R$, much of the field is coherent. As $R$ increases, the peak of the histogram shifts to lower values of $\Theta_R$ and the flow appears less coherent over these longer length scales. The standard deviation is very small, which implies that the general appearance of the flow stays constant over time. (Note that the plot shows results
Figure 3-29: Value of Alignment Parameter at $t = 40$ for Different Values of R
Figure 3-30: Average Area Fraction of Values of Alignment Parameter
Average Speed of Bacteria

One reason why collective motion is advantageous to bacteria is that they swim faster in jets and in vortices than they do individually [26]. Despite the absence of collective motion here, looking at the speed of the bacteria provides some similarity between the model and experiments. The speeds are averaged over time and over the number of bacteria and the results for each control policy are shown in figure 3-31.

In all cases, the average speed is at least 1.28 times as fast as the individual bacterium speed. This is a large increase and it is an encouraging result because it matches experimental observations. The control policy that leads to the highest speeds is \( \hat{k} = \hat{k}_1 \), where each bacterium wants to swim straight whenever possible. Interestingly, the second case has one of the lower average speeds. When \( \hat{k} = \hat{k}_2 \), each organism tries to align with the local induced flow and swim in the same direction as it. If one bacterium was swimming in a uniform flow, it would swim with the fluid and its speed would be equal to its own speed plus the speed of the flow. However, in the
case with many bacteria, the complicated interactions and flow patterns must prevent this additive affect from lasting for any significant time period.

3.4 Discussion

In previous research, complex bacterial patterns have been analyzed on the scale of a large group of organisms, and the individual behavior of one organism is difficult to observe in this framework. In the presence of global gradients or force fields, bacteria have been seen to adjust their behavior; *E. coli* and *B. subtilis* swim up oxygen gradients [19], *Daphnia* swim towards light [14], and other algae swim vertically against gravity [29]. The experiments performed on *B. subtilis* in [53] use a domain without any global gradients. The bacterial bath is placed in a thin horizontal soap film where gravitational effects are insignificant and all bacteria have equal access to oxygen—yet large-scale patterns are still observed. This thesis hypothesized that the bacteria might sense some property of the flow that is determined locally by the contributions of other bacteria, instead of only basing their “intelligent” behavior on global properties. The simulations suggest that this is not the case. The overall group behavior seen when the bacteria use the different proposed control policies does not change significantly from the baseline case where each organism wants to swim straight. Moreover, the other policies lower the average speed of the bacteria, which makes them seem even less feasible. They could still be useful for applications to microbots but they do not seem to capture the behavior of real organisms. This does not mean that the bacteria interact with each other; in fact, they interact just by swimming and being advected by each other’s motions. This advection due to the flow field created by the other organisms probably has the most significant impact on the bacteria’s motion and behavior.

Many improvements and adjustments could be made to the current model to make it more realistic. Most importantly, more work should be done on the near-field flow
and on modeling what happens when two organisms are close together. The model does not account for collisions between organisms, so even though it restricts the positions to a two-dimensional plane, it currently allows two (or more) bacteria to occupy the same space in the plane. Although it has been demonstrated that the effect of shape (ellipsoid/rod versus sphere) does not have a significant effect on the flow, the shape could play a significant role in bacteria interactions and the effect of collisions. Results from liquid crystal theory show that when two rods approach each other and they are closely packed, steric repulsion causes them to align with each other [40]. This short-range alignment, currently unaccounted for, could be the “missing link” that would lead the modeled bacteria to display collective motion.
Chapter 4

Conclusions

4.1 Contributions of this Research

4.1.1 Biomimetic Robots

Chapter 2 presents a model for schooling fish by analyzing the interactions between one fish and its neighbors. This model is then applied to simulations of underwater robots. It shows that group-wise coordinated behaviors can be achieved through simple control laws that depend only on local information. If each element aligns with its neighbors, moves away from elements that are too close, and moves towards elements that are farther away than some desired distance, all elements will come together in a crystal-like formation that translates in a straight line. The direction of this line and the speed at which the elements travel along it can be controlled by two simple parameters. The applied thrust input, set identically for all elements, determines the speed. The orientation of the line is set by ensuring one element moves in the desired direction and then making this element “blind” so that it does not sense its neighbors. All other elements will enter the same tightly packed formation and move in the direction of the leader. The thesis also presents preliminary theoretical
analysis of this behavior using contraction theory.

The coupling terms between elements are shown to provide cohesiveness even as the group responds to an external input. If an underwater search were performed with one element, the element would need to make many passes over an area to ensure full coverage. With a group of elements swimming in formation, the number of passes can be greatly reduced because only one or two elements need to sense a target in order to direct the whole group to that area. The elements do not need to communicate with each other because by sensing each others' positions and velocities, they can continue to school and stay close to each other as one or two change their directions due to identification of a target.

4.1.2 Low Reynolds Number Swimmers

Chapter 3 lays the foundations for a rich hydrodynamic model of organisms swimming at low Reynolds number. In this regime, inertia is negligible, and the Navier-Stokes equations are replaced by the linear, autonomous Stokes equations. Momentum dissipates over long distances in Stokes flow. The velocity around a sphere moving at high Re decays with the distance $r$ from the sphere like $1/r^3$, while at low Re it decays like $1/r$, so the effect of an object in fluid can be felt at significantly longer distances. At high concentrations, the motion of each bacterium influences the motion of all other bacteria directly and instantaneously due to hydrodynamic interactions. The model presented here quantifies the hydrodynamic flow field induced by one organism. Then, exploiting the linearity of Stokes equations, it finds the bulk effect of many swimming bacteria by superimposing the flow fields caused by the individual organisms onto each other.

Simulations of the model with different initial conditions and possible control policies are presented. While the simulations do not qualitatively match experimental results exactly, they bear similarities which indicate that the model captures some aspects
of the complex problem of many bacteria swimming together. This model could be expanded in a number of ways that should lead to a closer match to experimental results of collective motion in low Re swimming organisms.

4.2 Future Work

4.2.1 Biomimetic Robots

With random initial conditions and velocities, a group of elements can use local control algorithms to swim in a specified direction at a constant, pre-determined speed while staying in a closely-packed formation. What if the leader element follows a trajectory other than a straight line? For example, the leader might be commanded to trace out a weaving path that provides coverage of a given area, or it might swim in a protective circle around a ship. The behavior of the follower elements could be analyzed and tested in cases where the leader follows a curved or otherwise nonlinear trajectory. Another way to expand on the current control laws would be to further develop the groups’ reaction to external inputs. It has been shown that the group can be directed to a target that at first is only observed by a few elements, but the elements’ behavior once they are inside the target has not been fully specified.

An ideal biomimetic robot swarm should additionally be able to navigate around its environment easily. This means that the elements need control laws for obstacle avoidance. Preliminary simulation results show that replacing the radially-symmetric potential function with an elliptical potential function can lead to the elements forming an elongated cluster. This behavior could be useful for squeezing through a tight space. The elements also need an algorithm specifying their reaction to convex obstacles. The group could turn and go to one side of such an obstacle, or it could split into smaller groups that would pass by the obstacle on either side and join up again afterwards.
4.2.2 Low Reynolds Number Swimmers

As mentioned in §3.4, the model for swimming bacteria presented here does not account for collisions between individuals. Modeling collisions in the correct manner would greatly improve the current model. The bacteria might bounce off of each other in an elastic, energy-conserving collision. Alternatively, the steric effects of two rod-shaped bacteria swimming closely to each other might dominate and cause the organisms to align with each other before they actually collide. The flagella are not modeled here either. A bacterium that swims directly behind another one would collide with the latter's flagella, and this collision should be included in the model.

The model could be developed further by including wall effects. An organism swimming through a large volume of fluid produces a different flow field than one swimming near a wall or along a surface. When this effect is included, the simulation could be implemented in three dimensions to see whether the bacteria swim towards or away from the surface as discussed in [17]. Lastly, the run-and-tumble behavior of real bacteria could be incorporated into the model. Each organism generally swims straight for a short period of time (called a run) and then stops and changes orientation randomly (called a tumble). This random effect might be related to the timescales of collective motion observed in [26] and [41]. With these additions, the model would come closer to emulating how bacteria and other organisms that swim at low Reynolds numbers interact with each other and with their fluid environment.
Bibliography


