Computational Study of 2D Jellyfish with the Immersed Boundary Method

by

Mahdi Salehzadeh

B.Sc., AmirKabir University of Technology (Tehran Polytechnic), 2019

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Fall 2022

Declaration of Committee

Name:	**Mahdi Salehzadeh**
Degree:	**Master of Science**
Thesis title:	**Computational Study of 2D Jellyfish with the Immersed Boundary Method**
Committee:	**Chair:** Weiran Sun Associate Professor, Mathematics
	John Stockie Supervisor Professor, Mathematics
	JF Williams Committee Member Associate Professor, Mathematics
	Paul Tupper Examiner Professor, Mathematics

Abstract

Jellyfish have evolved the most energy-efficient method of propulsion of any animal on Earth, despite having an extremely simple physical and neural structure. For this reason, they are a very popular model organism for biologists and have been the subject of numerous experimental and computational studies. But despite the many attempts to comprehend their swimming dynamics and performance, there still remains a great deal to discover about jellyfish. This thesis employs numerical simulations using the immersed boundary method to investigate the fluid-structure interaction between a 2D model for a swimming jellyfish with the surrounding fluid. Various jellyfish species employ a variety of swimming "gaits" to generate forward motion, and we will focus on jet-like swimming occurring mostly in prolate jellies that contract their bell muscles periodically to expel water from their interior. We first study the scaling properties of jellyfish in terms of two dimensionless parameters – the Reynolds number and swimming number – to show that a power-law dependence derived for undulatory swimmers (such as fish and eels) extends naturally to jellyfish that use a jetting mode of propulsion. We next investigate the feeding technique used by jetting swimmers, in which trailing vortices generated by bell contractions are exploited to redirect into their bell interior the mostly passive prey such as algae and plankton that they feed on. Our numerical simulations are used to quantify and visualize the effect of changes in prey distribution and jellyfish size and shape with a degree of detail that is typically not possible in experiments. Finally, we present a preliminary study of pair-wise interactions between jellyfish in which nearby swimmers generate repulsion forces and initiate turning responses when they come into close proximity. The overall aim of this research is to lay the groundwork for future computational simulations of swimming and feeding dynamics in swarms of interacting jellyfish.

Keywords: jellyfish; immersed boundary method; biofluid mechanics; swimming dynamics; fluid-structure interaction

Dedication

This thesis is dedicated to the 176 beautiful souls onboard who left us so early. But will never be forgotten.

Acknowledgements

I wish to express my sincere gratitude to Dr. Stockie my supervisor whose guidance and support made this thesis possible. I also really appreciate my family's continuous love, passion, and interest in my academic accomplishment.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Jellyfish is the common name for a huge group of marine species known as the cnidarian medusae. They are easily observed in aquarium exhibits around the world, and their swimming dynamics are fascinating and beautiful to observe. Jellyfish are unique in the animal kingdom because of two characteristics: they are among the most energetically efficient animal swimmers [10], and their simple physical structure allows them to be modelled using only a small number of structural elements [9]. They have drawn the interest of biologists for a long time as the study [11] demonstrates, and their physical structure makes it especially easy to quantify their swimming dynamics: the transparency of their gelatinous bodies permits easy viewing through their body tissues, and their radially symmetric body architecture provides a significant advantage for extending two-dimensional measurements [17, 3, 2] to the whole three-dimensional body shape [15, 18, 12]. This combination of high efficiency, minimal complexity, and easy availability of data is also advantageous for using jellyfish as a model for energy-constrained underwater vehicles that do not require high speeds.

In this thesis, we study jellyfish swimming as a fluid-structure interaction (FSI) problem, involving a swimming jellyfish as a deformable elastic structure in an incompressible fluid using the immersed boundary method (IBM). The IBM was initially developed by Peskin [25] to study the flow of blood flow through human heart valves. To solve this FSI problem, we utilize the open-source immersed boundary software package `IB2d`. A detailed discussion of the IBM and `IB2d` can be found in Chapter 2. Instances of other FSI approaches used to study swimming jellyfish are extraction of bell cyclic movement from the video records of a swimming jellyfish while its displacement was simultaneously estimated from the numerically anticipated forces of the bell on the fluid [13], and a geometrically conservative arbitrary Lagrangian-Eulerian formulation for the moving boundary problems in the swirl-free cylindrical coordinates [29].

In Chapter 3, we describe the geometry of a jellyfish bell in our two-dimensional computational studies which is based on the genus *Sarsia*, similar to the model proposed by

Hoover et al. [19]. *Sarsia* were also studied computationally by Miles et al. [5] and Colin et al. [8].

Aquatic organisms have a variety of locomotion mechanisms and swimming dynamics. Different swimming modes may be more or less efficient at different scales or Reynolds numbers. In the intermediate to high range of Reynolds numbers, undulatory locomotion is common and includes anguilliform swimming in lampreys and eels, carangiform swimming in most fish and thunniform swimming in tuna, whales, and dolphins. However there are many other creatures that employ alternate modes of propulsion such as suction-based swimming in oblate jellyfish or jetting propulsion using momentum injection in prolate jellyfish. The dynamics of these various modes have received significant attention from the scientific community through experimental investigations and computational fluid dynamics (CFD). Bioinspired computational studies are crucial because they make it relatively easy for researchers to explore biological parameter spaces. For instance, researchers have looked at the characterization of undulatory locomotion and uncovered a power-law scaling relationship that connects body kinematics and fluid characteristics to swimming velocity, and swimmer-specific CFD models have also supported this scaling law. In particular, the study of Gazzola et al. [14] prompted us to investigate the scaling applied to jellyfish locomotion which uses jet-like propulsion in Chapter 4.

Investigating jellyfish feeding dynamics and efficiency is vital because they make up a dominant group of mesozooplankton that have experienced huge blooming events in oceanic waters in recent years. Jellyfish feeding rates vary based on characteristics such as the predator species capture surface morphology [6], size, and behaviour, as well as the prey's species, size, and behaviour. Several previous experimental studies have attempted to analyze jellyfish feeding behaviour [1, 30], but there has not yet been a thorough computational study of prey capture by a swimming jellyfish. In Chapter 5, we perform such a computational study, focusing on the role of various prey distribution patterns and parameters on jellyfish feeding performance which in practice is difficult to do using lab experiments, but relatively easy in simulations.

Jellyfish almost never swim alone but rather appear in large, dense swarms, so the interaction dynamics should also be carefully examined. One of the gaps in the literature is the careful evaluation of pairwise interactions. In Chapter 6, we were inspired by the numerical study of hydrodynamic interaction of swimming organisms at intermediate Reynolds number regimes by Li et al. [22] to investigate the repulsion between interacting jellyfish using our 2D implementation.

Chapter 2

Immersed Boundary Method

In the immersed boundary method (IBM) an immersed boundary is modelled as an elastic fiber in 2D (or an interwoven mesh of such fibers in 3D) which are immersed within an viscous and incompressible fluid. The immersed structures are neutrally buoyant and move with the local fluid velocity, while at the same time exerting on the adjacent fluid particles an elastic force which depends on their state of stretching and bending (the force term can be any function of the immersed boundary configuration). The IBM employs a mixed Eulerian-Lagrangian framework to discretize the fluid-structure interaction (FSI) problem.

2.1 Governing Equations

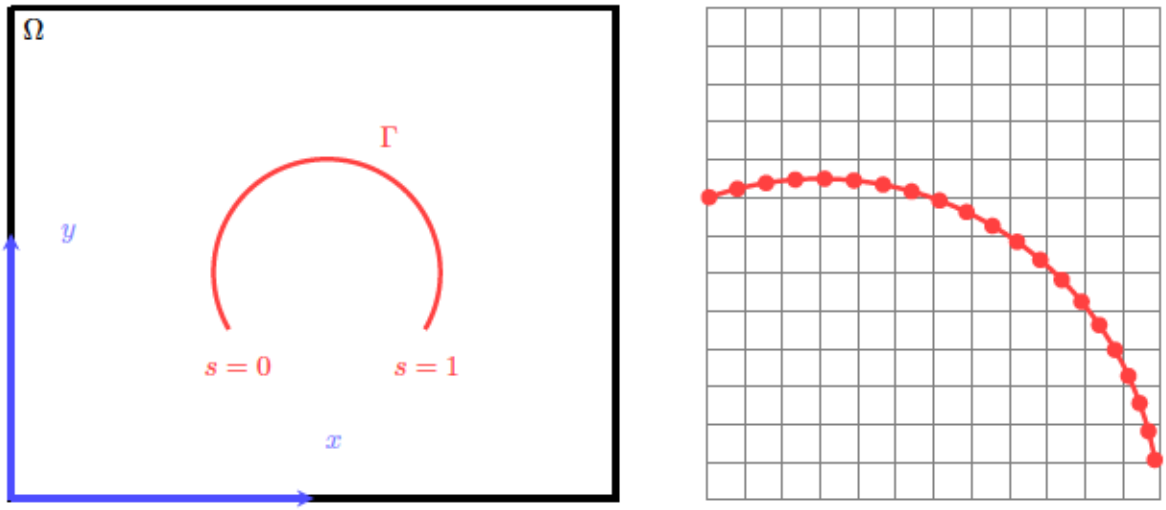
The fluid is described by velocity $\mathbf{u}(\mathbf{x},t) = (u(\mathbf{x},t), v(\mathbf{x},t))$, pressure $p(\mathbf{x},t)$ and body force $\mathbf{f}(\mathbf{x},t)$. Here, $\mathbf{x} = (x,y)$ is the Eulerian position variable in two dimensions and t is the time variable. The forces applied to the fluid by the immersed boundary are described by the function $\mathbf{f}(\mathbf{x},t)$, which gives the external force per unit volume applied to the fluid. The governing equations of motion for a viscous and incompressible fluid are the Navier-Stokes equations which are:

$$\rho\left(\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \mathbf{u}(\mathbf{x},t)\cdot\nabla\mathbf{u}(\mathbf{x},t)\right) = -\nabla p(\mathbf{x},t) + \mu\Delta\mathbf{u}(\mathbf{x},t) + \mathbf{f}(\mathbf{x},t) \tag{2.1}$$

$$\nabla\cdot\mathbf{u}(\mathbf{x},t) = 0 \tag{2.2}$$

where Equation (2.1) is equivalent to the conservation of fluid momentum, and Equation (2.2) is the incompressibility condition.

The immersed structure is typically represented as a collection of one-dimensional fibers, denoted by Γ. Each fiber can be viewed as a parametric curve $\mathbf{X}(s,t)$ where s is a Lagrangian parameter and t is time. The entire fluid domain is denoted by Ω, see Figure 2.1a.



(a) (b)

Figure 2.1: (a) The 2D immersed boundary model for a jellyfish bell, consisting of a fiber Γ immersed in a rectangular fluid Ω. (b) The fluid grid points and moving fiber mesh points.

2.2 Fluid-Structure Interaction (FSI)

The interaction between the fluid and the immersed structure is described as follows:

- The force exerted by the immersed boundary on the surrounding fluid is represented as a source term in the momentum Equation (2.1) using

$$\mathbf{f}(\mathbf{x},t) = \int_\Gamma \mathbf{F}(s,t)\delta(\mathbf{x} - \mathbf{X}(s,t))ds, \tag{2.3}$$

  where $\mathbf{F}(s,t)$ is the force density exerted by the immersed elastic boundary and δ is a two-dimensional delta function.

- The immersed structure moves at the local fluid velocity, thus its motion equation can be written as

$$\frac{\partial \mathbf{X}(s,t)}{\partial t} = \mathbf{u}(\mathbf{X}(s,t),t) = \int_\Omega \mathbf{u}(x,t)\delta(\mathbf{x} - \mathbf{X}(s,t))d\mathbf{x}. \tag{2.4}$$

Note that Equation (2.3) makes sense since integrating over all $\mathbf{x} \in \Omega$ yields the following

$$\int_\Omega \mathbf{f}(\mathbf{x})dx = \int_\Gamma \mathbf{F}(s,t)ds. \tag{2.5}$$

The force $\mathbf{F}$ as defined by Equation (2.3) is singular, being zero everywhere except on the immersed structure (Γ) and yet having a finite integral over the fluid domain (Ω). In other

words, the total of all the forces acting on the fluid is equal to the total of all the immersed structure forces (which are the only forces applied to the fluid).

Since the force density $\mathbf{F}(s,t)$ is determined by the boundary configuration $\mathbf{X}(s,t)$, it can be written as a function describing the elastic properties of the boundary, or in other words, it is a combination of all the fiber components modelling the desired material properties of the immersed structure. For instance, the simplest force density for a linearly elastic material is $\mathbf{F}(s,t) = \sigma\partial^2\mathbf{X}/\partial s^2$ which is analogous to having Hookean springs linking successive fiber points, each with spring constant σ and resting length zero.

Equations (2.1)-(2.4) are a coupled system of partial differential equations, which we refer to as the immersed boundary method. Equations (2.1)-(2.4) clearly show that the δ function is responsible for fluid and immersed structure interaction.

2.3 Numerical Method

We consider approximations at equally-spaced times $t_n = n\Delta t$ and divide the fluid domain $\Omega = [0, L_x] \times [0, L_y]$ into a regular $N_x \times N_y$ square lattice of points with spacing $\Delta x = \Delta y = h$. The Eulerian points on the 2D fluid grid are denoted by (i, j) so that discrete approximations of fluid quantities can be written $U_{ij}^n \approx \mathbf{u}(ih, jh, t_n)$ and $P_{ij}^n \approx p(ih, jh, t_n)$ representing the fluid velocity and pressure at node (i, j) and time $n\Delta t$, for $i = 0, 1, \cdots, N_x - 1$ and $j = 0, 1, \cdots, N_y - 1$, respectively. Similarly, we discretize the fiber at a set of N_b moving points, with spacing $\Delta s \leq 0.5\Delta x$ and the immersed boundary is represented by discrete points $\mathbf{X}_k^n \approx \mathbf{X}(k\Delta s, n\Delta t)$ for $k = 0, 1, \cdots, N_b - 1$. Note that choosing grid spacing as $\Delta s \leq 0.5\Delta x$ is a practical accuracy requirement that avoids leakage of the fluid between the immersed boundary points. The numerical scheme we use employs an explicit discretization

Numerical parameter	**Symbol**	**Value**	**SI units**
Spatial Cartesian grid	$\Delta x = \Delta y$	$L_x/N_x = L_y/N_y$	-
Spatial Lagrangian grid	Δs	$\Delta x/2$	m
Time step size	Δt	10^{-5}	s

Table 2.1: Table of all numerical parameters.

of advection terms and so there is a built-in stability restriction of the form $|\mathbf{u}|\Delta t \leq \Delta x$ everywhere, which implies that the displacement of the immersed boundary points is less than a mesh width during any time step [25]. A typical Eulerian-Lagrangian grid is pictured in Figure 2.1b, from which it is clear that Lagrangian nodes need not coincide with the Eulerian mesh. Here, the crucial role of the delta function as an interpolating function between fluid and immersed boundary becomes evident.

Upon discretizing the fluid-structure interaction (FSI) equations, (2.3) and (2.4), we replace $\delta(\mathbf{x})$ by the discrete approximation $\delta_{\Delta x}$, which is non-singular for each Δx but approaches $\delta(\mathbf{x})$ as $\Delta x \to 0$. Although there are many possible variants of the discrete delta

function, we use $\delta_{\Delta x}(\mathbf{x}) = \phi(x)\phi(y)$ where

$$\phi(x) = \begin{cases} \frac{1}{2\Delta x}\left(1 + \cos(\frac{\pi x}{2\Delta x})\right), & x \leq 2\Delta x \\ 0, & x > 2\Delta x \end{cases} \tag{2.6}$$

which is chosen according to [26] to satisfy a series of discrete compatibility conditions.

Now, consider a two dimensional rectangular fluid domain with periodic boundary conditions that contains an immersed boundary that interacts with the fluid. Although imposing periodic boundary conditions is not entirely physical, we can justify this choice since, for our study, the fluid is at a relatively low Reynolds number, and as long as the structure remains near the center of the domain, the fluid viscosity is high enough to damp out any disturbances near the boundaries that could interfere with the periodic copies. Furthermore, periodic boundary conditions mimic the presence of multiple jellyfish interacting in a swarm, which is a long-term goal of this work. Finally, the periodicity assumption allows a Fourier method to be employed as a fluid solver, which is both straightforward and efficient. An example of such an immersed structure which motivated the present study is a jellyfish that propels itself using regular contractions of its bell muscles. In Chapter 3 we will describe the 2D model of immersed jellyfish bell and its implementation.

The numerical algorithm for the IBM described as follows:

Step 1: Compute the elastic force $\mathbf{F}^n(s,t)$ on the immersed boundary from the current immersed structure configuration $\mathbf{X}^n$, where n indicates the n^{th} time-step.

Step 2: Spread these forces from the Lagrangian nodes to the neighboring fluid grid points, with a discretized form of the integral in Equation (2.3).

Step 3: Update the fluid velocity everywhere in the domain using the Navier-Stokes equations, Equations (2.1) and (2.2). E.g., update $\mathbf{u}^{n+1}$ and $\mathbf{p}^{n+1}$ from $\mathbf{u}^n$ and $\mathbf{f}^n$.

Step 4: Update the immersed structure position, $\mathbf{X}^{n+1}$, using the immersed boundary velocity computed from $\mathbf{U}^{n+1}$ and Equation (2.4).

Step 5: Increase n and go to step 1.

The above algorithm had been previously used in initial release of *IB2d* and the details of the discretization can be found in Section 2.4. However, the current release no longer requires square grids (with $\Delta x = \Delta y$), instead supporting all rectangular grids. A second-order time stepping scheme has also been implemented giving rise to a formally second-order accurate IBM. Details of this recent algorithm's implementation can be found in [4].

2.4 IB2d

An open source immersed boundary software package, *IB2d*, is described in this section. Implementations of *IB2d* are provided in both MATLAB [23] and Python 3.5 [28] which are capable of modeling a broad array of problems in engineering and biomechanics including aquatic locomotion, cellular deformation and transport, blood flow, and wave-induced deformation of underwater structures [5, 4].

We utilize *IB2d* to mimic jellyfish swimming due to its ease of use, simplicity compared to 3D solvers, and additional functionality (compared with other IB solvers) for specifying a wide range of immersed boundary forces that can mimic material properties that are encountered in applications (e.g., elastic, visco-elastic, plastic, etc.)

The *IB2d* uses a Fast Fourier Transform (FFT) approach to solve the discrete fluid equations, which exploits the periodic fluid boundary conditions to solve the pressure Poisson equation. The process of solving the incompressible Navier-Stokes equations in *IB2d* using Fast Fourier Transform (FFT) can be summarized as follows:

1. Compute and store coefficients of the discrete Fourier transform operator.
2. Compute 1st and 2nd derivatives of the fluid velocity.
3. Compute the right hand side of the Navier-Stokes equations.
4. Perform the Fast Fourier Transform to transform velocity into wave number space.
5. Calculate the fluid pressure and velocity.
6. Perform the inverse Fast Fourier Transform to get velocities/pressure in real space.

Next we describe the discrete approximation used in *IB2d* for solving the Navier-Stokes equations, which is based on Appendix A of [5]. The *IB2d* uses finite difference approximations to discretize the Navier-Stokes equations on a fixed lattice, e.g., the Eulerian (fluid) grid. The discrete equations are an implicit linear system for the velocity U^{k+1}:

$$\rho\left(\frac{U^{k+1}-U^{k}}{\Delta t}+S_{\Delta x}(U^{k})U^{k}\right)-\mathbf{D}^{0}P^{k+1}=\mu\sum_{\alpha=1}^{2}D_{\alpha}^{+}D_{\alpha}^{-}U^{k+1}+F^{k} \tag{2.7}$$

$$\mathbf{D}^{0}\cdot U^{k+1}=0 \tag{2.8}$$

such that Δt and Δx are the time-step size and Eulerian mesh width, respectively, and $\mathbf{D}^0$ is the central difference operator, defined as

$$\mathbf{D}^{0}=(D_{1}^{0},D_{2}^{0}), \tag{2.9}$$

with

$$(D_{\alpha}^{0}\phi)(\mathbf{x})=\frac{\phi(\mathbf{x}+\Delta\mathbf{x}e_{\alpha})-\phi(\mathbf{x}-\Delta\mathbf{x}e_{\alpha})}{2\Delta x} \tag{2.10}$$

for $\alpha = 1, 2$ where $\{e_1, e_2\}$ is the standard basis in $\mathbb{R}^2$. The viscous term, given by $D_\alpha^+ D_\alpha^- U^{k+1}$ in (2.7) is a difference approximation to the Laplacian $\Delta \mathbf{u}$, where the D_α^+ is a forward and D_α^- is a backward approximation to $\partial/\partial x_\alpha$. The two difference operators are defined as

$$(D_\alpha^+ \phi)(\mathbf{x}) = \frac{\phi(\mathbf{x} + \Delta \mathbf{x} e_\alpha) - \phi(\mathbf{x})}{\Delta x}, \tag{2.11}$$

$$(D_\alpha^- \phi)(\mathbf{x}) = \frac{\phi(\mathbf{x}) - \phi(\mathbf{x} - \Delta \mathbf{x} e_\alpha)}{\Delta x}. \tag{2.12}$$

The skew-symmetric difference operator, $S_{\Delta x}$, is an approximation to the non-linear advection term $(\mathbf{u} \cdot \nabla \mathbf{u})$, and is defined as follows

$$S_{\Delta x} = \frac{1}{2}[U \cdot \mathbf{D}_{\Delta x}^0 \phi + \mathbf{D}_{\Delta x}^0 \phi \cdot (U\phi)]. \tag{2.13}$$

The following provides the justification for the skew-symmetric differencing of the convection terms [21]. Consider a model problem in which a scalar function ϕ is transported by an incompressible fluid with velocity $\mathbf{u}$

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0, \tag{2.14}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{2.15}$$

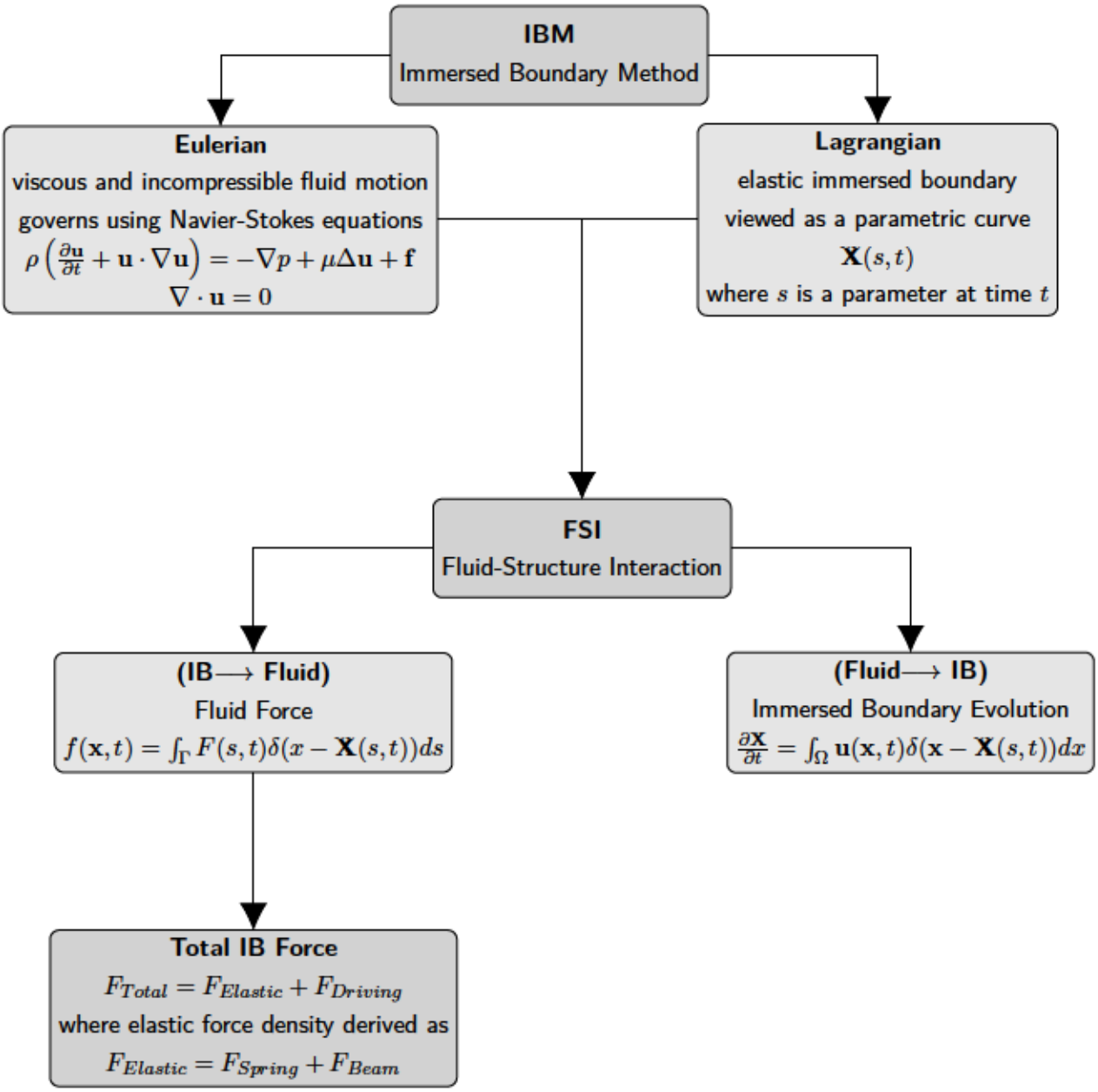
The integral of the L^2 norm of ϕ is conserved

$$\frac{d}{dt}||\phi||^2 = \frac{d}{dt}\int_\Omega \phi^2 dx = 0, \tag{2.16}$$

because the boundary conditions are periodic. Note that ϕ must be a continuous function of both time and space for Equation (2.16) to hold. It is desirable for the discrete solution to satisfy the analogous conservation property. This is because if the discrete convection equation satisfies both this conservation property and the standard Courant-Friedrichs-Lewy number criterion for linearized stability, the weak instability brought on by the nonlinear convection term can be avoided [27].

By discretizing (2.10), (2.11), (2.12) and (2.13), Equations (2.7) and (2.8) are linear in U^{k+1} and P^{k+1}. Note that there is no need to discretize the Navier-Stokes equations exactly as described, and the above can be replaced with any fluid solver of your choice, e.g., finite element, Lattice Boltzmann, etc.

A visual diagram of the immersed boundary method (IBM) steps is as follows (Note that the total IB force for the jellyfish model is fully explained in Chapter 3):



Chapter 3

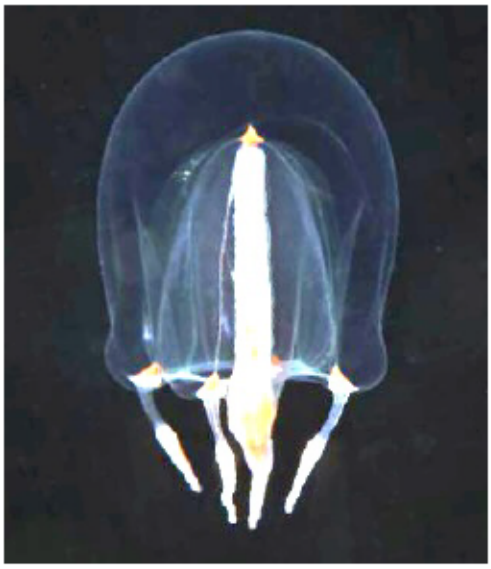
Jellyfish Structure and Dynamics

The current chapter introduces background preliminaries regarding jellyfish structure and swimming dynamics along with a simple 2D model. We start by defining the two dimensional geometry of the jellyfish bell and then introduce a simplified muscle structure that drives the swimming dynamics. We also describe the spatial discretization of the bell and its desired material properties. This is followed by a description of important nondimensional numbers like Reynolds number (Re) and swimming number (Sw) which characterize the jellyfish swimming dynamics. This chapter closes with a discussion of the implementation in the `IB2d` software package.

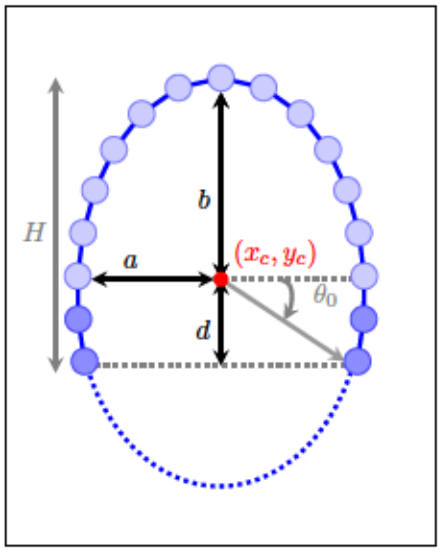
3.1 Computational Geometry

We chose the jellyfish genus *Sarsia*, pictured in Figure 3.1a, for our study since it propels itself by jetting motion and its bell configuration is straightforward to model. The other major mode of swimming that is more common to oblate jellyfish is called paddling, in which forward motion is driven by a much more complex and genuinely three dimensional wave of motion that propagates around the edge of the bell. This *Sarsia* jellyfish is similar to that studied by Hoover and Miller [19], where the jellyfish bell is approximated in 2D by a portion of an elliptical curve with a specified cut-off for the lower portion of the bell. In other words, the bell is a partial ellipse with semi-major axis b, semi-minor axis a and its overall height denoted as H (see Figure 3.1b). Various researchers have used this approach to model jellyfish, such as Hoover and Miller [19], Colin and Costello [8], and Herschlag and Miller [17]. The bell is made to resist bending and stretching and IB forces encourage the bell to return to its preferred configuration as this partial ellipse. The equation for the bell shape is

$$\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} = 1 \quad \text{for} \quad x_c - a \leq x \leq x_c + a, \quad y_c - d \leq y \leq y_c + b, \tag{3.1}$$



(a)



(b)

Figure 3.1: (a) Image of a swimming *Sarsia*. (b) The jellyfish geometry in 2D with bell height $H = b + d$, semi-major axis b and semi-minor axis (or bell radius) a.

where (x_c, y_c) is the center of the ellipse, a is the semi-minor axis of the bell which is also known as a bell radius, and $H = b + d$ is the height of the bell (see Figure 3.1b).

3.2 Jellyfish Dynamics

To determine the total force due to the elastic deformation of the bell and the force due to the contraction of the bell (driving force), we may separate the Lagrangian force density into the elastic force $F_{Elastic}$ and the driving force $F_{Driving}$ (contraction), such that the total force density is

$$F_{Total} = F_{Elastic} + F_{Driving}. \tag{3.2}$$

The elastic force in our two-dimensional jellyfish model consists of a stretching-resistant spring force and a bending-resistant beam force, therefore Equation (3.2) can be rewritten as

$$F_{Total} = F_{Spring} + F_{Beam} + F_{Driving}, \tag{3.3}$$

with more details in Sections 3.4 and 3.5.

3.3 Spatial Discretization

Based on the jellyfish model of Hoover and Miller [19] described in Section 3.1, the jellyfish bell is discretized by points a distance of Δs apart modelled in a Lagrangian framework. This initial discretization is a curvilinear mesh consisting of equidistant points along the

partial ellipse. The horizontal (x) and vertical (y) positions along the bell are prescribed using polar coordinates as follows:

$$x = a\cos(\theta) + x_c, \tag{3.4a}$$

$$y = b\sin(\theta) + y_c, \tag{3.4b}$$

where (x_c, y_c) is the center of the ellipse and the polar angle is $\theta \in (-\theta_0, \pi + \theta_0)$ such that $\theta_0 = \sin^{-1}(d/b)$, which satisfies Equation (3.1). Identifying k, $k+1$ as two successive Lagrangian points, the distance between points in the equilibrium configuration can be written as

$$\Delta s = \int_{\theta_k}^{\theta_{k+1}} \sqrt{\Delta s_1^2 + \Delta s_2^2}\, d\theta, \tag{3.5}$$

where Δs_1 and Δs_2 are defined as

$$\Delta s_1(\theta) = a\sin(\theta), \tag{3.6a}$$

$$\Delta s_2(\theta) = b\cos(\theta), \tag{3.6b}$$

and θ_k, θ_{k+1} are located so that Δs is constant. Then, the desired Δs can be written using a simple approximation of the integral as:

$$\Delta s = \int_{\theta_k}^{\theta_{k+1}} \sqrt{\Delta s_1^2 + \Delta s_2^2}\, d\theta \approx (\theta_{k+1} - \theta_k)\sqrt{(a\sin(\theta_k))^2 + (b\cos(\theta_k))^2}. \tag{3.7}$$

Simplifying Equation (3.7) results in the following formula for determining the position of Lagrangian points along the bell:

$$\theta_{k+1} = \frac{\Delta s}{\sqrt{(a\sin(\theta_k))^2 + (b\cos(\theta_k))^2}} + \theta_k, \quad \text{for} \quad k = 1, 2, \cdots, N_b. \tag{3.8}$$

Note that Equation (3.8) is used to determine both the initial position of the bell and the rest state configuration (to which the bell relaxes when there are no muscle contractions and the fluid is at rest). The number of Lagrangian points N_b is chosen to satisfy $\Delta s \lessapprox \Delta x/2$, which in the immersed boundary method minimizes interpolation errors between the Eulerian and the Lagrangian grids, and avoids mass conservation errors due to "leakage" between IB points [26].

3.4 Jellyfish Forces and Material Properties

This section presents all material properties of our jellyfish model relevant to desired application which are based on the spring and beam fiber models available in the *IB2d* implementation. See Figure 3.2 for a depiction of how immersed boundary points along the jellyfish bell are connected by spring and beam forces.

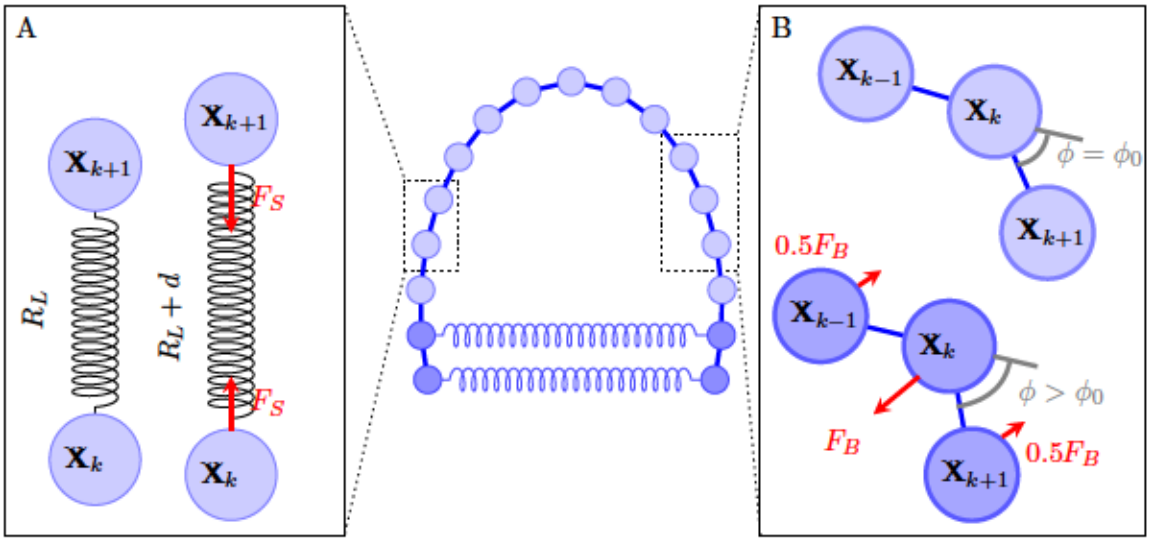


Figure 3.2: Critical aspects of the jellyfish model implemented in *IB2d*. (A) Each pair of successive Lagrangian points is connected by a virtual spring, shown at the resting length of the spring (left) and a rendering of the longitudinal forces induced when the spring is stretched (right). (B) Each set of three adjacent Lagrangian points is treated as an elastic beam with equilibrium angle ϕ_0. The resting configuration is shown above, and the case in which force being applied by the middle node ($\mathbf{X}_k$) while the system is not at its lowest energy state is shown below. There are forces corresponding to the bending force applied to $\mathbf{X}_{k-1}$,$\mathbf{X}_{k+1}$ in an opposite direction which are depicted as red arrows. A driving force is applied to the lower portion of the jellyfish bell (muscle points) which are the horizontal springs colored in dark blue.

For any given deformation energy `IB2d` never actually computes E. Instead, the code implements Equations (3.12) and (3.14) describing the stretching and bending state of the bell, by taking

$$E = E(\mathbf{X}_1(t), \mathbf{X}_2(t), \cdots, \mathbf{X}_{N_k}(t)), \tag{3.9}$$

and then the corresponding elastic force at point $\mathbf{X}_k$ can be written as the derivative of the energy

$$F_k = -\frac{\partial E}{\partial \mathbf{X}_k}, \tag{3.10}$$

where N_b is the number of immersed boundary points in the immersed structure. We note that Equation (3.9) is a combination of the deformation energies from all respective fiber models which are explained below. The code does not compute E, instead this is a convenient framework to define the forces that are implemented in the code.

3.4.1 Springs

The force between any two successive Lagrangian points that resists stretching and compression is modelled as a Hookean spring with resting length R_L and spring stiffness k_S.

The elastic potential energy for the spring connecting two successive Lagrangian nodes $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$, is given by

$$E_{Spring} = \frac{1}{2} k_S \left(||\mathbf{X}_{k+1} - \mathbf{X}_k|| - R_L \right)^2, \quad \text{for} \quad k = 1, 2, \cdots, N_b - 1. \tag{3.11}$$

The corresponding IB force is given by a derivative of the elastic energy in Equation (3.11) as

$$F_{Spring} = \frac{\partial E_{Spring}}{\partial \mathbf{X}_k} = k_S \left(1 - \frac{R_L}{||\mathbf{X}_{k+1} - \mathbf{X}_k||} \right) \cdot \begin{bmatrix} x_{k+1} - x_k \\ y_{k+1} - y_k \end{bmatrix}. \tag{3.12}$$

Note that there is a corresponding spring force at $\mathbf{X}_{k+1}$ in the negative direction given by the derivative $\frac{\partial E_{Spring}}{\partial \mathbf{X}_{k+1}}$.

3.4.2 Beams

A virtual beam connection is introduced between each set of three successive Lagrangian points to have a bending effect. The model assumes a desired external angle ϕ_0 which is a prescribed curvature between the three Lagrangian points along with a corresponding bending stiffness k_B. The bending energy can be written

$$E_{Beam} = \frac{1}{2} k_B (\hat{z} \cdot (\mathbf{X}_{k-1} - \mathbf{X}_k) \times (\mathbf{X}_k - \mathbf{X}_{k+1}) - C)^2, \tag{3.13}$$

where $\mathbf{X}_{k-1}$, $\mathbf{X}_k$ and $\mathbf{X} + \mathbf{1}_k$ are left, middle and right Lagrangian points coordinates, respectively, and the local curvature is $C = d_{(k-1,k)} d_{(k,k+1)} \sin(\phi_0)$ at the desired angle ϕ_0 and distances between links, $d_{(k-1,k)}$ and $d_{(k,k+1)}$. Thus, the corresponding bending force is

$$\begin{aligned} F_{Beam} = \frac{\partial E_{Beam}}{\partial \mathbf{X}_k} = k_B \left((x_{k+1} - x_k)(y_k - y_{k-1}) - (y_{k+1} - y_k)(x_k - x_{k-1}) - C \right) \\ \times \begin{bmatrix} (y_k - y_{k-1}) + (y_{k+1} - y_k) \\ (x_k - x_{k+1}) - (x_k - x_{k-1}) \end{bmatrix}, \end{aligned} \tag{3.14}$$

and there are additional contributions to the force at points $\mathbf{X}_{k-1}$ and $\mathbf{X}_{k+1}$ that come from the corresponding derivatives of the same energy term. An illustration of such a 2D beam connection is provided in Figure 3.2B.

3.5 Muscle Contraction

Jet swimming involves the contraction of circular muscle fibers lining the subumbrellar surface, called subumbrellar muscles. Shortening of the muscles contracts the bell and reduces the bell volume. This action forces fluid out of the subumbrellar region as a jet that forms a series of vortices behind the bell and drives the forward swimming motion. Virtual springs joining points on opposite sides of the bell mimic the subumbrellar muscles, which cause

the bell to contract by dynamically varying their resting lengths. While the deformation force is identical to Equation (3.12), a different spring stiffness coefficient k_{muscle} and a time-dependent spring resting length $R_L(t)$ are used. The subumbrellar muscles squeeze the bell toward the centre axis of the bell and push the fluid out during the contraction phase of forward swimming. We can mimic this contraction process by imposing a resting length of a pair of muscle points $\mathbf{X}_l$ and $\mathbf{X}_r$ on opposite sides of the bell in the following manner

$$R_L(t) = |\cos(2\pi f t)| \times \sqrt{(x_l - x_r)^2 + (y_l - y_r)^2} \tag{3.15}$$

where f is the contraction frequency (s^{-1}) and t is time. Note that during the contraction phase, the driving force term is positive, and contracts the bell towards the center line. During the expansion phase, the driving force term is negative and the force term is directed outwards from the center line.

3.6 Tracers or Food Particles

In this section, we will describe the *IB2d* implementation for tracer particles which move with the local fluid velocity without exerting any force on the fluid. Suppose that a tracer's position is denoted by $\mathbf{X}_{tr}$, then its equation of motion is simply

$$\frac{d\mathbf{X}_{tr}}{dt} = \mathbf{u}(\mathbf{X}_{tr}(t), t) = \int_\Omega \mathbf{u}(\vec{\mathbf{x}}, t)\delta(\vec{\mathbf{x}} - \mathbf{X}_{tr})d\vec{\mathbf{x}}. \tag{3.16}$$

Note that tracer points are moved in the same way as other force-bearing IB points during a simulation. Tracers are not only useful for visualizing the fluid motion during a simulation, but they can also be used to mimic the presence of passively floating prey such as plankton or algae which are the main food source for jellyfish like *Sarsia*.

3.7 Parameters and Nondimensional Numbers

Table 3.1 provides reference parameters related to our two dimensional jellyfish model. To compare our computed results with those from the literature across different stroke frequencies and sizes of jellyfish, we used results from dimensional analysis to define certain dimensionless quantities that are relevant to swimming. Defining dimensionless numbers ensures that various model results can be compared appropriately over a wide range of dimensional parameters. The three nondimensional parameters that are applicable to our study are described below.

3.7.1 Reynolds Number, Re.

The Reynolds number measures the ratio of inertial to viscous forces in the fluid and is obtained during the nondimensionalization of the Navier-Stokes equations. There are many

possible choices for length and velocity scales but we choose values relevant to swimming dynamics:

$$Re = \frac{\rho U_{jelly} D_{jelly}}{\mu}, \tag{3.17}$$

where U_{jelly} is a characteristic swimming velocity, D_{jelly} is the characteristic jellyfish length, ρ is the density of the fluid, and μ is the dynamic viscosity. The characteristic jellyfish length D_{jelly} is taken to be the diameter of the bell in its rest state configuration. Then we specify the characteristic velocity in terms of the bell contraction frequency f as $U_{jelly} = fD_{jelly}$; therefore, Equation (3.17) can be rewritten as

$$Re = \frac{\rho f D_{jelly}^2}{\mu}. \tag{3.18}$$

Jellyfish bell reference parameters			
Parameter	**Symbol**	**Value**	**SI units**
Contraction force magnitude	F_{Mag}	10^5	N
Bending stiffness	k_B	2.5×10^5	-
Spring stiffness	k_S	10^7	-
Muscle stiffness	$k_{muscles}$	10^7	-
Contraction frequency	f	1	s^{-1}
Bell radius	a	0.5	m
Bell height (top)	b	0.75	m
Bell height (bottom)	d	0.25	m
Characteristic length	$D(2a)$	1	m
Fluid property			
Parameter	**Symbol**	**Value**	**SI units**
Density	ρ	1000	kg/m^2
Dynamic viscosity	μ	6.5	Ns/m^2

Table 3.1: Reference parameters related to the jellyfish bell model, which are representative of a jellyfish within genus *Sarsia* [19].

3.7.2 Swimming Number, Sw.

The swimming number is a dimensionless number that has been described as a transverse Reynolds number characterizing the undulatory motions that drive swimming in organisms such as fish, lamprey, etc. We have adapted this definition for jellyfish and define:

$$Sw = \frac{\rho \omega A D_{jelly}}{\mu} \tag{3.19}$$

where A is the bell contraction amplitude, $\omega = 2\pi f$ is the contraction frequency, D_{jelly} is the bell diameter, ρ is the fluid density, and μ is dynamic viscosity. Note that this Sw was

defined for undulatory swimmers like fish and lamprey [14], and we are extending this to jellies to investigate whether or not Sw captures scaling in jet-like swimming.

3.7.3 Fineness Ratio, Fi

The fineness ratio is the ratio of bell height to diameter:

$$Fi = \frac{\text{bell height}}{\text{bell diameter}} = \frac{b+d}{2a}, \tag{3.20}$$

and characterizes jellyfish shapes. Whenever $Fi < 1$ the jellyfish is oblate or flat and plate-shaped in form. Oblate jellyfish predominantly use paddling motions (termed rowing) to move water around the bell margin during swimming. When $Fi > 1$ the jellyfish is prolate in shape which is more elongated or bullet-shaped. Prolate jellyfish typically swim producing a jet of water during swimming. To be consistent with other authors in the literature, we will refer to jellyfish having a fineness ratio $Fi = 1$ as "circular" in shape; however, they are not exactly circular because our parameter $a < b$. Our initial jellyfish bell geometry is chosen such that the bell's fineness ratio is roughly equal to 1 ($Fi \approx 1$).

3.8 IB2d Implementation

In this section, we briefly describe the typical workflow for using the *IB2d* code and how we have modified the code for our jellyfish model. The following data are stored at equally spaced time points during a simulation:

1. Position of Lagrangian points: $\mathbf{X}(s,t)$
2. Horizontal/vertical forces on each Lagrangian point: $\mathbf{F}(s,t)$
3. Fluid velocity: $\mathbf{u}(\mathbf{x},t) = (u(\mathbf{x},t), v(\mathbf{x},t))$
4. Fluid vorticity: $\omega(\mathbf{x},t)$
5. Fluid pressure: $P(\mathbf{x},t)$
6. Forces spread onto the fluid (Eulerian) grid from the jellyfish (Lagrangian) mesh: $\mathbf{f}(\mathbf{x},t)$.

Both MATLAB and Python versions of *IB2d* have their own respective directories which in turn contain two folders: `IBM_Blackbox` and `Jellyfish`. The `Jellyfish` folder contains all data for the current simulation, including input files needed to run each simulation. The `IBM_Blackbox` folder contains all methods for solving the fluid-structure interaction (FSI) problem. *IB2d* is designed such that the user will not have to change the underlying IB algorithm in the black-box, unless they need to make additions such as implementing more fiber models, etc.

First of all, the user needs to execute the `Jelly_Geometry` function which creates the jellyfish bell rest state configuration geometry as described in Section 3.3. All input files for each force type such as vertexes, springs, non-Invariant beams and tracers. In all simulations, the immersed structure is called "jelly" and the possible file types are as follows:

- jelly.**vertex**: A list of all initial coordinates of Lagrangian points.
- jelly.**spring**: A list of pairs of successive Lagrangian nodes (each of which is connected by a spring) along with their associated spring stiffness and resting-length.
- jelly.**beam**: A list of indices for three adjacent Lagrangian nodes (associated with a beam) and their beam stiffness and target curvature.
- jelly.**tracer**: A list of all tracer points and their initial coordinates. Tracers mimic the presence of passively floating prey.
- jelly.**user-force**: A list of all Lagrangian nodes and the user-defined force magnitude and jellyfish geometry parameters. Note that this file also keeps records of the jellyfish bell radius, semi-major axis, semi-minor axis and contraction frequency. Other desired details of the immersed structure can be kept in this file.

Inside the `Jellyfish` folder, are two essential files named `input2d` and `main2d.m`. The file `input2d` is where all parameters required for the simulation are specified by the user including fluid parameters, temporal information, grid parameters, the Lagrangian structure model and printing options. The `main2d` script reads the `input2d` file and then reads in the corresponding input files associated with the choices selected in `input2d`.

After setting desired parameters and selecting the necessary flags, `main2d.m` this information to the `IBM_Driver` script. Once the simulation finishes, a visualization folder (`viz_IB2d`) is created that contains all the Lagrangian structure and dynamical data from the simulation in `.vtk` format. The `.vtk` format files are visualized using VisIt [7]. A more detailed discussion of the workflow in *IB2d* can be found in [5].

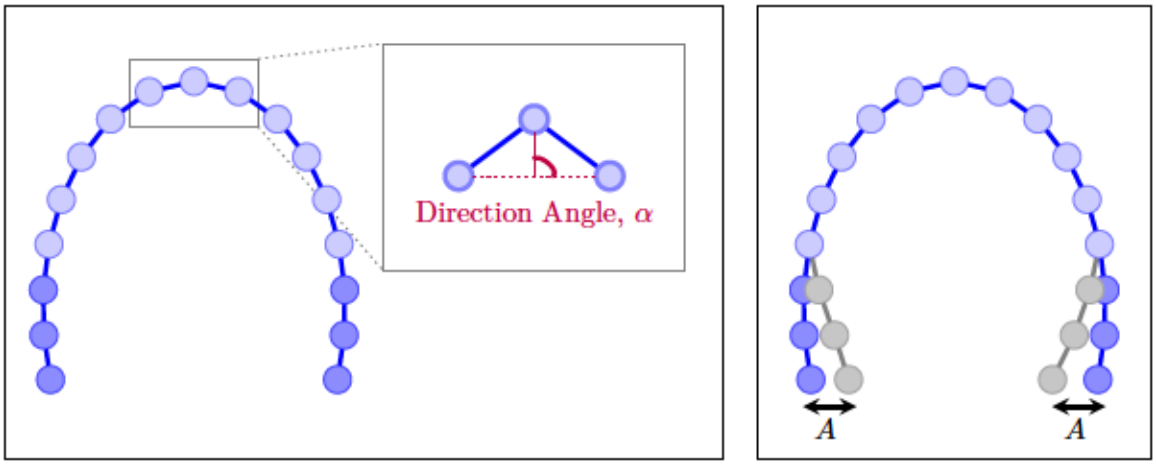
The remainder of this section describes the functions created to help interpreting the computational results of simulations. Along with the functions listed below.

The `Give_Me_Jellyfish_Details` function implemented in the `Jellyfish` folder, and its associated modifications made in the time-stepping part of the `IBM_Blackbox` where necessary to derive jellyfish swimming parameters such as swimming direction angle, swimming velocity magnitude, bell amplitude, and bell diameter as a function of time. This function computes the mentioned parameters during each time step, and itself is made up of four different functions which are listed below:

- `Give_Me_Jellyfish_Swimming_Angle`: Calculates the angle of the jellyfish swimming direction in relation to the angle of the bell's central axis. Note that to minimize the oscillations of the swimming direction angle, which may be caused by small oscillations

of Lagrangian points due to rapid local changes in spring and beam forces, the central axis of the bell is assumed to be the perpendicular bisector of the line joining the symmetric pair of Lagrangian points that flank the uppermost point, see Figure 3.3a.

- `give_Me_Jellyfish_Velocity_Magnitude`: Calculates the velocity magnitude of Lagrangian nodes by taking the average of IB points velocity.
- `give_Me_Jellyfish_Amplitude`: Computes the jellyfish bell amplitude, Figure 3.3b depicts bell amplitude during contraction phase.
- `give_Me_Jellyfish_Diameter`: Computes the bell diameter during simulation by measuring the distance between the lowermost points of the bell.



(a) (b)

Figure 3.3: (a) The illustration of how we compute the swimming direction angle α in the `Give_Me_Jellyfish_Swimming_Angle` function (We considered a very special case $\alpha = \pi/2$ for most of our simulations). (b) The illustration of jellyfish bell amplitude during the contraction phase. The dark blue IB points correspond to the bell margin points at rest state and the gray ones corresponding the bell margin points in the deformed state.

In addition, we were also interested to investigate jellyfish feeding performance, so we added a function `Check_Tracers_In_Bell` to keep track of tracer points, and determine whether they have entered the region inside the bell during a simulation. We used a flag for each tracer point which indicates whether or not the prey particle has been eaten by the jellyfish. A more detailed discussion of feeding performance can be found in Chapter 5.

An additional force term is applied to address unrealistic interactions between jellyfish as they collide, preventing unnatural looking motions. To implement this force term, we use the *IB2d* option of creating a user-defined force model. Many items are passed into the model automatically, such as the current and previous positions of the Lagrangian points, current time, time-step, and so on, and the user has control over how they define their

model and what Lagrangian points are involved. It also includes the functionality for the user to read in appropriately chosen data for parameters, etc.

The `user_defined_force` functionality works similar to the other fiber models, for instance, there is an associated input file that gets read into the `IBM_Driver` file and then finally into a function that computes the deformation forces at each time step. The style of the input file `user_defined_force` is shown in [5]. The input file data then gets passed to a script `give_Me_General_User_Defined_Force_Densities` which is located in the `jelly` folder. That script receives the input parameters listed in Table 3.2 from the `IBM_Driver` file. With these parameters and the data read in from the `jelly.user_force` file, the user can

Δs	Lagrangian spacing
Nb	Number of Lagrangian points
$xLag$	current x-Lagrangian coordinate positions
$yLag$	current y-Lagrangian coordinate positions
L_x	Computational domain size in x-direction
L_y	Computational domain size in y-direction
$general - force$	matrix containing all data from the `jelly.user_force`

Table 3.2: Parameters passed into the user-defined force script.

define their Lagrangian deformation force law. Upon running the simulation, during each time-step, the data is passed to the `give_Me_General_User_Defined_Force_Densities`. Here, that script computes a force. The entire procedure can be summarised as follows:

1. Create input files `jelly.user-force`.
2. Define user-defined force model in the `give_Me_General_User_Defined_Force_Densities` script.
3. Turn the flag for the `user-defined` force model in the `input2d` file.
4. Run simulation using `main2d.m`.

A detailed discussion of the force term and its implementation is given in Chapter 6.

For visualizing the simulation results we used a combination of MATLAB [23] and the open-source visualization software VisIt [7].

Chapter 4

Scaling Jellyfish Locomotion

In this chapter, we follow the work of Gazzola et al. [14] to test whether or not jellyfish that utilize jet swimming satisfy the same scaling relationship for aquatic organisms that employ undulatory swimming gaits. As a result, we extend their scaling relationship to connect jellyfish swimming speed (U) to jellyfish bell contraction amplitude (A) and frequency (ω), which reveals a unifying dimensionless equation describing their locomotion. Using the definition of the Reynolds number ($Re = \rho U L/\mu$) and swimming number ($Sw = \rho\omega A L/\mu$) discussed in Section 3.7, this principle can be expressed as a power law:

$$Re \sim Sw^{\alpha}. \tag{4.1}$$

Note that we are interested in confirming whether such a power law applies to jet-propelled swimmers and comparing to the exponent α observed for undulatory swimmers.

It is worth emphasizing that the underlying dynamics of aquatic swimmers are often described using the Strouhal number ($St = \omega A/U$), a dimensionless variable taken from engineering. Even though this makes sense in many engineering applications, such as vortex shedding and vibration, Strouhal number confuses input (A, ω) and output (U) variables in a biological context, captures only one length scale by assuming $A \sim L$, and does not take into account the effect of the fluid environment through different values of fluid viscosity. In contrast, swimming number captures the two usually distinct length scales for the stroke amplitude and body size, making it a more natural quantity to describe biological locomotion. This also allows us to directly connect the swimming dynamics (A, ω) with the resulting swimming velocity (U).

4.1 Scaling Aquatic Locomotion

An aquatic swimmer's locomotion is characterized by a complicated interaction between the swimmer's deforming body and the induced flow in the surrounding fluid. Swimming at low Reynolds number ($Re \ll 1$) is governed by linear hydrodynamics and is more analytically tractable, but at high Reynolds number ($Re \gg 1$) locomotion involves non-linear inertial

forces. The wide range of sizes, morphologies, and gaits of aquatic swimmers makes it challenging to find a unifying framework that holds across many different species. As a result, most researchers have taken a more restricted approach, assessing the challenge of swimming in specific settings from experimental, theoretical, and computational viewpoints.

First, we start with a brief review of Gazzola et al. [14], who combined simple scaling arguments, numerical simulations, and a comprehensive comparison with experiments to provide new insights into undulatory swimming. They begin by recalling the primary physical mechanism underlying the inertial motion of a slender swimmer length, tail beat frequency and amplitude, swimming speed and fluid viscosity. See Figure 4.1b illustrating an undulatory swimmer (fish) of characteristic length L that is propelled forward with velocity U by pushing a bolus of water by means of body undulations characterized by tail beat amplitude A and frequency ω. The body-induced fluid acceleration generates inertial thrust balanced by hydrodynamic resistance at high Reynolds numbers ($Re \gg 1$). Gazzola et al. suppose that the tail amplitude is minimal in comparison with the body length and that the organism's body is slender. This suggests that fluid acceleration can be converted into longitudinal thrust. Furthermore, all quantities are described per unit depth since undulatory motions are considered in the plane. For inertial aquatic swimmers that use undulatory gaits, the mass of fluid set in motion by the deforming body scales as ρL^2 per unit depth, assuming that the wavelength associated with the undulatory motions scales with the body length (L), which is compatible with experimental and empirical data. The surrounding fluid acceleration is scaled as $A\omega^2$, and hence the fluid reaction force on the swimmer is scaled as $\rho L^2 A\omega^2$. The effective thrust $\rho\omega^2 A^2 L$ is obtained when the body makes a local angle with the direction of motion that scales as A/L. The viscous resistance to motion (skin drag) per unit depth scales as $\mu UL/\delta$, where δ is the thickness of the boundary layer. For fast laminar flows, the classical Blasius theory predicts $\delta \sim LRe^{-1/2}$ [16], hence the skin drag force due to viscous shear scales as $\rho(\nu L)^{1/2}U^{3/2}$. Balancing thrust and skin drag yields the relation $U \sim A^{4/3}\omega^{4/3}L^{1/3}\nu^{-1/3}$ which may be rewritten as

$$Re \sim Sw^{4/3}. \tag{4.2}$$

When the boundary layer around the body becomes turbulent at very high Reynolds numbers ($Re \sim 10^3 - 10^4$), pressure drag overtakes skin drag. The corresponding force scales as $\rho U^2 L$ per unit depth, which when balanced by the thrust yields

$$Re \sim Sw. \tag{4.3}$$

Equations (4.2) and (4.3) can be restated in terms of the Strouhal number to obtain

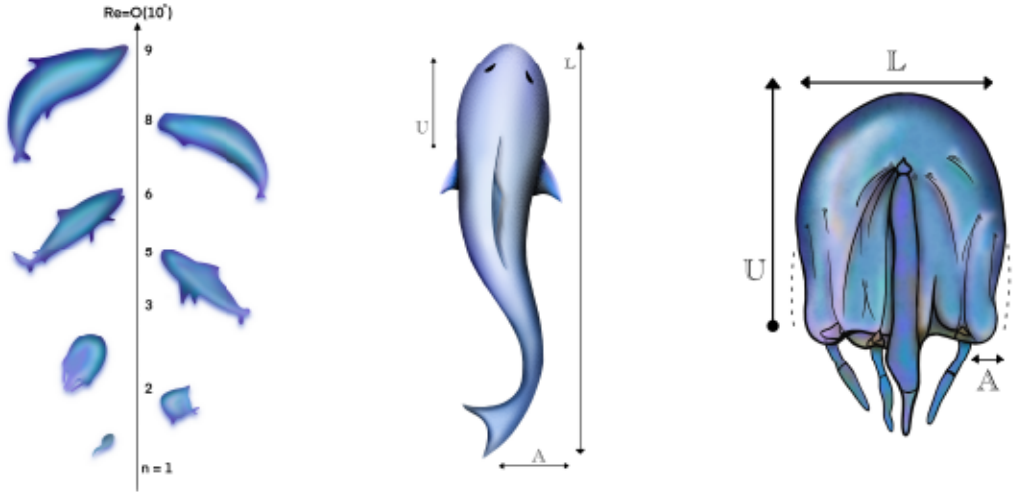
$$Sw = Re \cdot St, \tag{4.4}$$

which means that for laminar flows, $St \sim Re^{-1/4}$, and for turbulent flows, $St \sim Re$, indicating little or no effect of the Reynolds number on the Strouhal number.

To validate their proposed power laws (4.2) and (4.3), Gazzola et al. [14] show that data from over 1,000 measurements on fish, amphibians, larvae, reptiles, mammals, and birds, as well as direct numerical simulations, are consistent with their scaling relationships. Interestingly for us, jellyfish were not included in this comparison. Figure 4.1a depicts the diverse range of organisms covered by Gazzola et al. including, which span eight orders of magnitude in Reynolds number.

Our aim is to extend Gazzola et al.'s work to characterize jellyfish with jet-like swimming over a much more limited range of Re. We accomplish this by developing a scaling relationship that connects the swimming speed to the body dynamics and the fluid parameters. Note that the amplitude of the bell motion is relatively small compared to the length of the jellyfish (compared with undulatory swimmers). The jellyfish body is not slender, but despite these differences, the symmetric muscle contraction that induces jetting induces a backward jet in the (negative) longitudinal direction so that acceleration is still be effectively channelled into longitudinal thrust.

Figure 4.1c shows a jellyfish (*Sarsia*) with L equal to the bell diameter (D_{jelly}) that is propelled forward with swimming velocity U by contraction of circular muscle fibers lining the subumbrellar surface of jellyfish characterized by bell contraction amplitude A.



(a) (b) (c)

Figure 4.1: (a) List of aquatic organisms ordered according to Reynolds number. Characteristic scales for (b) an undulatory swimmer (inspired by Figure 1 from [14]) and (c) a jellyfish (*Sarsia*) that swims by jetting.

4.2 Computing Swimming Number in IB2d

To obtain values of Sw from simulations, we added code to save the jellyfish geometry details, such as bell radius, semi-minor axis and semi-major axis, and also compute the bell amplitude during the contraction phase. Moreover, we added lines of code right after the time-stepping section in the black-box to compute and print the jellyfish Reynolds number and swimming number. Note that we choose the maximum bell amplitude value observed during an entire simulation for computing the swimming number. This is because we assume that the jellyfish bell amplitude is minimal compared to the characteristic length.

4.3 Results

To uncover the scaling law for jellyfish swimming, we performed 30 2D simulations of single jellyfish swimming vertically in a rectangular domain with dimensions $L_x = 3m$ and $L_y = 6m$ over a computational time of $T = 2s$. The grid spacing is set to be $\Delta x = 6/320$. The fluid viscosity is $\mu = 1\ Ns/m^2$ and density $\rho = 1000\ kg/m^2$. Note that the jellyfish bell contraction frequency equals 1 for all simulations, thus $\omega = 2\pi$. The jellyfish vary in size and fineness ratio. Details of the initial jellyfish bell geometry are provided in Table 4.1. The methodology of creating the initial bell geometry is the same for all simulations and is determined via the parameterization proposed in Chapter 3. The jellyfish vary in diameter from 0.6 to 1 metres [19, 3] in our 2D numerical simulations and are characterized by a swimming number and Reynolds number spanning the range $150 < Sw < 650$ and $350 < Re < 1000$.

Note that to uncover the power law holding for jellyfish with jet-like swimming, we can rewrite Equation (4.1) as

$$Re = c \cdot Sw^{\alpha}. \tag{4.5}$$

Taking the logarithm of both sides yields

$$\log(Re) = \log(c) + \alpha \log(Sw), \tag{4.6}$$

where α and $\log(c)$ represent the slope and intercept of the best fit line on a log-log scale. To determine the parameters in this scaling relationship we perform a linear least squares fit to data computed for swimming number and Reynolds number.

Since our dimensionless scaling relationship depends heavily on the characteristic length scale L, we tested our results for two choices:

- bell diameter $L = D = 2a$.
- an average length scale computed as $L_{av} = \frac{2a+b+d}{2}$, where a and b are the semi-minor and semi-major axis of the jellyfish bell, and d is the height of the lower portion of the partial ellipse, as shown in Figure 3.1b.

Fi	Simulation	a	b	d	Re	Sw
Circular $Fi \approx 1$	1	0.5	0.75	0.25	153.846	42.638
	2	0.45	0.675	0.225	124.615	33.418
	3	0.4	0.6	0.2	98.461	25.656
	4	0.35	0.525	0.175	75.384	19.099
	5	0.3	0.45	0.15	55.384	18.203
	6	0.48	0.72	0.24	141.784	37.793
	7	0.36	0.54	0.18	79.753	20.216
	8	0.42	0.63	0.21	108.553	27.885
	9	0.33	0.495	0.165	67.015	17.294
	10	0.39	0.585	0.195	93.600	24.112
Oblate $Fi < 1$	11	0.5	0.6	0.2	153.846	43.543
	12	0.45	0.6	0.2	124.615	34.231
	13	0.5	0.45	0.15	153.846	48.138
	14	0.4	0.45	0.15	98.461	28.233
	15	0.45	0.45	0.15	124.615	37.786
	16	0.5	0.675	0.225	153.846	42.676
	17	0.5	0.525	0.175	153.846	46.705
	18	0.45	0.525	0.175	124.615	36.358
	19	0.35	0.45	0.15	75.384	20.119
	20	0.4	0.525	0.175	98.461	27.347
Prolate $Fi > 1$	21	0.45	0.75	0.25	124.615	28.591
	22	0.4	0.75	0.25	98.461	24.246
	23	0.35	0.6	0.2	75.384	17.494
	24	0.3	0.75	0.25	55.384	13.092
	25	0.3	0.6	0.2	55.384	15.839
	26	0.35	0.75	0.25	75.384	17.154
	27	0.3	0.525	0.175	55.384	17.673
	28	0.4	0.675	0.225	98.461	25.239
	29	0.35	0.675	0.225	75.384	17.438
	30	0.3	0.675	0.225	55.384	14.848

Table 4.1: Parameter for two-dimensional jellyfish simulations.

Note that to find the best fit to data and compare them to Gazzola et al. [14], we used the coefficient of determination R^2 which measures how well a fit can predict the data. The coefficient of determination falls between 0 and 1, and the quality of the fit increases as R^2 gets closer to 1. The coefficient of determination is

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{4.7}$$

where $\hat{y}_i$ is a calculated data, and $\bar{y}$ is the mean for y data.

Figure 4.2 displays the best-fitting outcomes for both version of characteristic lengths, D and L_{av}. A linear fit for the case with $L = D$ yields the power law $Re = 2.93Sw^{0.91}$

with $R^2 = 0.95$, and for $L = L_{av}$ yields the power law $Re = 10.0Sw^{0.69}$ with $R^2 = 0.62$. The fitting errors and comparison with later results in both cases suggests that choosing the bell diameter as characteristic length is more suitable to use for a definition of jellyfish swimming number.

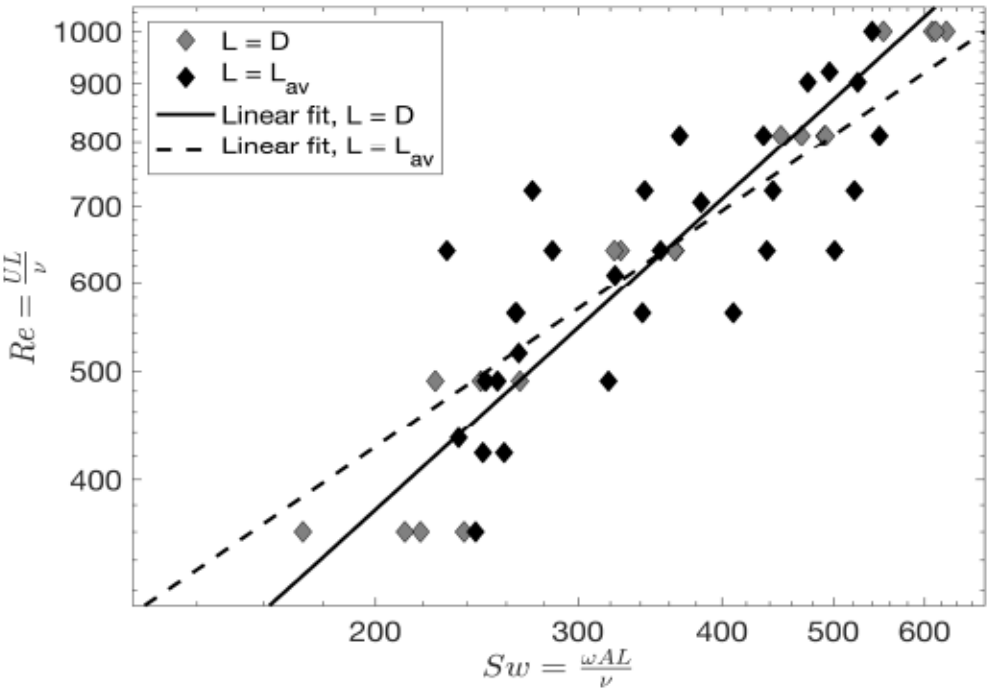


Figure 4.2: Scaling for different characteristic lengths: D and L_{av}.

Figure 4.3 shows a more refined picture of three categories of jellyfish separated by fineness ratio as follows: prolate ($Fi < 1$), circular ($Fi \approx 1$) and oblate ($Fi > 1$). Figure 4.3 depicts the data from simulations considering $L = D$, a linear fit for circular jellyfish yields a power law $Re = 1.32Sw^{1.05}$ with $R^2 = 0.97$, for prolate jellyfish yields $Re = 1.99Sw^{0.98}$ with $R^2 = 0.89$, and for oblate jellyfish $Re = 3.79Sw^{0.86}$ with $R^2 = 0.97$. According to the exponent of the linear fit for each of the three categories and their relative errors, circular jellyfish ($Fi \approx 1$) exhibit a scaling relationship that is closest to what Gazzola et al. [14] observed for undulatory swimmers in the turbulent regime ($Re \sim Sw$).

To evaluate a wider range of Reynolds numbers, we changed the fluid viscosity μ rather than changing other parameters, which could affect the grid size and time step, and thus the computational cost. Figure 4.4 depicts the power law fitting for our simulations considering two values of fluid viscosity, a linear fit for $\mu = 1$ yields $Re = 2.93Sw^{0.91}$ with $R^2 = 0.95$, for $\mu = 3$ yields $Re = 2.71Sw^{0.91}$ with $R^2 = 0.96$. These results emphasize that the scaling results are insensitive to changes in μ.

To provide an illustration of the fluid viscosity effect on swimming dynamics, Figure 4.5 is provided which depicts the vorticity field ($\omega = \nabla \times \mathbf{u}$) that the swimming jellyfish generates in the flow considering five different values of fluid viscosity. We observe that as

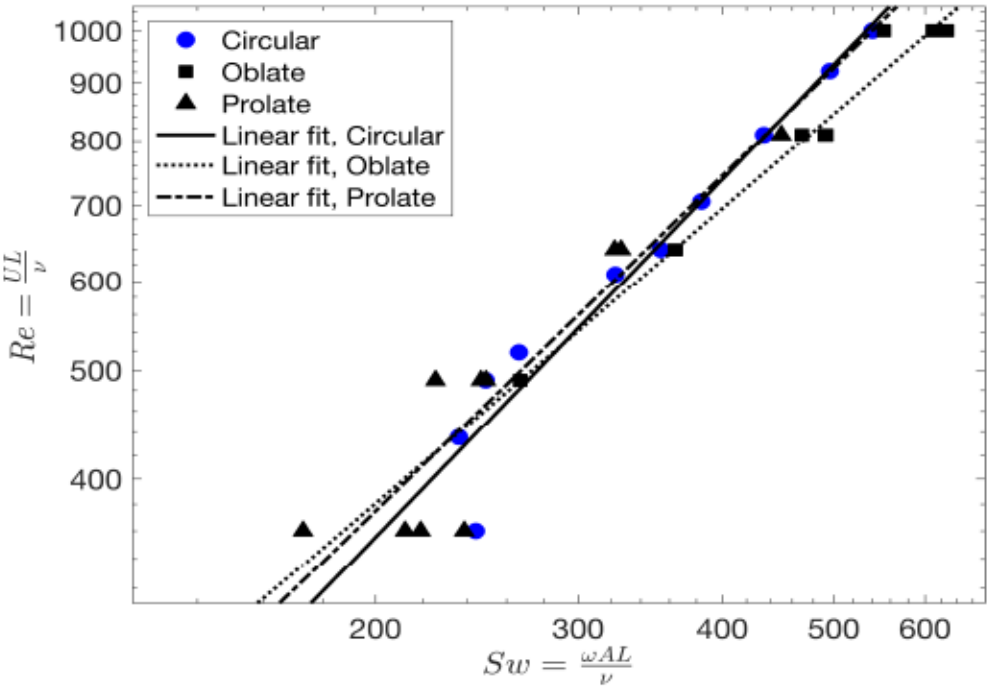


Figure 4.3: Effect of fineness ratio on the jellyfish scaling law with $L = D$.

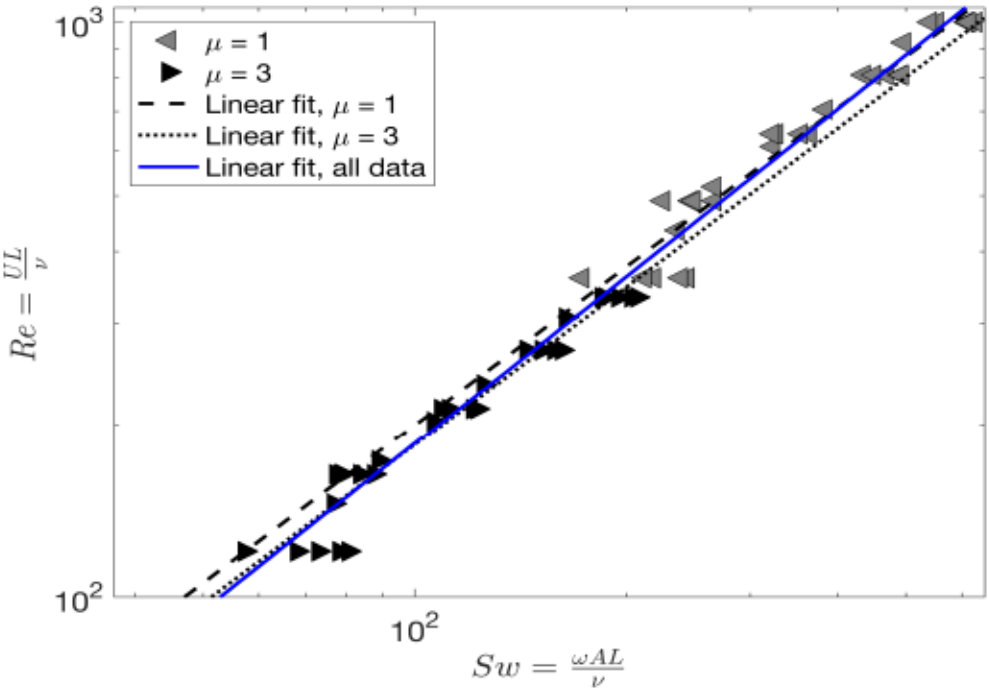


Figure 4.4: Various fluid viscosity. Data of 2D jellyfish simulations for $\mu = 1$ and $\mu = 3$.

μ increases, the swimming velocity decreases and the generated trailing line of vortices is shorter.

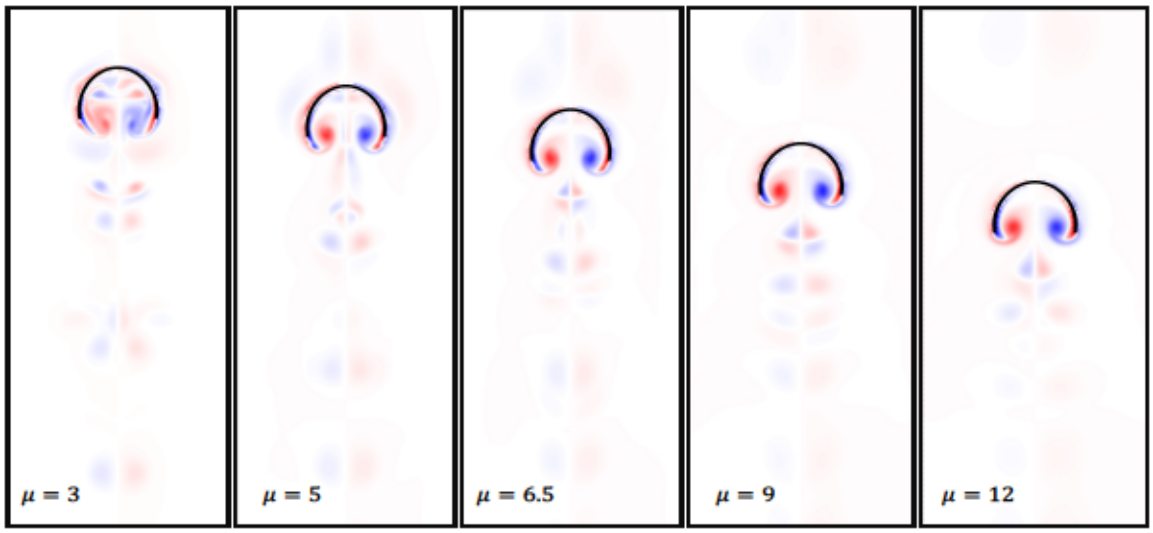


Figure 4.5: Each screenshot shows counter plot of ω generated by simulations at time = 5s.

To summarize, we reproduced Figure 2a from [14] in Figure 4.6 and added direct data from our two-dimensional jellyfish simulations in blue. Figure 2 from [14] is a plot of all the data in terms of Reynolds numbers and swimming number from more than 1,000 measurements for fish (ranging from zebrafish larvae to stingrays and sharks), amphibians (tadpoles), reptiles (alligators), marine birds (penguins), and large mammals (ranging from manatees and dolphins to belugas and blue whales). The size of the organisms ranged from 0.001 to 30 meters. A power law fit for undulatory swimmers results in $Re = 0.03Sw^{1.31}$ with $R^2 = 0.95$ for the laminar regime, and $Re = 0.4Sw^{1.02}$ with $R^2 = 0.92$ for the turbulent regime. The power law fit of our jellyfish simulation data (shown in blue) ranging between 0.6 and 1 metre results in $Re = 2.93Sw^{0.91}$ with $R^2 = 0.94$, and suggests that our results are closer to Gazzola et al.'s turbulent case because of the close match in slope (exponent).

Given that the critical Reynolds number (Re_c) at which a transition from laminar to turbulent flow can occur is highly dependent on the flow geometry, it is not surprising that undulatory swimmers may have a different Re_c than jetting swimmers such as jellyfish. For instance, $Re_c \sim 2 \times 10^5$ for flow over a cylinder, $Re_c \sim 2.3 \times 10^3$ for flow in a cylindrical pipe with no obstruction, and $Re_c \in (10^3, 2 \times 10^5)$ for flow over various NACA airfoils.

In addition, we reproduced Figure 3a from [14] in Figure 4.7 and added data from our 2D circular jellyfish simulations in blue. Figure 3a from [14] displays the outcomes of 2D anguilliform swimmers and 3D simulations carried out by several groups using diverse numerical approaches to further test their theoretical scaling relationship. Slopes in Figure 4.7 are match but the vertical shift could be explained by a variety of factors, most importantly that the jellyfish use a completely different mode of locomotion (jet-like motion) than anguilliform swimmers.

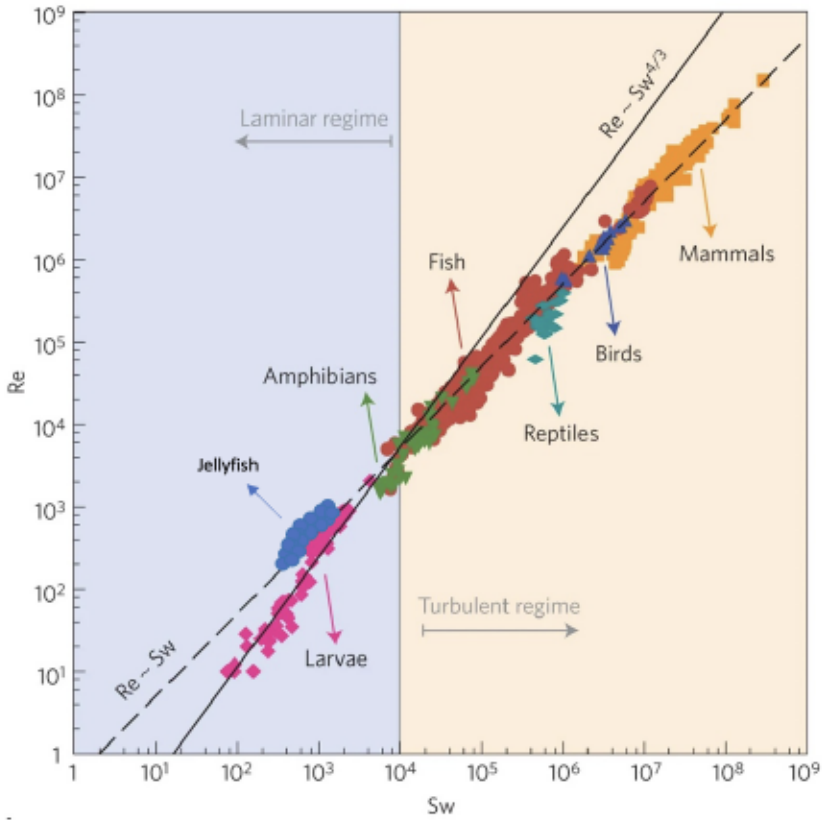


Figure 4.6: Scaling undulatory swimmers and jellyfish locomotion. This figure is reproduced from [14] and our jellyfish data is added as blue circles superimposed on the plot.

4.4 Conclusion

Our results suggest that jet-like swimmers like jellyfish do not follow the scaling laws proposed by Gazzola et al. [14] for undulatory swimmers strictly. We attribute these variations to the fact that jellyfish adopt an intermittent type of locomotion that involves a combination of acceleration, steady swimming, and coasting, such as a quick jet phase followed by a recovery period during which the bell returns to its equilibrium state. Other explanations for the variances may be related to various gaits or anatomical changes related to the body, tail, and fins which may influence the hydrodynamic profile directly or indirectly by changing the gaits.

Moreover, our numerical data set does not cover the complete range of sizes, shapes and swimming dynamics for jellyfish species. Further studies can look at the simulations of jellyfish in three dimensions and examine a wider variety of jellyfish species, sizes and shapes.

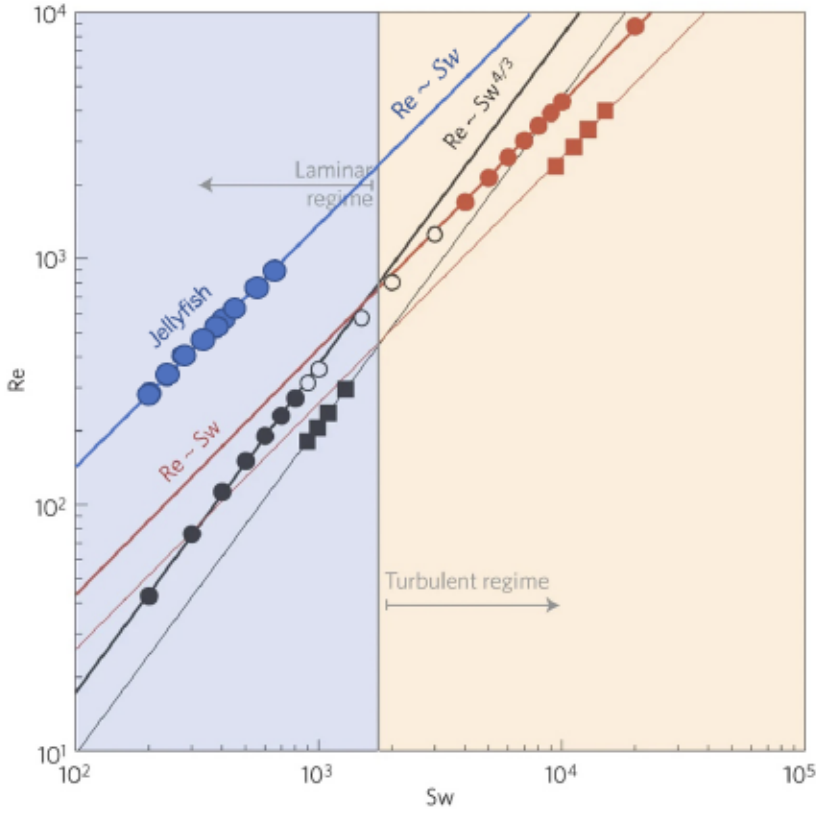


Figure 4.7: Scaling locomotion for numerical simulations. The circles and squares correspond to 2D and 3D undulatory swimmers simulations from [14], while the blue circles correspond to our 2D circular jellyfish simulations.

Chapter 5

Jellyfish Feeding

In this chapter, we will propose a simple model for jellyfish feeding to examine their feeding dynamics and efficiency using the two-dimensional model presented in chapter 3. Our aim is to investigate the role of various parameters and prey distribution patterns on jellyfish feeding performance, which in practice is difficult to do using lab experiments, but relatively easy in simulations. Several previous experimental studies have attempted to analyze jellyfish feeding behaviour [1, 30], but there has not yet been a thorough computational study of prey capture by a swimming jellyfish. We will focus on the computational assessment of passive prey capture using the numerical data from simulations (note that "passive" refers to prey that do not swim themselves). The basis of jellyfish swimming is the flexible bell's repetitive contraction over short time intervals, followed by an unforced bell expansion over a longer rest or recovery period. To evaluate jellyfish feeding ability, it is necessary to study the fluid-structure interaction (FSI) and the associated flow patterns or vortex structures.

5.1 Literature Review

The ability of jellyfish to capture food occurs in two stages: first, they come into near contact with potential prey through natural swimming motions (we are not considering prey searching or active predation); and second, the pressure and momentum of the swimming-induced fluid dynamics is exploited to capture prey by entraining it within the bell where the mouth is located. Jellyfish frequently use their self-generated swimming currents to capture stationary prey. Based on morphological and behavioural traits, jellyfish can be divided into two functional types: (1) traveling predators, which actively create a feeding current to bring prey into contact with the tentacles, and (2) ambush predators, which stretch their tentacles and wait for an opportunity to capture prey. Note that tentacles are not explicitly treated in our simulations, since jellyfish in the *Sarsia* genus have tentacles that are relatively thinner, shorter and fewer in number than other species. Moreover, *Sarsia* belongs to the first type jellyfish called traveling predators, which is another reason we ignore tentacles.

In general, jellyfish use a primitive prey capture mechanism that necessitates direct contact with the prey, whereas fish rely more on effective visual detection and active hunting. Acuna et al. [1] have used a large collection of published data to show that despite their primitive nature, jellyfish have prey clearance and respiration rates comparable to that of fish, as well as similar growth and reproduction potential. Furthermore, their results show that jellyfish have evolved relatively large, water-filled bodies that help to boost prey contact rates. Feeding differences between visual and tactile predators have been suggested as the reason for differences in competing capacities between fish and jellyfish. Fish have compact bodies and detect prey via their eyes. On the other hand, many jellyfish are passive hunters that use their bells to create vortical feeding currents, carrying prey to within reach of their tentacles and the prey capture region near their mouth inside their bell. A predator's competitive ability is determined by prey capture, ingestion rates and how well the energy collected is converted into body growth and reproduction.

Acuna et al. [1] proposed an approach to model the feeding performance, in which they assume that total fluid volume cleared of prey by a traveling jellyfish is proportional to swimming velocity U and projected cross sectional area S (see Figure 5.1a), such that

$$C = \beta SU. \tag{5.1}$$

Here, β is the searching efficiency which is the ratio of fluid volume cleared from prey to fluid volume perturbed by jellyfish. Solving Equation (5.1) for β results in

$$\beta = \frac{C}{SU}, \tag{5.2}$$

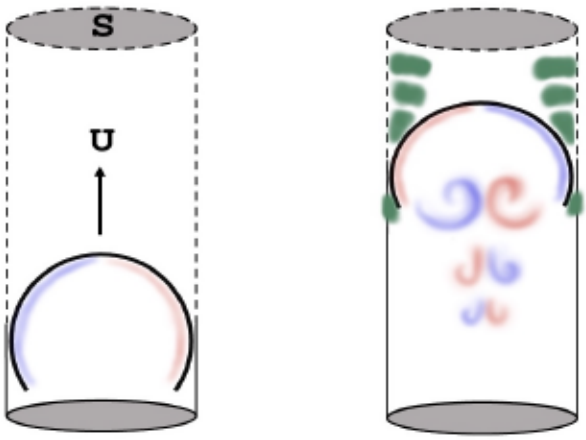
which varies between 0.02 and 2.7 for jellyfish, whereas varies between 3 and 5 for fish [1].

Titelman et al. [30] have suggested a simple method of analyzing jellyfish feeding performance. They quantified rates for *Aurelia aurita* feeding on the yolk sac of cod larvae in a series of experiments. Their results depict that the ingestion rates increase linearly in a short-time experiment and over a wide range of prey concentrations, leading to comparable clearance rates for all concentrations. Note that Titelman et al. [30] considered homogeneous prey distribution for the incubation containers at the start of their experiments by adding the prey to the jellyfish containers and mixing carefully. They defined the clearance rate ($m^3\ h^{-1}$) as follows:

$$F = \frac{V}{t} \times \ln\left(\frac{C_{start}}{C_{end}}\right), \tag{5.3}$$

where V is the container volume (m^3), t is time (h) and C_{start} and C_{end} are the prey concentrations (m^{-3}) at the start and the end of the experiment, respectively.

Note that Titelman et al. [30] have chosen this particular definition among many other possible fits in the literature because it gives a good fit with their experimental data, but they provide no justification for using the natural logarithm term. We choose the same form



(a) Volume perturbed. (b) Volume cleared.

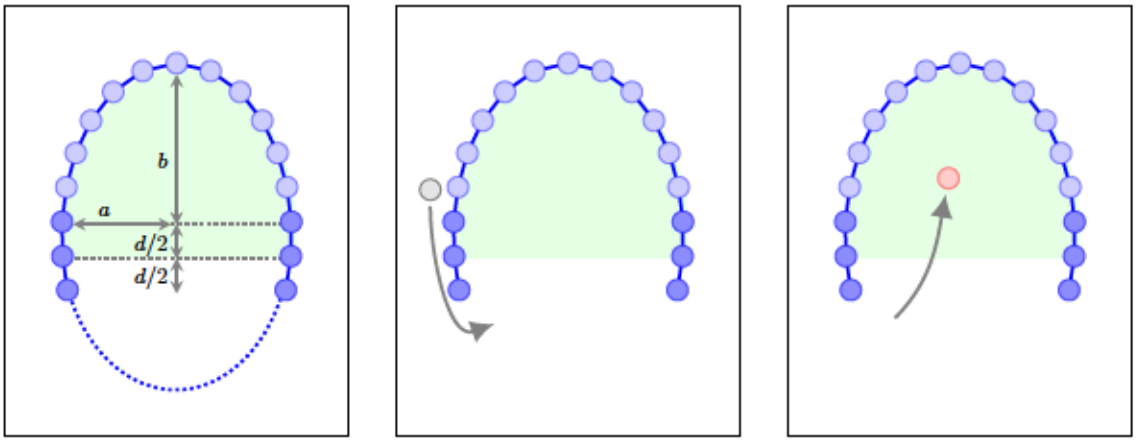
Figure 5.1: (a) The fluid volume that a jellyfish swims through can be approximated by a cylinder with dimensions determined by the projected cross sectional area S and its velocity U. (b) Only a portion of the cylinderical fluid region actually interacts with the bell margin and facilitates prey capture (volume shaded in green).

for reporting F since it allows us to compare with their results on feeding in another species (*Aurelia aurita*).

5.2 Simulations

Inspired by Equation (5.3), we implement a method to compute the clearance rate of a swimming jellyfish during simulations. We simulate feeding by adding tracer points in the fluid that mimic the presence of passively floating prey like plankton or algae. Tracer points are one of the standard immersed boundary point types in `IB2d` discussed in Chapter 3.

To estimate the number of prey captured by a swimming jellyfish during a simulation, we check the location of each tracer point in every time step to see whether or not it lies inside the prey capture region. We implement a function for this purpose that takes as input the Lagrangian point coordinates, tracer point coordinates, and an array of labels for the tracers. Figure 5.2 provides an illustration of how the labelling procedure works. Figure 5.2a shows the prey capturing region of the jellyfish in green, Figure 5.2b shows a tracer point outside the capturing region while its corresponding label is "FREE", and Figure 5.2c shows a tracer point inside the capturing region while its corresponding label changed is from "FREE" to "CAPTURED". Note that any tracer point marked as "CAPTURED" remains so for the rest of the simulation, even if the tracer point is ejected from the bell later on.



(a) (b) (c)

Figure 5.2: (a) Prey capturing region shaded in green. (b) Tracer point shown in gray lies outside the bell and is labelled "FREE". (c) Tracer point shown in red is drawn inside the bell capturing region and label is updated to "CAPTURED".

5.3 Results

We considered a rectangular computational fluid domain with $L_x = 3$, $L_y = 9$ and with periodic boundary conditions. The grid spacing $\Delta x = \Delta y$ is fixed as $L_x/N_x = 3/160$ for all simulations and the time-step is $\Delta t = 4 \times 10^{-5}$. Other numerical parameters are the same as Table 3.1 and the initial jellyfish bell geometry is the same as Table 4.1.

5.3.1 Prey Distribution

To study the effect of prey distribution on feeding efficiency, we performed three sets of simulations with different patterns of prey distribution. We are visualizing feeding performance as shown in Figures 5.3, 5.4, and 5.5 since the first pattern is homogeneous and is consistent with previous experiments in [1], the second pattern has a narrow horizontal prey strip which can be thought of as approximating a concentrated region of prey that a jellyfish swims through, and the third pattern is an analogous vertically oriented strip that the jellyfish swims alongside to investigate the ability of swimming vortices to entrain nearby prey. The simulations for homogeneously distributed prey performed with prey concentration of 400 (prey m^{-3}) as shown in Figure 5.3. Figure 5.4 depicts the simulations consisting of a horizontal strip of prey with a prey concentration of 232 (prey m^{-3}) and the simulations with a vertical strip with a concentration of 464 (prey m^{-3}) shown in Figure 5.5.

Note that to be consistent across all simulations, the initial distance between the jellyfish bell and the prey strip has been chosen the same. We also assumed $L_z = L_x = 3$ for computing the container volume to compare our 2D numerical results with experiments in [1], therefore, taking $V = 54\ m^3$ for all simulations.

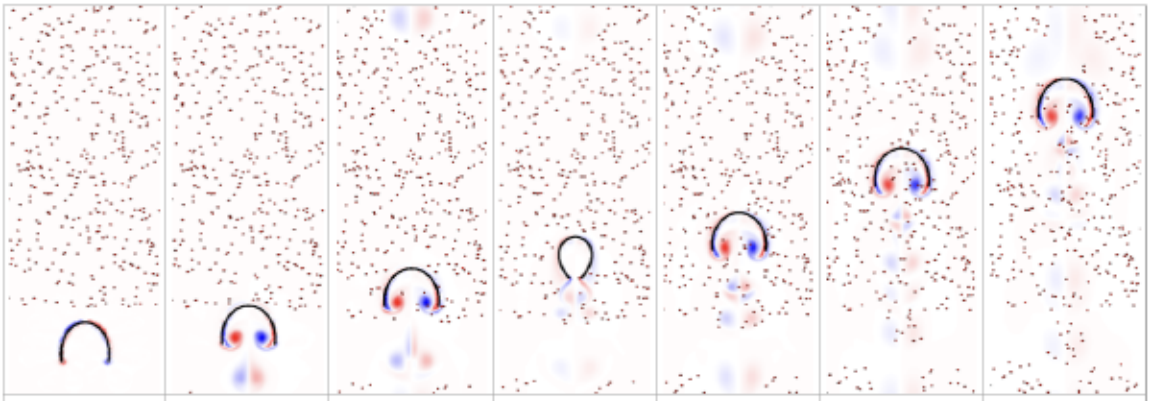


Figure 5.3: Screenshots of swimming jellyfish through homogeneously distributed prey.

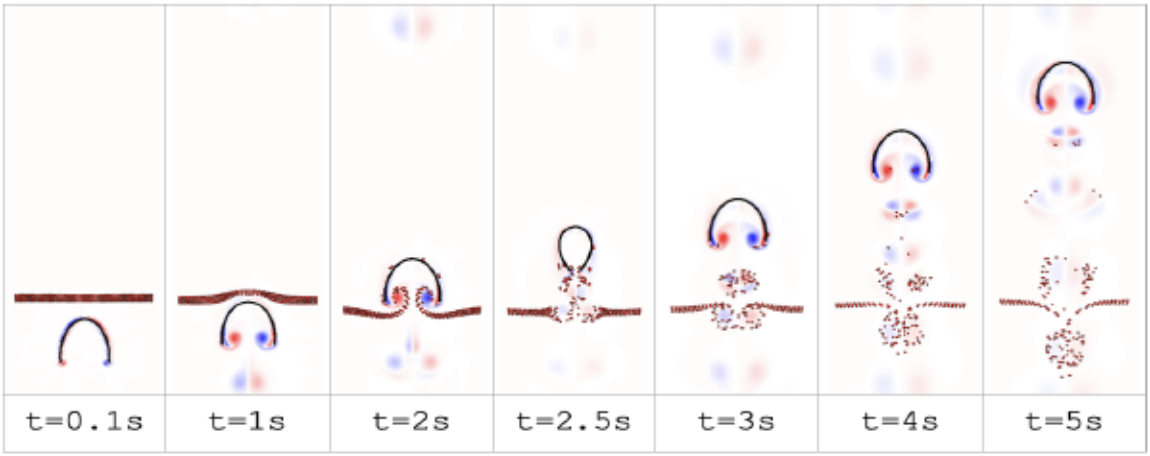


Figure 5.4: Screenshots of swimming jellyfish through a horizontal block of prey.

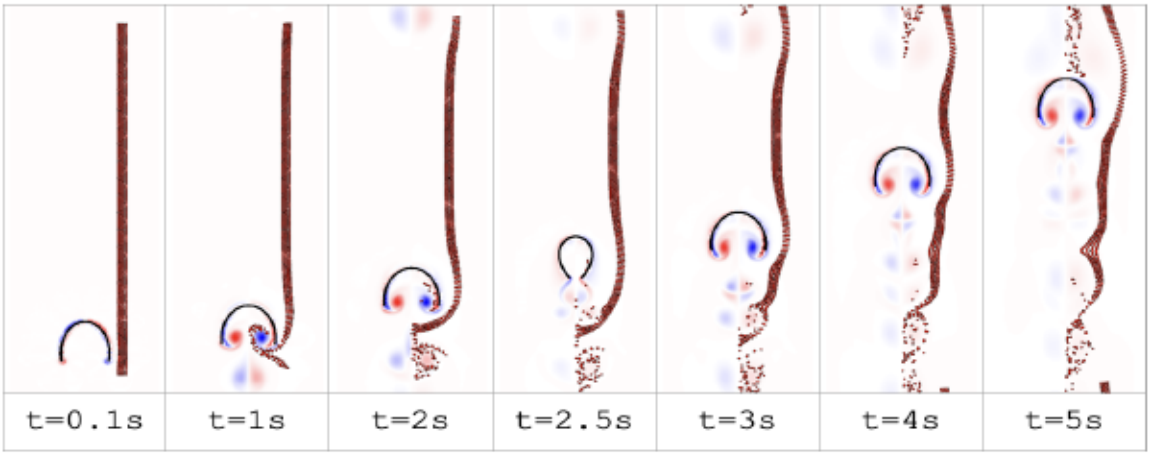


Figure 5.5: Screenshots of swimming jellyfish alongside across a vertical block of prey.

5.3.2 Jellyfish Shapes

Figures 5.6 to 5.8 illustrate the role of various prey distributions on the feeding dynamics while considering three different bell shapes determined by fineness ratio denoted as Fi. Recall that for $Fi < 1$, the jellyfish is called oblate; for $Fi > 1$, the jellyfish is called prolate; and for $Fi = 1$, the jellyfish is called circular. Note that we use the relative mean square error ($rRMS$), a standard least square fitting error measure, to test the effectiveness and accuracy of the linear fitting performed on simulation results; and we left out the fits for prolate jellyfish since the data points are too disperse and lack any clear trend. The initial bell geometry for entire simulations is the same as in Table 4.1.

Figure 5.6 shows the simulations data for homogeneous prey distribution: Figure 5.6a depicts the jellyfish clearance rate as a function of diameter with linear fits of two different bell shapes determined by fineness ratio:

- Circular: $F = 7786D - 3881$ with $rRMS = 0.03$
- Oblate: $F = 2654D - 1494$ with $rRMS = 0.01$

and Figure 5.6b shows the jellyfish clearance rate as a function of the swimming number with linear fits as:

- Circular: $F = 109Sw - 605$ with $rRMS = 0.05$
- Oblate: $F = 16Sw - 281$ with $rRMS = 0.02$

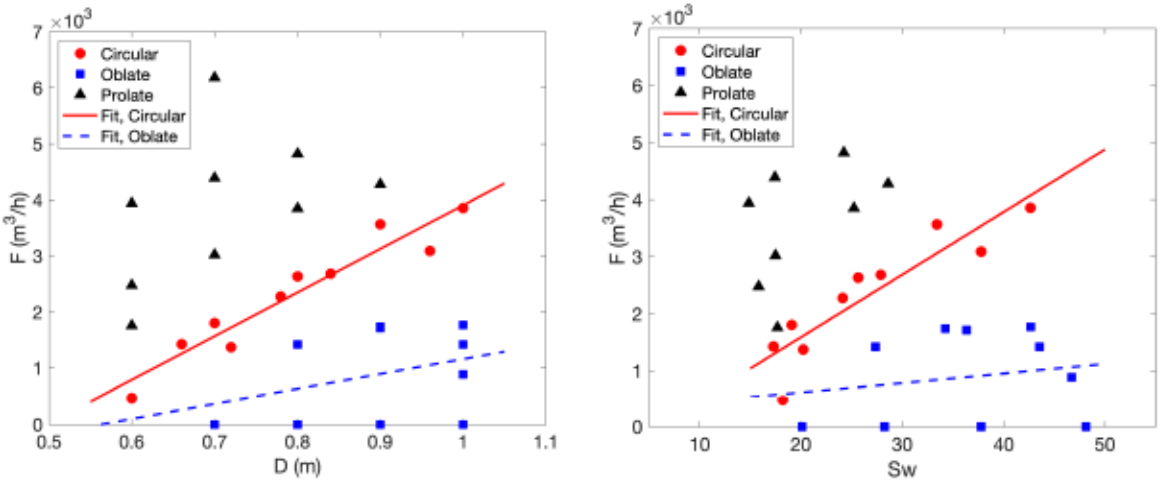


Figure 5.6: Homogeneous prey distribution: (a) F versus D, (b) F versus Sw.

Our simulations suggest that the prey clearance rate for circular jellyfish has an obvious upward trend, increasing roughly linearly with both diameter and swimming number. For oblate jellyfish, there is a much less obvious linear trend in both plots with a significantly

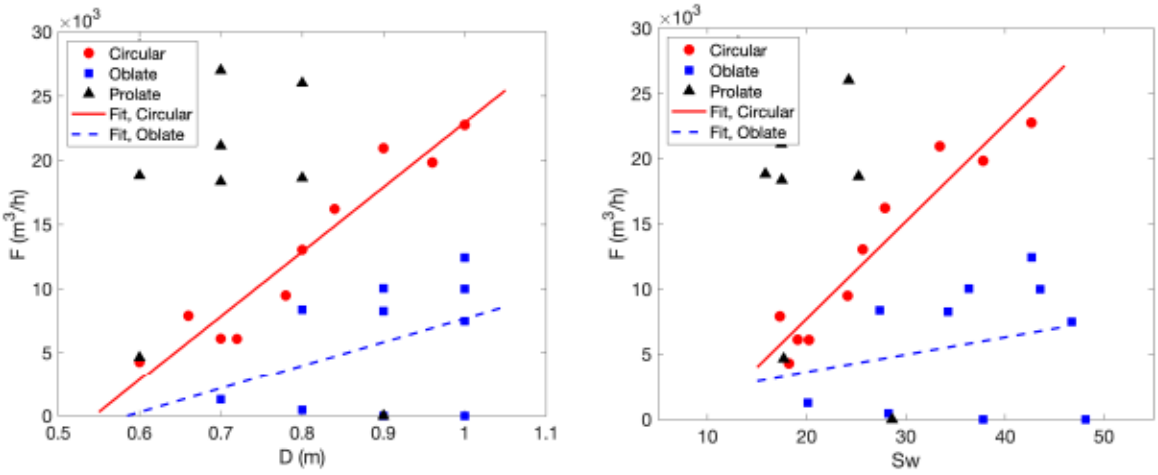
smaller slope. For prolate jellyfish, the data are much more disperse and so a linear fit seems inappropriate which emphasizes the effect of elongated bell shape on feeding efficiency. One can argue that for a prolate jellyfish swimming through a horizontal or vertical prey strip, the bell opening is possibly too small to allow prey to penetrate deeply enough to be captured compared to other bell shapes. The clearance rate of circular jellyfish increases linearly as the diameter and swimming number increase. This is an expected feeding performance given that larger jellyfish generate larger vortices. Also oblate jellyfish generate less energetic vortices that do not penetrate as far into the bell, so it is not surprising the clearance rate is much smaller.

Figure 5.7 shows the simulation data for a horizontal prey distribution. Linear fits for data shown in Figure 5.7a are:

- Circular: $F = 50318D - 27388$ with $rRMS = 0.04$
- Oblate: $F = 18493D - 10818$ with $rRMS = 0.1$

and in Figure 5.7b are:

- Circular: $F = 746Sw - 7211$ with $rRMS = 0.04$
- Oblate: $F = 133Sw + 937$ with $rRMS = 0.1$



(a) (b)

Figure 5.7: Horizontal prey distribution: (a) F versus D, (b) F versus Sw.

Figure 5.8 depicts the data for a vertical prey distribution, with linear fits in Figure 5.8a

- Circular: $F = 3891D - 1317$ with $rRMS = 0.1$
- Oblate: $F = 6207D - 4277$ with $rRMS = 0.1$

and in Figure 5.8b are:

- Circular: $F = 44Sw + 582$ with $rRMS = 0.1$
- Oblate: $F = 74Sw - 1411$ with $rRMS = 0.1$

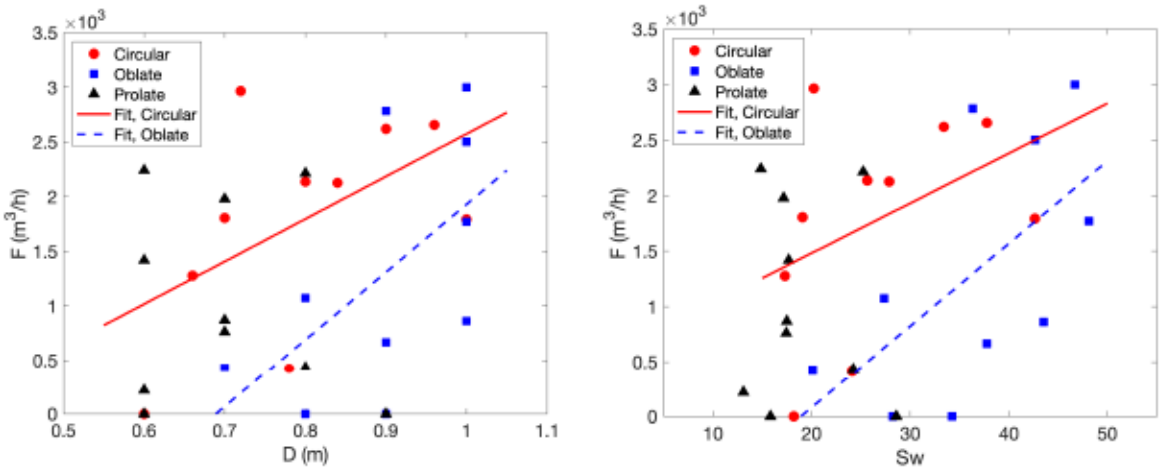


Figure 5.8: Vertical prey distribution: (a) F versus D, (b) F versus Sw.

Figures 5.7 and 5.8 also exhibit similar linear trends and so the results can be explained similarly to the homogeneous prey distribution.

5.3.3 Time Series of Prey Capture

Figure 5.9 illustrates the number of captured prey as a function of time. Simulation numbers 2, 12 and 21 from Table 4.1 are chosen as a sample of circular, oblate and prolate jellyfish types, respectively. The bell contraction frequency is $f = 1$, so that the jellyfish experiences a muscle contraction once every second, which is easy to identify in the resulting plots. Figure 5.9 sheds light on the time-dependence of feeding performance by depicting the time periods in which jellyfish do not capture any prey. As expected for homogeneously distributed prey cases feeding performance is more consistent during simulations shown in Figure 5.9a. These results also suggest that the homogeneous distribution is the best because it is simple to implement in laboratory experiments and because feeding models depend less on homogeneous prey distribution than on other distribution types. Figure 5.9b corresponds to the jellyfish swimming through a narrow horizontal strip of prey. There is a clear jump in prey captured as the strip is passed through, after which the jellyfish encounters no more prey until it approaches the periodic boundary and approaches the dispersed remains of the original strip. And at least the oblate jellyfish is able to capture a few additional prey. Time-dependent capture of prey is shown in Figure 5.9c for the vertical strip, where the jellyfish remains closer to regions containing prey through its motion. As a result, there is a rapid rise in the number of prey captured each second when the swimming muscles contract and jetting vortices act to entrain nearby prey lying to the right of the bell margin.

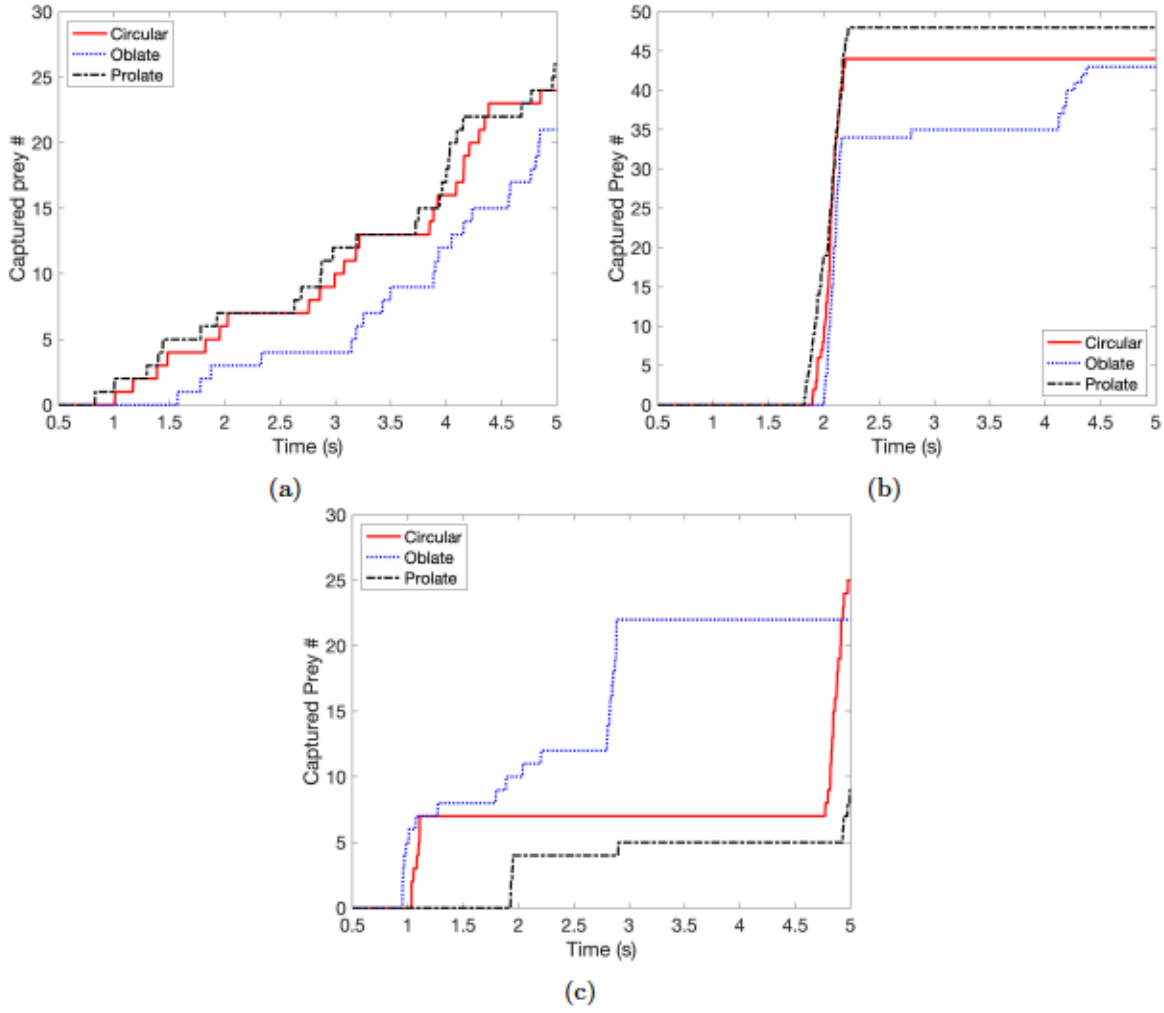


Figure 5.9: Number of prey captured as a function of time for 3 types of distribution: (a) Homogeneous prey distribution, (b) Horizontal prey distribution, and (c) Vertical prey distribution.

5.3.4 Contraction Frequency

The number of captured prey as a function of time for various bell contraction frequencies is shown in Figure 5.10, demonstrating the role of frequency on capture success. We ran a second simulation set with frequency of $f = 2$, which is twice that of the first set considering prey that are homogeneously distributed for all three types of jellyfish: circular, oblate, and prolate. Assuming no behavioral changes with size, our findings are consistent with those obtained experimentally [30], as an increase in frequency leads to increased fluid volume encountered, directly affecting feeding success.

Comparing the three jellyfish types, circular and oblate jellyfish exhibit superior prey capturing as their contraction frequency increases compared to prolate jellyfish. Results imply that the number of prey captured by circular and oblate jellyfish increase roughly

proportional to contraction frequency (doubling the frequency doubles the number of prey captured). Less frequency dependence is another characteristic of prolate jellyfish that can be explained by the its distinct bell shape. In other words, a prolate jellyfish is relatively narrow and prevents large numbers of prey from being captured. Therefore, doubling the bell contraction frequency has much less effect on prey capturing.

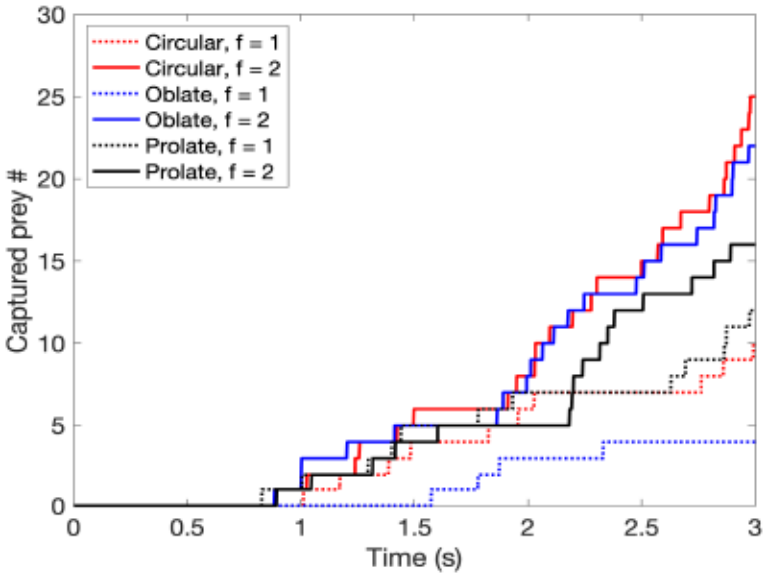


Figure 5.10: Contraction frequency role on feeding dynamics.

Note that we did not investigate the relationship between bell frequency and jellyfish size, even though it has been demonstrated experimentally that bell frequency is also affected by jellyfish size [30]; this demonstrates the difficulty of modelling the interactions between bell frequency and feeding dynamics realistically. Therefore, we must determine the bell frequency for each jellyfish size based on experimental data for more reliable results.

5.4 Conclusion

This study can be viewed as an initial step toward the ultimate goal of modelling the feeding of jetting jellyfish. The results indicate the capability of simulations in examining the feeding dynamics and efficiency. Despite the challenges of conducting lab experiments, such as their limitations like incapability to modify the shape and size of predators and the size, concentration, and distribution of prey, computational simulations allow us to circumvent all these restrictions. Possible areas for future work include more in-depth studies of different prey concentrations and different jellyfish sizes including the corresponding differences in contraction frequency.

Chapter 6

Pairwise Interactions

In this chapter, we investigate the dynamics of interactions between pairs of jellyfish which is important because they are so seldom alone but rather swim in close proximity or even in dense swarms. Many animals depend on their ability to manoeuvre which is exemplified in circumstances like feeding and avoiding predators [20]. As a result, we were motivated to study the pairwise interaction of jellyfish using numerical simulations. To evaluate close contact, we first extend our `IB2d` implementation, which initially was designed for single jellyfish as explained in Chapter 3, to a model for two jellyfish. We investigate several approaches to handle the jellyfish reactions to colliding or near contact, namely rebounding and changing swimming direction. In section 6.1, we propose an approach which creates an imbalance in forces across the body to generate repulsion forces and asymmetric turning forces that cause both jellyfish to alter their swimming trajectories and repel each other. In section 6.2, we discuss the implementation of our repulsion strategy in `IB2d`, and in section 6.3 we compiled the results of simulations. Lastly, in section 6.4 we discuss the unsuccessful strategies we investigated to make jellyfish turn.

6.1 Strategy

We employ a strategy to make jellyfish repel each other during a collision, inspired by Li et al. [22] who employ direct numerical simulations to examine the hydrodynamic interaction of model swimming organisms in the small to intermediate Reynolds number regime. This repulsive reaction can be induced by applying a force term directed normal to the nearest point of contact between the two jellyfish. We call this repulsive force term $F_{repulsive}$ and we apply this force to a subset of the bell points from each jellyfish as they approach within a user-defined "active force range" as shown in Figure 6.1. The repulsive force causes the jellyfish to gradually change their swimming trajectories during the near-contact and hence to separate from one another. The definition of the repulsive force term imposes force only on bell points lying within the "active force range". Suppose that C_m denotes the maximum

or peak repulsive force, then we define

$$F_{repulsive} = C_m \left(\frac{D - dr}{dr} \right)^2 \mathbf{e}, \tag{6.1}$$

where D is the Euclidean distance between two jellyfish Lagrangian points, dr is the force range and is set to be $dr = 2\Delta x$ and $\mathbf{e}$ is the direction of the repulsive force which is along the vector joining each pair of points. The Euclidean distance is

$$D = \sqrt{dx^2 + dy^2}, \tag{6.2}$$

and the direction of the force term is

$$\mathbf{e} = \left(\frac{dx}{D}, \frac{dy}{D} \right), \tag{6.3}$$

where dx, dy are the distance between the Lagrangian points in the x and y directions, respectively. It is worth mentioning that the characteristic force is tuned by using the results of numerical simulations to have natural-looking reactions. The choices of characteristic force also depend on grid spacing since increasing resolution causes more Lagrangian points to be affected by the repulsive force. Assuming a fixed grid size of $\Delta x = 10/400$ and comparing various numerical simulations suggest that a value of C_m in the range of $[10^4, 10^5]$ is most appropriate.

Consider the following concrete example. Suppose two jellyfish are approaching each other with half-angle $\alpha = \pi/3$, and that the pair come within the force range dr. At this moment, the repulsion force begins to apply on the Lagrangian points within the force range and Figure 6.1 depicts how the repulsion force function operates for this case. The interaction distance does not have to be the same for other IB forces since it is not a numerical parameter but is connected with jellyfish ability to sense others nearby. However, in the absence of experimental values of this sensing distance, we have found that taking a value of $2\Delta x$ gives at least visually believable results.

6.2 IB2d Implementation

To implement two jellyfish with pairwise repulsion forces, we change the primary file used to create the single jellyfish geometry to define a second jellyfish and call the corresponding file, `Two_Jellies.m`. The methodology used to do so is the same as that used to create the geometry of single jellyfish bell explained in chapter 3. To permit increasing the number of jellyfish beyond two in future, we define an integer parameter `NJ` that is set to 2 and is used to output information regarding the Lagrangian points, virtual springs and beams. Moreover, we also need to generate additional force information for repulsive forces in the

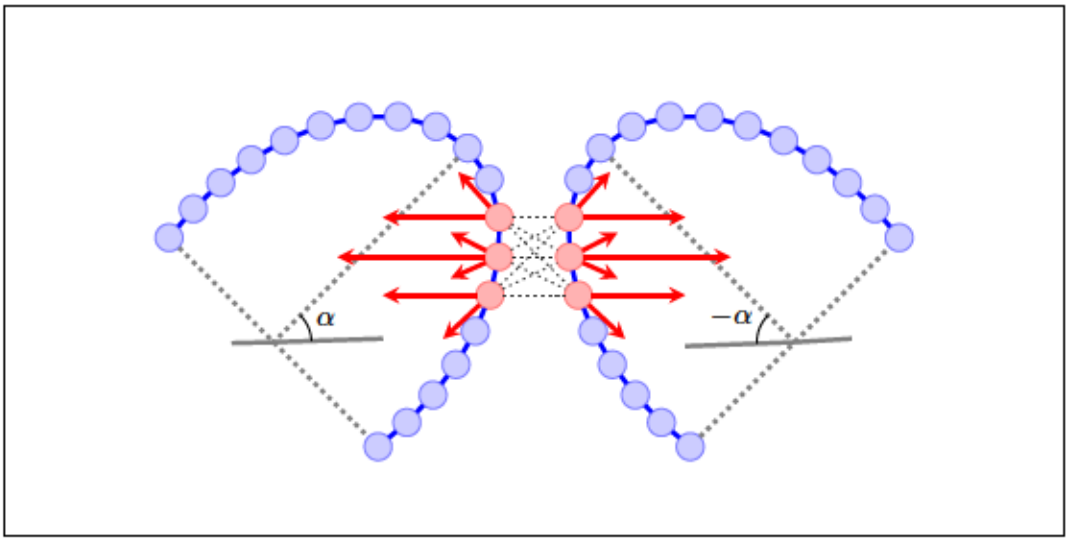


Figure 6.1: The IB points within the force range and the corresponding repulsion force arrows are shown in red. Dotted black lines show the Euclidean distance D between each pair of IB points.

file `jelly.user_force`. In addition, alterations are made to the time-stepping part of the `IBM_Driver.m` file in the black box to adapt the implementation for multiple jellyfish.

We utilize the option of creating a user-defined force model to impose the repulsion force term into `IB2d` implementation. By this option we have the flexibility to design any force term. Many parameters are automatically passed into the force model, such as the current and past positions of the immersed boundary points, current time, time-step, etc.

Like the other fibre models, the user-defined force reads data from an attached input file into an `IBM_Driver` file before passing it on to a function that calculates the deformation forces at each time step. The information from the input file is then sent to the script `give_Me_General_User_Defined_Force_Densities` in the `jelly` folder where our repulsive force term is implemented. In each time step, this force function computes the Euclidean distance between all immersed boundary points of two jellyfish and checks whether or not they place within the force range. The repulsive force applies to those immersed boundary points which are within the force range.

6.3 Simulations of Pairwise Interactions

Simulations of pairwise interactions are conducted on a fixed grid using the IBM solver `IB2d`. The reference parameters related to the initial bell geometry and fluid properties are the same as those listed in Table 3.1. The computational domain is set to $L_x = 5$, $L_y = 10$ with grid size $\Delta x = 10/400$ and time step $\Delta t = 4 \times 10^{-5}$ are fixed for all cases. The two jellyfish are the same size and shape with center points initially located at positions $(1.5, 1.5)$ and $(3.5, 1.5)$. As shown in Figures 6.2 and 6.3, the two jellyfish swim toward

each other at a swimming angle of $\alpha = \pi/3$ and they collide. The images show how jellyfish interact with one another with and without imposing the repulsion force. These plots depict vorticity with blue corresponding to negative (clock-wise) and red corresponding to positive (counter-clockwise), and represent a time sequence. In the first scenario shown in Figure 6.2, no contact forces were imposed and so this pair of jellyfish swim towards one another, collide and adhere without separating. This is not realistic behaviour since real jellyfish resist adhering by altering their trajectories. As a result this simulation is not regarded as a sensible pairwise interaction.

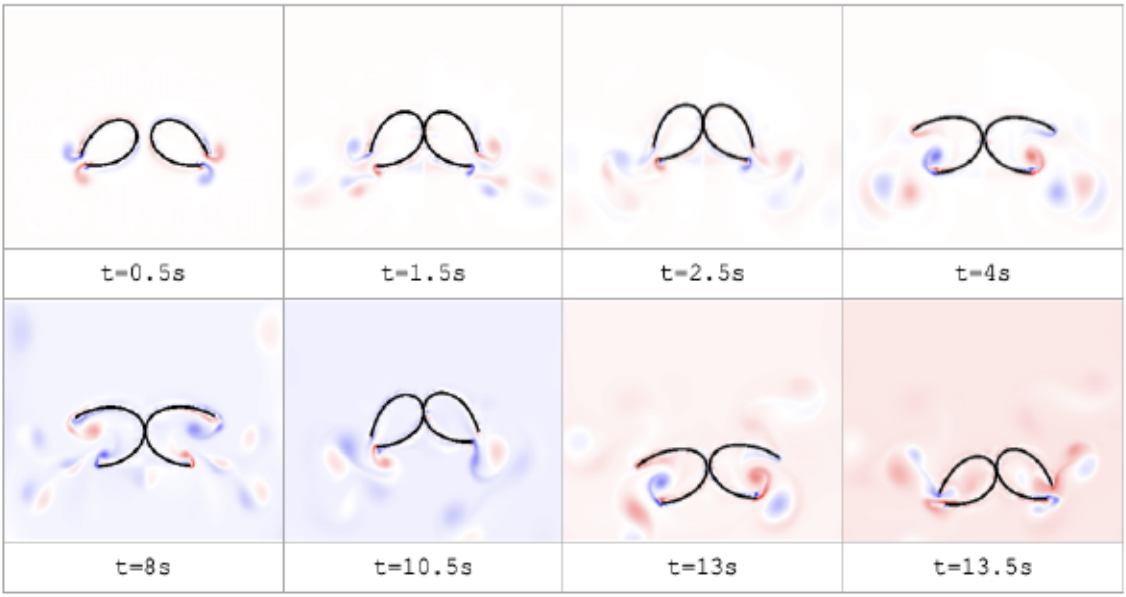


Figure 6.2: Screenshots of jellyfish interacting each other without taking into account the repulsive force term. A video is provided as supplementary information on my personal web-page.

To obtain more realistic interactions, we imposed the repulsive force as described earlier and Figure 6.3 provides screenshots from the simulation. The force term was applied to the subsets of both jellyfish bell points, causing them bounce off each other a few times then separate throughout the time sequences depicted in Figure 6.3.

It is worth mentioning that the choice of characteristic force and grid size significantly impacts interaction dynamics according to our characterization of the repulsive force term in Equation (6.1). As a result, we provide Figure 6.4 which plots the minimum separation distance versus time for three simulations using 3 different values of the characteristic force, $C_m = 5 \times 10^4, 10^5, 5 \times 10^5$. This plot suggests that for the particular simulation with mentioned initialization, the characteristic force 10^5 is the most appropriate choice since it can cause a change in swimming trajectory and repulsion while the two other choices are unsuccessful.

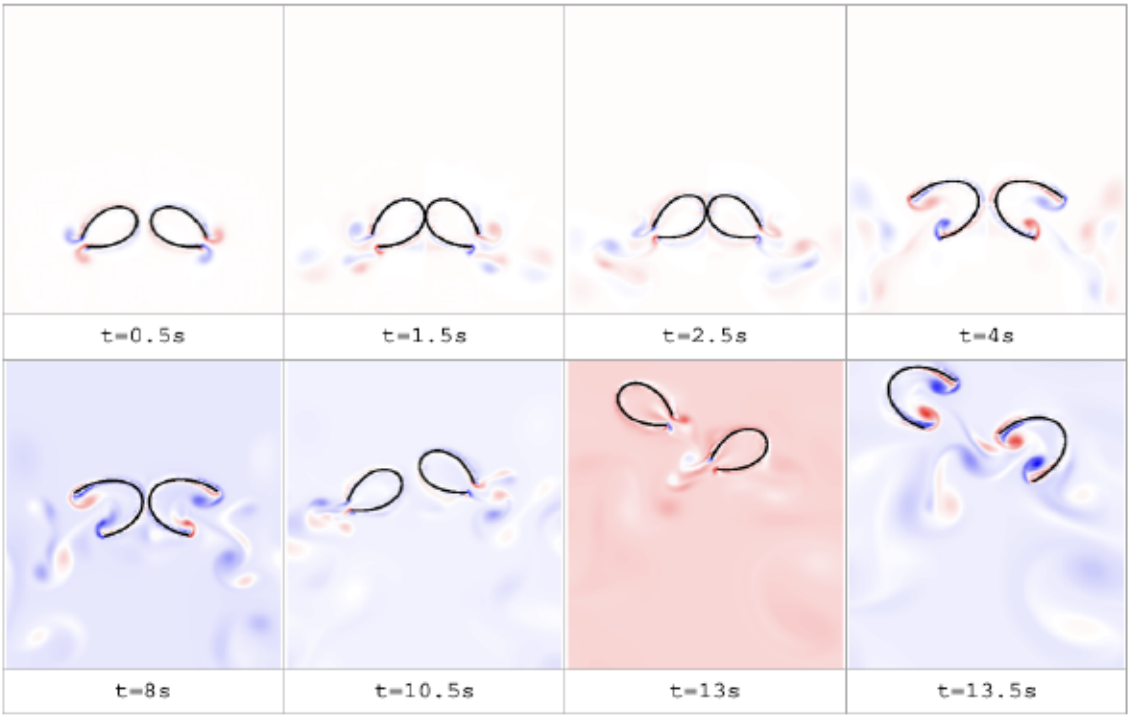


Figure 6.3: Screenshots of jellyfish interacting under the influence of a repulsive force. A video is provided as supplementary information on my personal web-page.

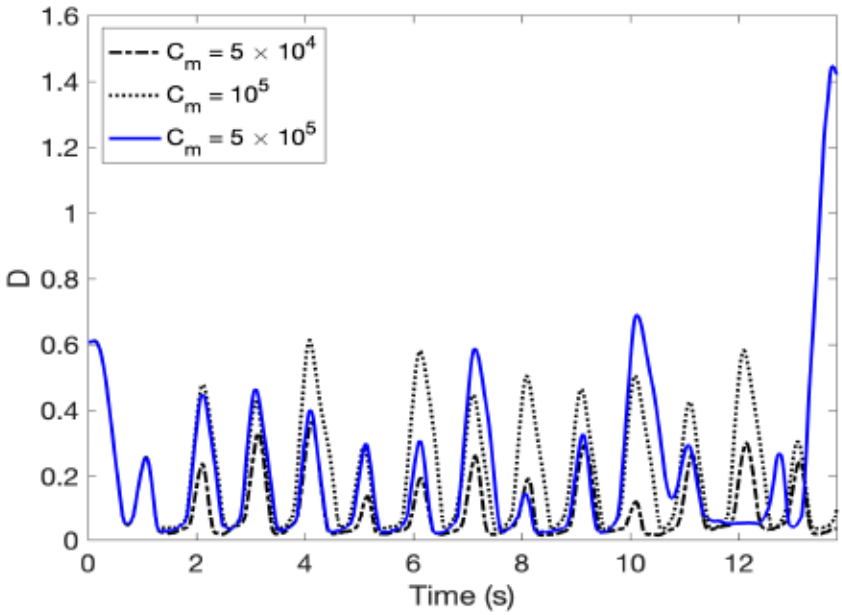


Figure 6.4: Jellyfish distance as a function of time for different characteristic force.

We consider these simulations to only be partially successful, in that the jellyfish repel each other and separate after some close interactions, however the relative swimming angle α never changes and this aspect seems unrealistic.

Another factor affecting the interaction dynamics using the repulsion force is the role of the grid size. The number of jellyfish bell points in the force range strongly influences the repulsive force field. Thus, it would be interesting to study the grid size when closely examining the hydrodynamic interaction in future work.

6.4 Attempts at Inducing Turning Dynamics

In current section, we describe three attempts to implement forces that induce an additional turning reaction by introducing an asymmetry in IB forcing:

- **Asymmetric muscle stiffness:** In our 2D simulations, we implement a function that stiffens one side of the bell margin to attempt to induce the jellyfish to turn. We implemented a method that is roughly similar to that discussed by Pallasdies et al. [24], who discussed the motion of the bell margin used by some oblate species in which the outer side of the bell (i.e., facing outside the turn) is stiffer than the inside and paddling action drives motion instead of jetting contraction. They observed that turning motion in jellyfish occurs by varying the stiffness of the bell in proportion to one another and impacting the strength of the vorticity created by the bell, which explains the difference in rotational speed. Figure 6.5 provides an illustration of how we aimed to mimic this behaviour with the points corresponding to stiffened beam forces on the bell margin shaded in grey. The reason that the current attempt failed is probably because it calls for adding new paddling motions on one side, which are not typically seen in jet-like swimmers.

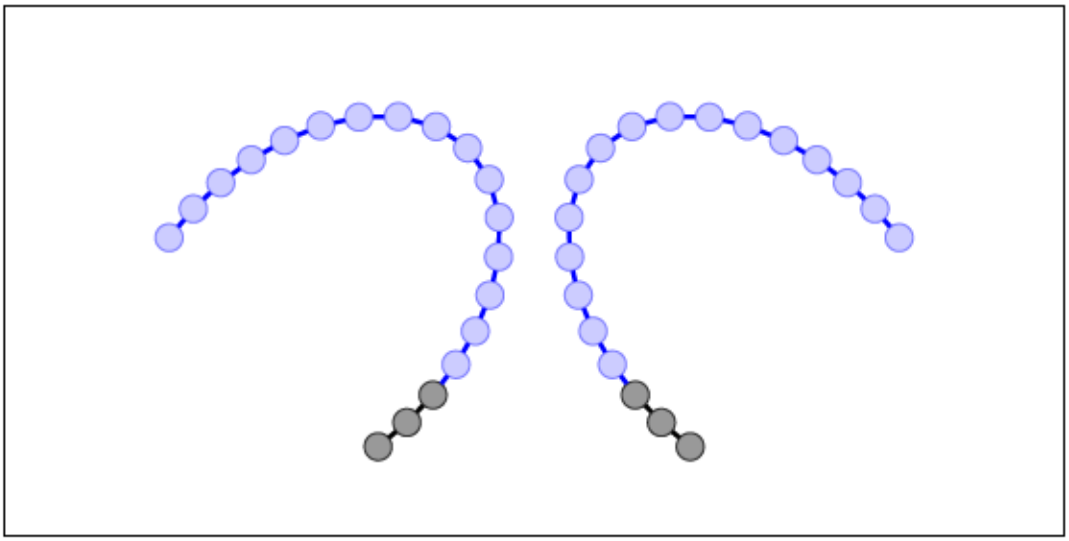


Figure 6.5: Varying bell margin stiffness to impact the strength of vortices and induce turning. The “normal” beam points are colored in blue, while the stiffened points are coloured gray.

- **Apply opposing forces at the central IB points:** In addition to the subsets of bell points within the force range which are already affected by repulsion force introduced earlier, a force is applied to a subset of points near the front edge or "nose" of each jellyfish, but pointing in opposite directions to cause a relative rotation. Figure 6.6 illustrates the normal repulsion forces in read that operate at the points of nearest approach, whereas the turning forces are imposed at the blue points and depicted with arrows directed in opposite directions to induce a turning motion. Applying this strategy strengthens the repulsion forces but does not cause any noticeable turning or relative rotation between the two jellyfish.

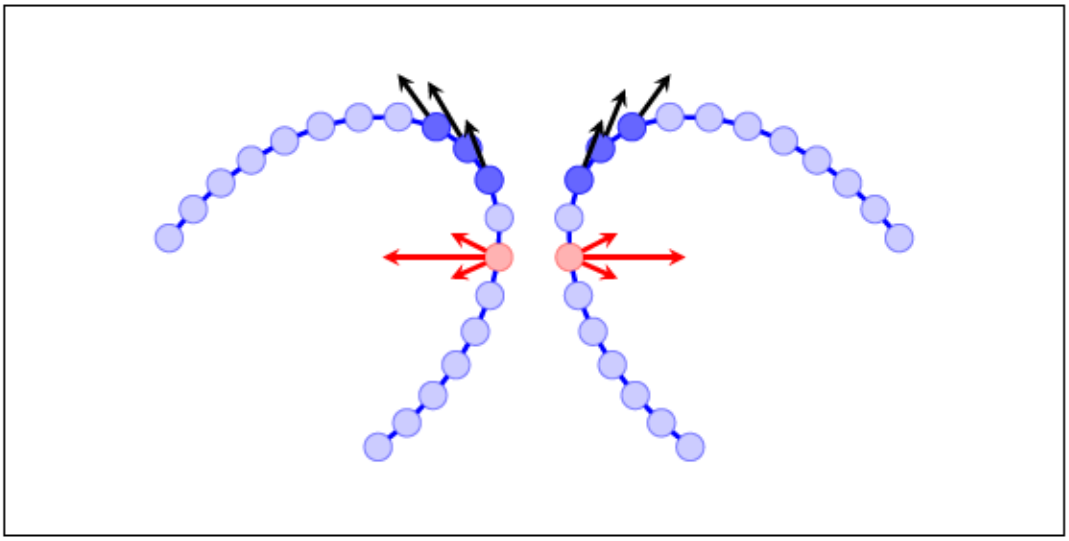


Figure 6.6: Applying an additional force to the central points of the bell (three dark blue central points) as they place within the force range, and any other bell point within the force range undergo a repulsion force shown in red.

- **Introduce asymmetric muscle points:** In a third attempt to simulate turning, we defined additional asymmetric muscle contractions, which begin contracting alongside the original muscle points as soon as a pair of jellyfish enters the force range. This method is depicted in Figure 6.8. Note that the extra asymmetric pairs of muscle points are defined according to this particular orientation of jellies. Consequently, different collision orientations may necessitate distinct types of asymmetric muscle points. This strategy fails to cause turning motion and results in an asymmetric, tangential skew in the bell shape that seems non-realistic since we have never observed such motions in videos of actual jellyfish. Figure 6.7 illustrates the mentioned tangential skew, one aspect of this seems promising in that they appear to be turning relative to each other; however, the turn is in the wrong direction (that is, turning inwards toward each other). But, if we reverse the orientation of those asymmetric forces, the behaviour of

jellyfish through the interaction when they are close to each other is even less realistic than the one we have proposed.

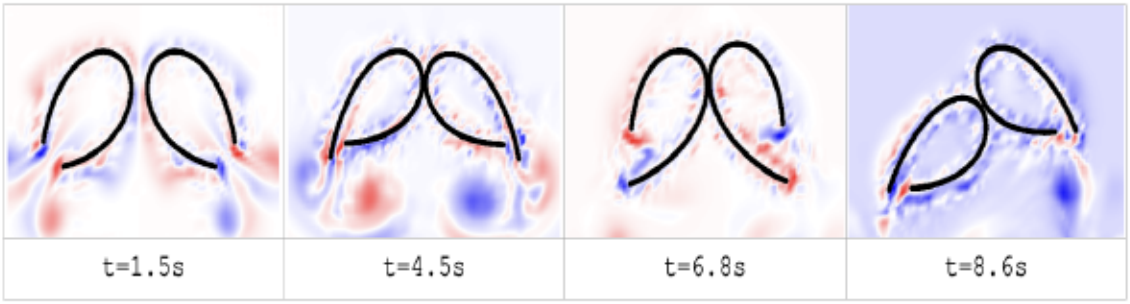


Figure 6.7: Screenshots of jellyfish interacting while considering asymmetric muscle points.

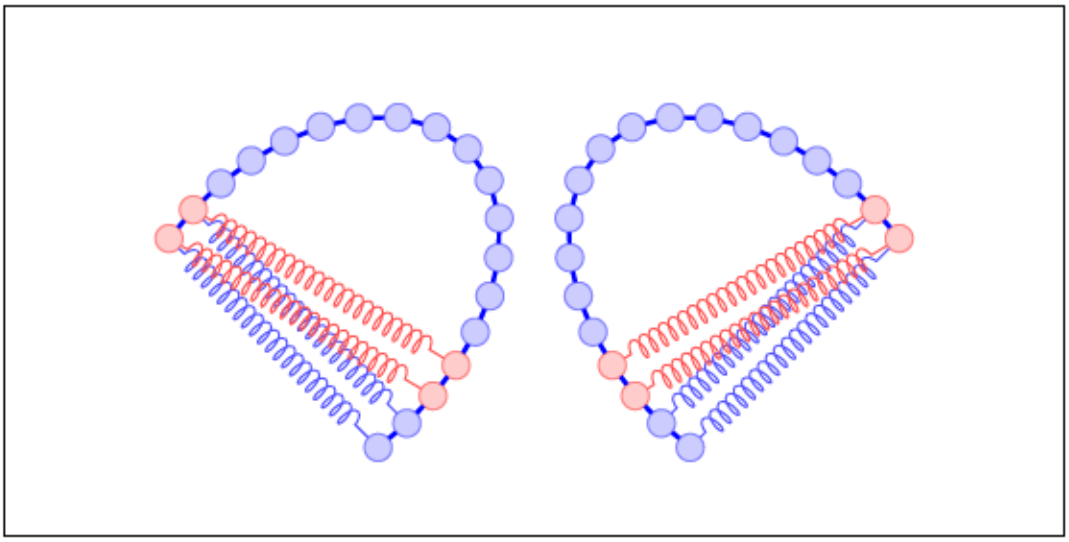


Figure 6.8: An extra asymmetric pairs of muscles points start contracting within force range.

In summary, none of these three attempts described above were able to produce any measurable turning. Turning appears to be difficult to induce in two dimensions and so it may be that we must introduce some form of artificial torque to the entire bell. Because torque forces are not one of the options in `IB2d`, we have not yet attempted to implement this.

6.5 Conclusion

The long-term purpose of our effort in this chapter is to simulate jellyfish swarms in two dimensions. We examined two different aspects of interactions: turning, which hasn't yet worked perhaps because turning requires a fully three-dimensional response, and repulsion, which produced acceptable results and caused two jellyfish swimming symmetrically towards each other to separate after a couple of jetting cycles.

Chapter 7

Conclusion

This thesis aims to develop a 2D model for jet-like swimming in jellyfish and use numerical simulations with the immersed boundary method to study questions related to scaling of swimming dynamics, feeding efficiency and pairwise interactions. We validated our model and compared our numerical simulations with other results from the literature related to scaling locomotion of aquatic animals, jellyfish feeding efficiency and dynamics, and jellyfish interaction. Even though certain aspects of jellyfish swimming, such as turning motion, can be more realistically modelled in three dimensions, our two-dimensional model sheds light on certain aspects of swimming dynamics.

First, we showed that dimensionless swimming numbers derived for undulatory swimmers can be applied to jet-like swimming in jellyfish. The results are consistent with a wide variety of other aquatic animals. Some small variations in the power law scaling relationship can likely be attributed to various gaits related to the body.

Secondly, we used passive tracer particles to represent passive prey such as algae or plankton. We generated swimming visualizations that demonstrate how the vortices generated by periodic bell contractions are responsible for entraining food particles within the bell for consumption. Furthermore, we are able to quantify feeding efficiency in our various simulations using a prey clearance rate proposed by marine biologists and compare it with their experimental results.

Thirdly, we studied pairwise interactions between colliding jellyfish intending to develop IB forces that generate realistic-looking repulsion and turning behaviour. Our repulsion force kinetics seem to mimic real jellyfish, but further work is required to obtain forces that yield realistic turning motions as we assumed only identical forces for both jellyfish while in reality the two jellyfish would not react in exactly the same way. Future studies may include better looking repulsion behaviour by introducing some random component to the force amplitude that captures the innate variation between individual organisms that cannot be predicted in advance, moreover, they may include more approaches to the turning problem, and once it is resolved, our immersed boundary code may be applied to simulating suspensions of several jetting jellyfish.

Bibliography

[1] José L. Acuña, Á. López-Urrutia, and S. Colin. Faking giants: The evolution of high prey clearance rates in jellyfishes. *Science*, 333(6049):1627–1629, 2011.

[2] S. Alben, L. A. Miller, and J. Peng. Efficient kinematics for jet-propelled swimming. *Journal of Fluid Mechanics*, 733:100–133, 2013.

[3] T. Baldwin and N. A. Battista. Hopscotching jellyfish: combining different duty cycle kinematics can lead to enhanced swimming performance. *Bioinspiration and Biomimetics*, 16(6):066021, 2021.

[4] Nicholas A. Battista, Christopher P. Strickland, Aaron Barrett, and Laura A. Miller. IB2d reloaded: A more powerful Python and MATLAB implementation of the immersed boundary method. *Mathematical Methods in the Applied Sciences*, 41:8455–8480, 2017.

[5] Nicholas A. Battista, **W**. Christopher Strickland, and Laura A. Miller. IB2d: a Python and MATLAB implementation of the immersed boundary method. *Bioinspiration and Biomimetics*, 12(3):036003–036003, 2017.

[6] Nicholas Bezio, John H. Costello, Elijah Perry, and Sean P. Colin. Effects of capture surface morphology on feeding success of *Scyphomedusae*: a comparative study. *Marine Ecology Progress Series*, 596:83–93, 2018.

[7] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, K. Bonnell, M. Miller, G. **W**eber, C. Harrison, D. Pugmire, T. Fogal, C. Garth, A. Sanderson, E. Wes Bethel, M. Durant, D. Camps, Jean M. Favre, Oliver R. Bel, M. Wheeler P. Navratil, and P. Selby. *VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data*. https://visit-dav.github.io/visit-website/index.html, 2011.

[8] Sean P. Colin and John H. Costello. Morphology, swimming performance and propulsive mode of six co-occurring hydromedusae. *Journal of Experimental Biology*, 205(3):427–437, 2002.

[9] John H. Costello, Sean P. Colin, and John O. Dabiri. Medusan morphospace: phylogenetic constraints, biomechanical solutions, and ecological consequences. *Invertebrate Biology*, 127:265–290, 2008.

[10] John H. Costello, Sean P. Colin, John O. Dabiri, Brad J. Gemmell, Kelsey N. Lucas, and Kelly R. Sutherland. The hydrodynamics of jellyfish swimming. *Annual Review of Marine Science*, 13(1):375–396, 2021.

[11] Thomas L. Daniel. Cost of locomotion: Unsteady medusan swimming. *Journal of Experimental Biology*, 119:149–164, 1985.

[12] M. Dawoodian and A. Sau. Kinetics and prey capture by a paddling jellyfish: three-dimensional simulation and Lagrangian coherent structure analysis. *Journal of Fluid Mechanics*, 912:A41, 2021.

[13] Matevž Dular, Tom Bajcar, and Brane Širok. Numerical investigation of flow in the vicinity of a swimming jellyfish. *Engineering Applications of Computational Fluid Mechanics*, 3(2):258–270, 2009.

[14] Mattia Gazzola, Médéric Argentina, and L. Mahadevan. Scaling macroscopic aquatic locomotion. *Nature Physics*, 10(10):758–761, 2014.

[15] Brad J. Gemmell, Daniel R. Troolin, John H. Costello, Sean P. Colin, and Richard A. Satterlie. Control of vortex rings for manoeuvrability. *Journal of The Royal Society Interface*, 12(108):20150389, 2015.

[16] Robert A. Granger. *Fluid Mechanics*. Dover Publication, Inc, New York, 1995.

[17] Gregory Herschlag and Laura Miller. Reynolds number limits for jet propulsion: A numerical study of simplified jellyfish. *Journal of Theoretical Biology*, 285(1):84–95, 2011.

[18] A. Hoover, B. Griffith, and L. Miller. Quantifying performance in the medusan mechanospace with an actively swimming three-dimensional jellyfish model. *Journal of Fluid Mechanics*, 813:1112–1155, 2017.

[19] A. Hoover and L. Miller. A numerical study of the benefits of driving jellyfish bells at their natural frequency. *Journal of Theoretical Biology*, 374:13–25, 2015.

[20] Howard C. Howland. Optimal strategies for predator avoidance: the relative importance of speed and manoeuvrability. *Journal of Theoretical Biology*, 47(2):333–350, 1974.

[21] Ming-Chih Lai and Charles S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000.

[22] Gaojin Li, Anca Ostace, and Arezoo M. Ardekani. Hydrodynamic interaction of swimming organisms in an inertial regime. *Physical Review E*, 94:053104, 2016.

[23] MATLAB. *Release R2021a*. The MathWorks Inc., Natick, Massachusetts, 2021.

[24] F. Pallasdies, S. Goedeke, W. Braun, and Raoul-Martin Memmesheimer. From single neurons to behavior in the jellyfish *Aurelia aurita*. *eLife*, 8:e50084, 2019.

[25] Charles S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[26] Charles S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[27] Steve Piacsek and Gareth P. Williams. Conservation properties of convection difference schemes. *Journal of Computational Physics*, 6:392–405, 1970.

[28] G. Van Rossum. *Python version 3.5*. https://www.python.org, 2015.

[29] M. Sahin and K. Mohseni. An arbitrary Lagrangian–Eulerian formulation for the numerical simulation of flow patterns generated by the hydromedusa *Aequorea victoria*. *Journal of Computational Physics*, 228(12):4588–4605, 2009.

[30] J. Titelman and L. J. Hansson. Feeding rates of the jellyfish *Aurelia aurita* on fish larvae. *Marine Biology*, 149(2):297–306, 2006.

Appendix A

Supplementary Videos

Creator: Mahdi Salehzadeh

Description: The videos are complementary to this thesis. They explore the idea of jellyfish feeding performance considering different types of prey distribution pattern.

Filenames:

- Homogeneous.mp4
- Horizontal.mp4
- Vertical.mp4

Description: The videos are complementary to this thesis. They explore the idea of jellyfish pairwise interaction with and without the repulsion force.

Filenames:

- NoForce.mp4
- WithForce.mp4

Description: The video is complementary to this thesis. It explores the idea of introducing asymmetric muscles points.

Filename:

- AsymMuscles.mp4

Feeding simulation and interaction simulation videos can be found in the link1 and link2, respectively.