

## Bachelor thesis

Cairui Duan and Tobias Hartung Bitsch Holm

## Low-Reynolds number swimming and sensing using reinforcement learning

Date: June 2023

Advisor: Julius Bier Kirkegaard

## Abstract

The swimming kinematics for the general, non-axisymmetric case is derived and only three modes are found to be responsible for translation. Then, using reinforcement learning ("RL"), we model and solve the following cases: First, using the analytical flow field, the trajectory of a squirmer catching a point prey is investigated, and then, using the Boundary Element Method ("BEM") the optimal squirmer modes for sensing a finite-sized prey are found. The numerical flow found using the BEM is validated against the analytical flow, and the relative error between the two fields is found as a function of the BEM regularization offset,  $\epsilon$ , and the number of surface points,  $N$ . Taking computation time into account, we choose  $N = 700$  surface points and a regularization offset of  $\epsilon = 0.05$ . By using RL we show that when the squirmer's power consumption is constant its optimal catching strategy is to move in a straight line to the target regardless of their initial angle. However, when the power is not limited the squirmer moves in either a L-shape or a zigzag motion for non-diagonal initial angles to maximize both translational modes simultaneously. By varying target radius, sensor noise, squirmer-target centre distance and angle we show that in the stable mode regions, the translational modes are always efficient at sensing the target, meaning optimal sensing is optimal swimming.

# Contents

<b>List of symbols</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
<b>2 The Squirmer Model</b>	<b>3</b>
2.1 The pumping problem . . . . .	3
2.2 The translating squirmer and its swimming velocity . . . . .	4
2.2.1 Calculating the traction vector . . . . .	6
2.2.2 Calculating the frictional force and torque . . . . .	8
<b>3 Numerical Methods</b>	<b>10</b>
3.1 Boundary Element Method (BEM) . . . . .	10
3.1.1 The Oseen tensor . . . . .	10
3.2 Validation and choice of parameters values . . . . .	12
3.3 Constant power . . . . .	14
<b>4 Reinforcement Learning</b>	<b>15</b>
4.1 Essential reinforcement learning and the squirmer model . . . . .	15
4.1.1 Proximal Policy Optimization . . . . .	15
4.2 Catching a point target with perfect sensing . . . . .	16
4.3 Imperfectly sensing a finite target . . . . .	17
<b>5 Results</b>	<b>18</b>
5.1 Perfect sensing and constant power . . . . .	18
5.2 Perfect sensing and no power limit . . . . .	19
5.3 Imperfectly sensing a finite target . . . . .	19
<b>6 Discussion</b>	<b>22</b>
6.1 Idealizations . . . . .	22
6.1.1 Constant force and Oseen tensor . . . . .	23
6.1.2 One object power equation . . . . .	23
6.2 Further improvements . . . . .	23
6.2.1 Catching a finite target with imperfect sensing . . . . .	23
6.3 Comparing our results to the Zhu 2022 article . . . . .	24
<b>7 Conclusion</b>	<b>24</b>
<b>Acknowledgement</b>	<b>24</b>
<b>References</b>	<b>24</b>
<b>A Appendix</b>	<b>26</b>
A.1 Derivation of derivatives of spherical harmonics . . . . .	26
A.2 Derivation of cross product of spherical harmonics . . . . .	27
A.3 Derivation of surface integral theorems . . . . .	27
A.3.1 Evaluating $\iint_S \Gamma_{-n-1} dS$ . . . . .	27
A.3.2 Evaluating $\iint_S a\hat{\mathbf{r}}\Gamma_{-n-1} dS$ . . . . .	28
A.3.3 Evaluating $\iint_S \nabla \Gamma_{-n-1} dS$ . . . . .	29
A.3.4 Evaluating $\iint_S \nabla \times (\mathbf{r}\Gamma_{-n-1}) dS$ . . . . .	30

## List of symbols

Symbol	Description
$a$	Squirmer radius
$\mathbf{u}$	Velocity field in squirmer's reference frame
$\mathbf{v}$	Velocity field in lab frame
$p$	Pressure field
$p_{-n-1}, \Phi_{-n-1}, \chi_{-n-1}$	Spherical harmonics
$P_n^m(\cos \theta)$	Associated Legendre polynomials. Its argument is $(\cos \theta)$
$P$	Power consumption
$\mu$	Dynamic viscosity
$B_{mn}, \tilde{B}_{mn}, C_{mn}, \tilde{C}_{mn}$	Modes. A number in $[-1, 1]$
$\mathbf{U}$	Translational velocity
$\boldsymbol{\Omega}$	Angular velocity
$\mathbf{F}_{\text{squirm}}, \mathbf{T}_{\text{squirm}}$	Frictional force and torque on squirmer boundary due to its squirming actions
$\mathbf{F}_{\text{swim}}, \mathbf{T}_{\text{swim}}$	Drag force and torque due to swimming and rotating
$\boldsymbol{\Pi}_r$	Stress vector: Stress tensor dotted with radial vector
$\mathbf{S}$	Oseen tensor
$\Delta A$	Boundary element area
$N$	Number of surface points used to discretize the squirmer sphere
$\epsilon$	BEM regularization offset
$d, d_i, d_\epsilon$	Total distance-, $i$ 'th direction distance-, total distance including regularization offset between two boundary elements
${}^l \mathbf{a}^k$	Coordinate of the $k$ 'th boundary element relative to the center of sphere $l$ .
$\mathcal{A}, a_t$	Set of actions, action at time $t$
$\mathcal{R}, R_t$	Set of rewards, reward at time $t$
$\mathcal{S}, s_t$	Set of agent states, agent state at time $t$
$\pi_\Theta$	Policy with weights $\Theta$

$\hat{\mathcal{A}}_t$	Estimator of advantage function
$r_0, D_0$	Center to center-, surface to surface initial distance
$\hat{\mathbf{O}}$	Normalized average force field difference
$\sigma_{\text{noise}}$	Strength of sensor noise
$a_{\text{target}}$	Target radius
$\theta_0, \phi_0,$	Target's initial polar angle, and azimuthal angle

## 1 Introduction

The squirmer is a well-established model and still an active field, efficiently modelling microorganisms' behaviour. Reinforcement learning ("RL"), on the other hand, has been going in and out of popularity for half a century [1], and only in the last couple of decades has RL been gaining significant academic momentum. Its applications have begun extending beyond computer science, such as optimizing active particle navigation in flow (Colabrese et al. [2]) or combining the squirmer model and RL to investigate time and efficiency-optimal predation (Zhu et al. [3]).

In this thesis, we first introduce and derive the swimming kinematics of a squirmer. After introducing the mathematical model, we implement an RL algorithm to investigate the trajectory of a squirmer with perfect sensing chasing after a point-sized prey. The RL agent controls the squirmer, which can activate different modes to translate forward and laterally and generate different fluid flows. This optimal chasing trajectory is then compared to the results in Zhu et al. [3]. Next, a Boundary Element Method ("BEM") is developed to model the fluid field around a squirmer in the presence of a finite-sized passive particle. We validate the BEM with the analytical solution when only a squirmer is present, and determine the parameters needed for acceptable accuracy and runtime. Lastly, we adopt RL to explore the optimal modes needed for the squirmer to sense the flow disturbances caused by the finite-sized passive particle for different sensor noise, particle size, position and angle.

## 2 The Squirmer Model

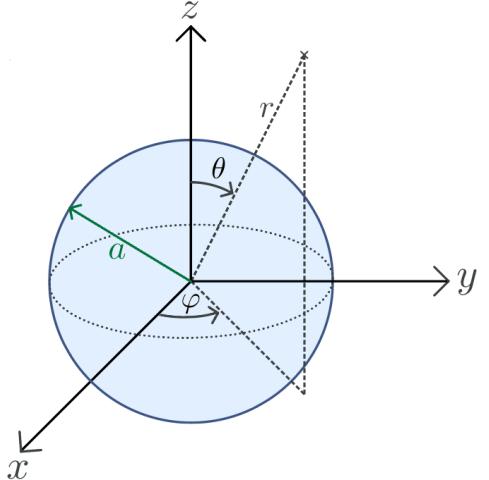
The squirmer is a model for describing self-propelled microswimmers in Stokes flow [4]. We will here introduce this model and derive which modes are responsible for the swimming and rotational velocity.

### 2.1 The pumping problem

We consider a spherical squirmer in an incompressible fluid at 0 Reynolds number. We use spherical coordinates with  $(\hat{\mathbf{r}}, \hat{\theta}, \hat{\phi})$  as the basis vectors (Figure 1). Lamb [5, 6] derived a general solution for a spherical squirmer in Stokes flow given that the flow is axisymmetric. Recently, Pak and Lauga [7] derived the flow field for nonaxisymmetric squirming motion. We will briefly go through their results here. For a squirmer fixed in space by an external force with the requirement that the velocities are purely tangential on the surface of the squirmer and zero at infinity, Pak and Lauga [7] found the flow and pressure field to be:

$$\mathbf{u}(r, \theta, \phi) = \sum_{n=1}^{\infty} \left[ -\frac{(n-2)r^2 \nabla p_{-n-1}}{2\mu n(2n-1)} + \frac{(n+1)\mathbf{r} p_{-n-1}}{\mu n(2n-1)} + \nabla \Phi_{-n-1} + \nabla \times (\mathbf{r} \chi_{-n-1}) \right] \quad (1)$$

$$p(r, \theta, \phi) = \sum_{n=1}^{\infty} p_{-n-1} \quad (2)$$



**Figure 1:** Spherical coordinates with origin at the centre of a squirmer with radius  $a$ .

$\mu$  is the dynamic viscosity, and  $p_{-n-1}, \Phi_{-n-1}, \chi_{-n-1}$  are spherical harmonics given by:

$$p_{-n-1} = r^{-n-1} \sum_{m=0}^n P_n^m(\cos \theta) [A_{mn} \cos m\phi + \tilde{A}_{mn} \sin m\phi] \quad (3)$$

$$\Phi_{-n-1} = r^{-n-1} \sum_{m=0}^n P_n^m(\cos \theta) [B_{mn} \cos m\phi + \tilde{B}_{mn} \sin m\phi] \quad (4)$$

$$\chi_{-n-1} = r^{-n-1} \sum_{m=0}^n P_n^m(\cos \theta) [C_{mn} \cos m\phi + \tilde{C}_{mn} \sin m\phi] \quad (5)$$

Here  $P_n^m(\cos \theta)$  are the Associated Legendre polynomials and its argument is  $\cos \theta$ . Inserting (3)-(5) into (1) and requiring that the velocity is purely tangential on the surface of squirmer  $u_r(r = a, \theta, \phi) = 0$ , gives the constraints:

$$A_{mn} = \frac{2(2n-1)\mu}{a^2} B_{mn}, \quad \tilde{A}_{mn} = \frac{2(2n-1)\mu}{a^2} \tilde{B}_{mn} \quad (6)$$

The flow field in the explicit form is:

$$u_r = \sum_{n=1}^{\infty} \sum_{m=0}^n \frac{(n+1)P_n^m}{r^{n+2}} \left( \frac{r^2}{a^2} - 1 \right) [B_{mn} \cos m\phi + \tilde{B}_{mn} \sin m\phi] \quad (7)$$

$$u_{\theta} = \sum_{n=1}^{\infty} \sum_{m=0}^n \left[ \sin \theta P_n^{m'} \left( \frac{n-2}{na^2 r^n} - \frac{1}{r^{n+2}} \right) (B_{mn} \cos m\phi + \tilde{B}_{mn} \sin m\phi) \right. \\ \left. + \frac{m P_n^m}{r^{n+1} \sin \theta} (\tilde{C}_{mn} \cos m\phi - C_{mn} \sin m\phi) \right] \quad (8)$$

$$u_{\phi} = \sum_{n=1}^{\infty} \sum_{m=0}^n \left[ \frac{\sin \theta P_n^{m'}}{r^{n+1}} (C_{mn} \cos m\phi + \tilde{C}_{mn} \sin m\phi) \right. \\ \left. - \frac{m P_n^m}{\sin \theta} \left( \frac{n-2}{na^2 r^n} - \frac{1}{r^{n+2}} \right) (\tilde{B}_{mn} \cos m\phi - B_{mn} \sin m\phi) \right] \quad (9)$$

$P_n^{m'}(\cos \theta)$ , with a prime, are the Associated Legendre polynomials differentiated with respect to  $\cos(\theta)$ . The coefficients  $B_{mn}, \tilde{B}_{mn}, C_{mn}, \tilde{C}_{mn}$  are arbitrary constants that determine how the squirmer moves and the subsequent flow field. We will refer to these constants as *modes*. Equations (7), (8), (9) are often referred to as the solution to the pumping problem. Lastly, Pak and Lauga found the power to be [7]:

$$P = \frac{64\pi\mu}{3a^5} (B_{01}^2 + B_{11}^2 + \tilde{B}_{11}^2) + \sum_{n=2}^{\infty} \frac{4n(n+1)\pi\mu}{a^{2n+1}} \left( \frac{4}{n^2 a^2} B_{0n}^2 + \frac{n+2}{2n+1} C_{0n}^2 \right) \\ + \sum_{n=2}^{\infty} \sum_{m=1}^n \frac{2n(n+1)(n+m)!\pi\mu}{a^{2n+1}(n-m)!} \left[ \frac{4}{n^2 a} (B_{mn}^2 + \tilde{B}_{mn}^2) + \frac{n+2}{2n+1} (C_{mn}^2 + \tilde{C}_{mn}^2) \right] \quad (10)$$

Which later will be used to limit the modes.

## 2.2 The translating squirmer and its swimming velocity

We now allow the squirmer to move freely and wish to find the squirmer's swimming speed  $\mathbf{U}$  and angular velocity  $\boldsymbol{\Omega}$ , as a function of the modes:  $B_{mn}, \tilde{B}_{mn}, C_{mn}, \tilde{C}_{mn}$ . To our knowledge, no complete derivation of this has been published for the general, non-axisymmetric case. Most articles refer to Lamb's results [6], where the flow is axisymmetric. Pak and Lauga have simply stated, that the solution for the general, non-axisymmetric case is the same as the axisymmetric

case, but more involved [7]. We will here derive which modes are responsible for the squirmer's velocity  $\mathbf{U}$  and angular velocity  $\boldsymbol{\Omega}$ .

The overall flow field of a swimming squirmer,  $\mathbf{v}$ , is the solution to the pumping problem (7)-(9), superimposed with the flow field,  $\mathbf{v}_T$ , of a sphere translating with the velocity  $\mathbf{U} = (U_x, U_y, U_z)$  and the flow field,  $\mathbf{v}_R$ , due to a sphere rotating with angular velocity  $\boldsymbol{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$  [5, 8].  $\mathbf{v}_T$  is simply the stokes flow past a sphere, and  $\mathbf{v}_R$  is a rotlet. Here  $U_x, U_y, U_z$  is the velocity of the squirmer in the  $\mathbf{x}, \mathbf{y}, \mathbf{z}$  - direction, respectively, and  $\Omega_x, \Omega_y, \Omega_z$  denotes angular velocity around the  $\mathbf{x}, \mathbf{y}, \mathbf{z}$  axis, respectively. The overall field, when  $\mathbf{v}_T$  and  $\mathbf{v}_R$  have been superimposed with  $\mathbf{u}$  from (7)-(9), is:

$$v_r = \left( \frac{3a}{2r} - \frac{a^3}{2r^3} \right) \cdot \{U_x \sin \theta \cos \phi + U_y \sin \theta \sin \phi + U_z \cos \theta\} \quad (11)$$

$$+ \sum_{n=1}^{\infty} \sum_{m=0}^n \frac{(n+1)P_n^m}{r^{n+2}} \left( \frac{r^2}{a^2} - 1 \right) [B_{mn} \cos m\phi + \tilde{B}_{mn} \sin \phi]$$

$$v_\theta = \left( \frac{3a}{4r} + \frac{a^3}{4r^3} \right) \cdot \{U_x \cos \theta \cos \phi + U_y \cos \theta \sin \phi - U_z \sin \theta\} + \frac{a^3}{r^2} (-\Omega_x \sin \phi + \Omega_y \cos \phi) \quad (12)$$

$$+ \sum_{n=1}^{\infty} \sum_{m=0}^n \left[ \sin \theta P_n^{m'} \left( \frac{n-2}{na^2 r^n} - \frac{1}{r^{n+2}} \right) (B_{mn} \cos m\phi + \tilde{B}_{mn} \sin \phi) + \frac{m P_n^m}{r^{n+1} \sin \theta} (\tilde{C}_{mn} \cos m\phi - C_{mn} \sin m\phi) \right]$$

$$v_\phi = \left( \frac{3a}{4r} + \frac{a^3}{4r^3} \right) \cdot \{-U_x \sin \phi + U_y \cos \phi\} + (-\Omega_x \cos \theta \cos \phi + \Omega_y \cos \theta \sin \phi + \Omega_z \sin \theta) \quad (13)$$

$$+ \sum_{n=1}^{\infty} \sum_{m=0}^n \left[ \frac{\sin \theta P_n^{m'}}{r^{n+1}} (C_{mn} \cos m\phi + \tilde{C}_{mn} \sin m\phi) - \frac{m P_n^m}{\sin \theta} \left( \frac{n-2}{na^2 r^n} - \frac{1}{r^{n+2}} \right) (\tilde{B}_{mn} \cos m\phi - B_{mn} \sin m\phi) \right]$$

In order to find  $\mathbf{U}$  and  $\boldsymbol{\Omega}$ , we first compute all the forces and torques acting on the squirmer:  $\mathbf{F}_{squirm}$ ,  $\mathbf{T}_{squirm}$  is the frictional force and torque acting on the boundary of the squirmer due to the squirming actions.  $\mathbf{F}_{swim}$ ,  $\mathbf{T}_{swim}$  is the drag force and torque due to the squirmer swimming and rotating with  $\mathbf{U}$  and  $\boldsymbol{\Omega}$ . The drag is simply the stokes drag  $\mathbf{F}_{swim} = -6\pi\mu a \mathbf{U}$  and  $\mathbf{T}_{swim} = -8\pi\mu a^3 \boldsymbol{\Omega}$  [5]. The influence of gravity and other body forces has been neglected. When the squirmer is swimming and rotating with  $\mathbf{U}$ ,  $\boldsymbol{\Omega}$ , the overall force and torque must sum up to zero:

$$\mathbf{F}_{squirm} + \mathbf{F}_{swim} = \mathbf{F}_{squirm} - 6\pi\mu a \mathbf{U} = \mathbf{0} \quad (14)$$

$$\mathbf{T}_{squirm} + \mathbf{T}_{swim} = \mathbf{T}_{squirm} - 8\pi\mu a^3 \boldsymbol{\Omega} = \mathbf{0} \quad (15)$$

From the definition of the stress tensor,  $\boldsymbol{\Pi}$ , the force  $\mathbf{F}_{squirm}$  can be found:

$$\mathbf{F}_{squirm} = \iint_S \boldsymbol{\Pi} \cdot d\mathbf{S} = \iint_S \boldsymbol{\Pi} \cdot \hat{\mathbf{r}} dS = \iint_S \boldsymbol{\Pi}_r dS \quad (16)$$

$d\mathbf{S}$  is an element of surface area on the sphere pointing in the direction of  $\hat{\mathbf{r}}$  and  $dS = a^2 \sin \theta d\theta d\phi$ . The stress tensor dotted with the radial vector,  $\boldsymbol{\Pi}_r$ , is also called the stress vector or traction vector, and in spherical coordinates, it is given by [6]:

$$\boldsymbol{\Pi}_r = -\frac{\mathbf{r}}{r} p + \mu \left( \frac{\partial \mathbf{v}}{\partial r} - \frac{\mathbf{v}}{r} \right) + \frac{\mu}{r} \nabla(\mathbf{r} \cdot \mathbf{v}) \quad (17)$$

Since force acting on an element of surface area,  $dS$ , is  $\boldsymbol{\Pi}_r dS$ . The torque on the sphere is:

$$\mathbf{T}_{squirm} = \iint_S \mathbf{r} \times \boldsymbol{\Pi}_r dS \quad (18)$$

### 2.2.1 Calculating the traction vector

We will now calculate the traction vector,  $\mathbf{\Pi}_r$ , and in order to do so, some useful derivatives of a spherical harmonic are introduced. Let  $\Gamma_{-n-1}$  be an arbitrary spherical harmonic of the form:

$$\Gamma_{-n-1} = r^{-n-1} \sum_{m=0}^n P_n^m(\cos \theta) [K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi] \quad (19)$$

Then:

**SH 1** *Radial derivative of SH*

$$\frac{\partial}{\partial r}(\Gamma_{-n-1}) = -\frac{n+1}{r} \Gamma_{-n-1} \quad (20)$$

**SH 2** *Radial derivative of the gradient of SH*

$$\frac{\partial}{\partial r}(\nabla \Gamma_{-n-1}) = -\frac{n+2}{r} \nabla \Gamma_{-n-1} \quad (21)$$

**SH 3** *Radial derivative of the curl of SH multiplied by radial vector*

$$\frac{\partial}{\partial r}(\nabla \times \mathbf{r} \Gamma_{-n-1}) = -\frac{n+1}{r} \nabla \times (\mathbf{r} \Gamma_{-n-1}) \quad (22)$$

The full derivation of the above derivatives can be found in appendix (A.1). Now,  $\mathbf{\Pi}_r$  in Eq. 17, can be found term by term, starting with  $\frac{\partial \mathbf{v}}{\partial r}$ . From Eq. (1), we get four terms:

$$\frac{\partial \mathbf{u}}{\partial r} = \underbrace{\frac{\partial}{\partial r} \left( -\frac{n-2}{2\mu n(2n-1)} r^2 \nabla p_{-n-1} \right)}_{\text{Term 1}} + \underbrace{\frac{\partial}{\partial r} \left( \frac{n+1}{\mu n(2n-1)} \mathbf{r} p_{-n-1} \right)}_{\text{Term 2}} + \underbrace{\frac{\partial}{\partial r} (\nabla \Phi_{-n-1})}_{\text{Term 3}} + \underbrace{\frac{\partial}{\partial r} (\nabla \times (\mathbf{r} \chi_{-n-1}))}_{\text{Term 4}} \quad (23)$$

Term 1 in (23) is found using the product rule and SH2:

$$\frac{\partial}{\partial r} \left( -\frac{n-2}{2\mu n(2n-1)} r^2 \nabla p_{-n-1} \right) = -\frac{n-2}{2\mu n(2n-1)} \left( 2r \nabla p_{-n-1} + r^2 \left[ -\frac{n+2}{r} \nabla p_{-n-1} \right] \right) \quad (24)$$

$$= \frac{n(n-2)}{2\mu n(2n-1)} r \nabla p_{-n-1} \quad (25)$$

Term 2 in (23) is found with the product rule and SH1:

$$\frac{\partial}{\partial r} \left( \frac{n+1}{\mu n(2n-1)} \mathbf{r} p_{-n-1} \right) = \frac{n+1}{\mu n(2n-1)} \left( \frac{\mathbf{r}}{r} p_{-n-1} - \mathbf{r} \frac{n+1}{r} p_{-n-1} \right) \quad (26)$$

$$= -\frac{n(n+1)}{\mu n(2n-1)} \frac{\mathbf{r}}{r} p_{-n-1} \quad (27)$$

Term 3 in (23) is found using SH2:

$$\frac{\partial}{\partial r} (\nabla \Phi_{-n-1}) = -\frac{n+2}{r} \nabla \Phi_{-n-1} \quad (28)$$

Term 4 in (23) is found using SH3:

$$\frac{\partial}{\partial r} (\nabla \times (\mathbf{r} \chi_{-n-1})) = -\frac{n+1}{r} \nabla \times (\mathbf{r} \chi_{-n-1}) \quad (29)$$

Putting the four terms together, we have the expression for  $\frac{\partial \mathbf{u}}{\partial r}$ :

$$\frac{\partial \mathbf{v}}{\partial r} = \frac{n(n-2)}{2\mu n(2n-1)} r \nabla p_{-n-1} - \frac{n(n+1)}{\mu n(2n-1)} \frac{\mathbf{r}}{r} p_{-n-1} - \frac{n+2}{r} \nabla \Phi_{-n-1} - \frac{n+1}{r} \nabla \times (\mathbf{r} \chi_{-n-1}) \quad (30)$$

Next term in,  $\Pi_r$ , in Eq. (17) is  $\nabla(\mathbf{r} \cdot \mathbf{v})$ . The dot product,  $\mathbf{r} \cdot \mathbf{v}$ , gives four terms:

$$\mathbf{r} \cdot \mathbf{v} = \underbrace{-\frac{n-2}{2\mu n(2n-1)} r^2 \nabla p_{-n-1} \cdot \mathbf{r}}_{\text{Term 1}} + \underbrace{\frac{n+1}{\mu n(2n-1)} p_{-n-1} \mathbf{r} \cdot \mathbf{r}}_{\text{Term 2}} + \underbrace{\mathbf{r} \cdot \{\nabla \Phi_{-n-1}\}}_{\text{Term 3}} + \underbrace{\mathbf{r} \cdot \{\nabla \times (\mathbf{r} \chi_{-n-1})\}}_{\text{Term 4}} \quad (31)$$

Term 1 in (31) the radial component of  $\nabla p_{-n-1}$  is  $\frac{\partial}{\partial r} p_{-n-1}$ . Then using SH1:

$$-\frac{n-2}{2\mu n(2n-1)} r^2 \nabla p_{-n-1} \cdot \mathbf{r} = -\frac{n-2}{2\mu n(2n-1)} r^3 \frac{\partial}{\partial r} p_{-n-1} \quad (32)$$

$$= \frac{(n+1)(n-2)}{2\mu n(2n-1)} r^2 p_{-n-1} \quad (33)$$

Term 2 in (31) is simply  $\frac{n+1}{\mu n(2n-1)} p_{-n-1} r^2$ .

Term 3 in (31) is similar to the first term:

$$\mathbf{r} \cdot \{\nabla \Phi_{-n-1}\} = r \frac{\partial}{\partial r} \Phi_{-n-1} \quad (34)$$

$$= -(n+1) \Phi_{-n-1} \quad (35)$$

And term 4 is 0, since the curl of  $\mathbf{r} \chi_{-n-1}$  is orthogonal to  $\mathbf{r}$ . Combining the four terms, the dot product,  $\mathbf{r} \cdot \mathbf{v}$ , is:

$$\mathbf{r} \cdot \mathbf{v} = \frac{n(n+1)}{2\mu n} r^2 p_{-n-1} - (n+1) \Phi_{-n-1} \quad (36)$$

The gradient of this is:

$$\nabla(\mathbf{r} \cdot \mathbf{v}) = \frac{n(n+1)}{2\mu n(2n-1)} (2\mathbf{r} p_{-n-1} + r^2 \nabla p_{-n-1}) - (n+1) \nabla \Phi_{-n-1} \quad (37)$$

$\Pi_r$  can now be found by inserting the found expressions Eq. (1), (2), (30) and (37) into (17). After collecting the  $p_{-n-1}$ ,  $\nabla p_{-n-1}$ ,  $\nabla \Phi_{-n-1}$  and  $\nabla \times (\mathbf{r} \chi_{-n-1})$  terms, the final expression for the traction vector becomes:

$$\begin{aligned} \Pi_r = \frac{\mu}{r} \sum_{n=1}^{\infty} & \left\{ -(n+2) \nabla \times (\mathbf{r} \chi_{-n-1}) - 2(n+2) \nabla \Phi_{-n-1} \right. \\ & \left. - \frac{2n^2+1}{\mu n(2n-1)} \mathbf{r} p_{-n-1} + \frac{(n-1)(n+1)}{\mu n(2n-1)} r^2 \nabla p_{-n-1} \right\} \end{aligned} \quad (38)$$

Finally, we also need to find an expression for  $\mathbf{r} \times \Pi_r$ . Here we will introduce two useful cross products:

**SH 4** *Cross product rule 1*

$$\mathbf{r} \times (\nabla \times (\mathbf{r} \Gamma_{-n-1})) = r^2 \nabla \Gamma_{-n-1} + (n+1) \cdot \mathbf{r} \Gamma_{-n-1} \quad (39)$$

### SH 5 Cross product rule 2

$$\mathbf{r} \times \nabla \Gamma_{-n-1} = -\nabla \times (\mathbf{r} \Gamma_{-n-1}) \quad (40)$$

Again, derivations of the above relations can be found in the appendix A.2. With these, the expression for  $\mathbf{r} \times \Pi_r$  is obtained to be:

$$\begin{aligned} \mathbf{r} \times \Pi_r = \frac{\mu}{r} \sum_{n=1}^{\infty} \left\{ -(n+2)r^2 \nabla \chi_{-n-1} - (n+1)(n+2)\mathbf{r} \chi_{-n-1} + 2(n+2)\nabla \times (\mathbf{r} \Phi_{-n-1}) \right. \\ \left. - \frac{(n-1)(n+1)}{\mu n(2n-1)} r^2 \nabla \times (\mathbf{r} p_{-n-1}) \right\} \end{aligned} \quad (41)$$

#### 2.2.2 Calculating the frictional force and torque

To perform the integration of the traction vector over the sphere we will use the following surface integral theorems. We recall that  $\Gamma_{-n-1}$  is an arbitrary solid spherical harmonic given by equation (19) with  $0 < n$  and  $0 \leq m \leq n$ . Then:

$$\iint_S \mathbf{r} \Gamma_{-n-1} dS \Big|_{r=a} = \begin{cases} \frac{4}{3}\pi a K_{01} \hat{\mathbf{z}}, & \text{for } n = 1, m = 0 . \\ -\frac{4}{3}\pi a K_{11} \hat{\mathbf{x}} - \frac{4}{3}\pi a \tilde{K}_{11} \hat{\mathbf{y}}, & \text{for } n = 1, m = 1 . \\ 0, & \text{otherwise.} \end{cases} \quad (42)$$

$$\iint_S \nabla \Gamma_{-n-1} dS \Big|_{r=a} = 0 , \text{ for all } n > 0 \text{ and } n \geq m \geq 0 \quad (43)$$

$$\iint_S \nabla \times (\mathbf{r} \Gamma_{-n-1}) dS \Big|_{r=a} = 0 , \text{ for all } n > 0 \text{ and } n \geq m \geq 0 . \quad (44)$$

Proofs for the above integral theorems can be found in the appendix A.3. Using these integral theorems and the expression for  $\Pi_r$ ,  $\mathbf{F}_{squirm}$  is:

$$\mathbf{F}_{squirm} = \iint_S \Pi_r dS \quad (45)$$

$$\begin{aligned} &= \iint_S \frac{\mu}{r} \sum_{n=1}^{\infty} \left\{ -(n+2)\nabla \times (\mathbf{r} \chi_{-n-1}) - 2(n+2)\nabla \Phi_{-n-1} \right. \\ &\quad \left. - \frac{2n^2+1}{\mu n(2n-1)} \mathbf{r} p_{-n-1} + \frac{(n-1)(n+1)}{\mu n(2n-1)} r^2 \nabla p_{-n-1} \right\} dS \Big|_{r=a} \end{aligned} \quad (46)$$

$$= \frac{8\pi\mu}{a^2} (B_{11} \hat{\mathbf{x}} + \tilde{B}_{11} \hat{\mathbf{y}} - B_{01} \hat{\mathbf{z}}) \quad (47)$$

Here we have also used the relation given in (6). The torque  $\mathbf{T}_{squirm}$  can be found similarly:

$$\mathbf{T}_{squirm} = \iint_S \mathbf{r} \times \Pi_r dS \quad (48)$$

$$\begin{aligned} &= \iint_S \frac{\mu}{r} \sum_{n=1}^{\infty} \left\{ -(n+2)r^2 \nabla \chi_{-n-1} - (n+1)(n+2)\mathbf{r} \chi_{-n-1} + 2(n+2)\nabla \times (\mathbf{r} \Phi_{-n-1}) \right. \\ &\quad \left. - \frac{(n-1)(n+1)}{\mu n(2n-1)} r^2 \nabla \times (\mathbf{r} p_{-n-1}) \right\} dS \Big|_{r=a} \end{aligned} \quad (49)$$

$$= 8\pi\mu (C_{11} \hat{\mathbf{x}} + \tilde{C}_{11} \hat{\mathbf{y}} - C_{01} \hat{\mathbf{z}}) \quad (50)$$

Now that  $\mathbf{F}_{squirm}$  and  $\mathbf{T}_{squirm}$  have been obtained, the swimming velocity can be computed from the zero force and zero torque condition (14), (15):

$$\mathbf{U} = \frac{4}{3a^3} (B_{11}\hat{\mathbf{x}} + \tilde{B}_{11}\hat{\mathbf{y}} - B_{01}\hat{\mathbf{z}}) \quad (51)$$

$$\boldsymbol{\Omega} = \frac{1}{a^3} (C_{11}\hat{\mathbf{x}} + \tilde{C}_{11}\hat{\mathbf{y}} - C_{01}\hat{\mathbf{z}}) \quad (52)$$

Note that only 3 modes contribute to squirmer swimming. Mode  $B_{11}$  is responsible for the squirmer moving along the  $\hat{\mathbf{x}}$ -direction,  $\tilde{B}_{11}$  along the  $\hat{\mathbf{y}}$ -direction, and  $B_{01}$  the  $\hat{\mathbf{z}}$ -direction. Likewise, only three modes contribute to the squirmer rotating.

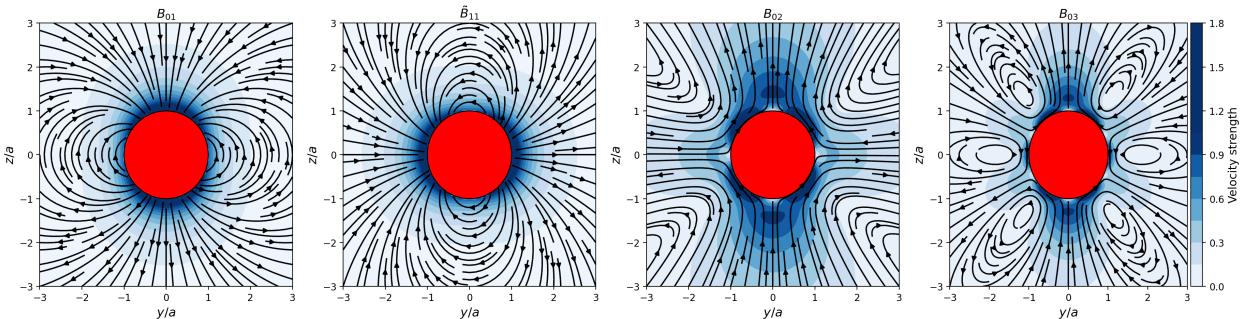
By inserting  $\mathbf{U}, \boldsymbol{\Omega}$  back into equation (11)-(13), the final expression for the overall field of a swimming squirmer in lab frame is:

$$v_r = \frac{4}{3r^3} (B_{11} \sin \theta \cos \phi + \tilde{B}_{11} \sin \theta \sin \phi + B_{01} \cos \theta) + \sum_{n=2}^{\infty} \sum_{m=0}^n \frac{(n+1)P_n^m}{r^{n+2}} \left( \frac{r^2}{a^2} - 1 \right) [B_{mn} \cos m\phi + \tilde{B}_{mn} \sin \phi] \quad (53)$$

$$v_\theta = -\frac{2}{3r^3} (B_{11} \cos \theta \cos \phi + \tilde{B}_{11} \cos \theta \sin \phi + B_{01} \sin \theta) + \sum_{n=2}^{\infty} \sum_{m=0}^n \left[ \sin \theta P_n^{m'} \left( \frac{n-2}{na^2 r^n} - \frac{1}{r^{n+2}} \right) (B_{mn} \cos m\phi + \tilde{B}_{mn} \sin \phi) + \frac{m P_n^m}{r^{n+1} \sin \theta} (\tilde{C}_{mn} \cos m\phi - C_{mn} \sin m\phi) \right] \quad (54)$$

$$v_\phi = \frac{2}{3r^3} (B_{11} \sin \phi - \tilde{B}_{11} \cos \phi) + \sum_{n=2}^{\infty} \sum_{m=0}^n \left[ \frac{\sin \theta P_n^{m'}}{r^{n+1}} (C_{mn} \cos m\phi + \tilde{C}_{mn} \sin m\phi) - \frac{m P_n^m}{\sin \theta} \left( \frac{n-2}{na^2 r^n} - \frac{1}{r^{n+2}} \right) (\tilde{B}_{mn} \cos m\phi - B_{mn} \sin m\phi) \right] \quad (55)$$

Note that the modes  $C_{11}, \tilde{C}_{11}, C_{01}$  are absent in the above equations, as all terms including them have been cancelled out due to the torque-free condition. Physically, it means the purely rotational modes are not torque-free; hence we do not allow them. In Figure 2, the velocity field for the first few modes is plotted in the lab frame. The continuous lines are streamlines, and the colour density represents the magnitude of flow velocity. The red circle is the squirmer.



**Figure 2:** The velocity generated by the squirmer for the first few modes has been plotted in the lab frame. The continuous lines are streamlines, and the colour density represents the magnitude of flow velocity. The red circle is the squirmer. Mode  $B_{01}, \tilde{B}_{11}$  are source dipoles and are responsible for the squirmer moving along the  $\hat{\mathbf{z}}, \hat{\mathbf{y}}$  direction, respectively.  $B_{02}$  is a source quadrupole which represents a shaking squirmer.

### 3 Numerical Methods

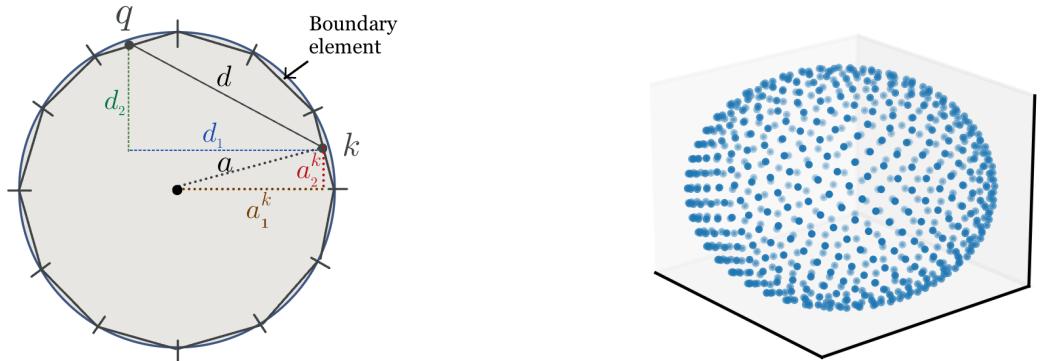
So far, we have only been dealing with the analytical solution of a swimming squirmer. There is, however, no analytical solution to the fluid field when there is more than one sphere. In the presence of a target, we, therefore, consider a numerical method for solving the fluid field.

#### 3.1 Boundary Element Method (BEM)

The idea behind the Boundary Element Method ("BEM") is to first discretize the boundary into  $N$  boundary elements (See Figure 3a). Assuming that the velocity  $\mathbf{v}$  is constant over each boundary element, the velocity values on the boundary elements are used to calculate the forces,  $\mathbf{F}$ , acting on each of the boundary elements. More specifically, we solve this set of linear equations [9]:

$$\mathbf{v} = \frac{\Delta A}{8\pi\mu} \mathbf{S} \cdot \mathbf{F} \quad (56)$$

Here  $\mathbf{S}$  is the Oseen tensor, and its construction will be discussed below, and  $\Delta A$  is the area of each boundary element. Given that we know the forces  $\mathbf{F}$  acting on each of the boundary elements, a new Oseen tensor can be computed at an arbitrary point in space. Then by matrix multiplication:  $\mathbf{v} = \frac{\Delta A}{8\pi\mu} \mathbf{S} \cdot \mathbf{F}$ , the velocity at that point in space can be computed.



(a) The boundary of a 2D circle has been discretized into  $N$  boundary elements.  $d$  is the distance between two boundary elements  $k$  and  $q$ , and  $d_i$  is the distance between the two points in the  $i$ 'th direction.  $a_1^k, a_2^k$  is the coordinate of the  $k$ 'th boundary element relative to the center of the sphere.

(b) Discretization of a sphere using the canonical Fibonacci method with  $N = 700$  surface points. We assume the points are evenly spaced so  $\Delta A = 4\pi a^2/N$ .

**Figure 3**

##### 3.1.1 The Oseen tensor

The Oseen tensor becomes rather complex when it includes more than one object and with the requirement that the objects are force and torque-free. As to not get confused about its construction, we will start with the simplest case, being that of only one object and no extra constraints, and then in steps go towards the end goal including all the above mentioned. We will use the regularized Stokeslet method by D. J. Smith [10]. He introduces the Oseen tensor as:

$$S_{ij} = \frac{\delta_{ij}(d^2 + 2\epsilon^2) + d_i d_j}{d_\epsilon^3} \quad (57)$$

Here  $d = \sqrt{d_1^2 + d_2^2 + d_3^2}$  is the total distance between two boundary elements  $k$  and  $q$ , with  $d_i$  being the distance between the two points in the  $i$ 'th direction (See Figure 3a), and  $d_\epsilon = \sqrt{d^2 + \epsilon^2}$  includes a small number  $\epsilon$  called the regularization offset. This ensures that we never divide by zero in equation (57). The choice of  $\epsilon$  will be discussed in section (3.2). The canonical Fibonacci lattice method [11] is used to discretize the spherical squirmer surface into  $N$  approximately evenly distributed points (See Figure 3b). We then calculate how each of the  $N$  surface points interacts

with all other points through the use of Eq. (57). With  $N$  points for each of the three coordinates,  $\mathbf{S}$  is a  $3N$  by  $3N$  matrix, given by:

$$\mathbf{S} = \begin{bmatrix} S_{11}^{11} & \cdots & S_{11}^{1N} & S_{12}^{11} & \cdots & S_{12}^{1N} & S_{13}^{11} & \cdots & S_{13}^{1N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{11}^{N1} & \cdots & S_{11}^{NN} & S_{12}^{N1} & \cdots & S_{12}^{NN} & S_{13}^{N1} & \cdots & S_{13}^{NN} \\ \hline S_{21}^{11} & \cdots & S_{21}^{1N} & S_{22}^{11} & \cdots & S_{22}^{1N} & S_{23}^{11} & \cdots & S_{23}^{1N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{21}^{N1} & \cdots & S_{21}^{NN} & S_{22}^{N1} & \cdots & S_{22}^{NN} & S_{23}^{N1} & \cdots & S_{23}^{NN} \\ \hline S_{31}^{11} & \cdots & S_{31}^{1N} & S_{32}^{11} & \cdots & S_{32}^{1N} & S_{33}^{11} & \cdots & S_{33}^{1N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{31}^{N1} & \cdots & S_{31}^{NN} & S_{32}^{N1} & \cdots & S_{32}^{NN} & S_{33}^{N1} & \cdots & S_{33}^{NN} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{S}}_{11} & \bar{\mathbf{S}}_{12} & \bar{\mathbf{S}}_{13} \\ \bar{\mathbf{S}}_{21} & \bar{\mathbf{S}}_{22} & \bar{\mathbf{S}}_{23} \\ \bar{\mathbf{S}}_{31} & \bar{\mathbf{S}}_{32} & \bar{\mathbf{S}}_{33} \end{bmatrix} \quad (58)$$

The lines in the above matrix are purely a visual aid. In  $S_{ij}^{kq}$ , subscripts indicate *coordinates*, while superscripts are *points*. Note that the matrix is symmetrical, as  $d_id_j = d_jd_i$  in (57), and all other terms are magnitudes. By defining  $\bar{\mathbf{S}}_{ij}$  as the  $N$  by  $N$  matrix with all points' interactions between the  $i$ 'th and  $j$ 'th coordinate, the Oseen tensor can be written more compactly as seen in the right-hand side of Eq. (58)

We now consider when there are two spheres with different radii. The first sphere has  $N_1$  boundary elements, and the second sphere has  $N_2$  boundary elements.  $N_1$  and  $N_2$  are related, such that  $\Delta A$  is the same for both spheres, meaning  $N_2 = (a_2^2/a_1^2)N_1$ . Let  ${}^{12}\bar{\mathbf{S}}_{ij}$  serve as the two-object analogue to the one-object  $\bar{\mathbf{S}}_{ij}$  in Eq. (58), where instead of all boundary elements on one object interact with each other,  ${}^{12}\bar{\mathbf{S}}_{ij}$  is how each boundary elements on object 1 interacts with all other boundary elements on object 2. With this notation,  ${}^{11}\bar{\mathbf{S}}_{ij}$  represents how each boundary elements on object 1 interacts with itself, and similarly for  ${}^{22}\bar{\mathbf{S}}_{ij}$ . Lastly,  ${}^{21}\bar{\mathbf{S}}_{ij}$  represents how each boundary element on object 2 interacts with all other boundary elements on object 1. Note that  ${}^{12}\bar{\mathbf{S}}_{ij} = {}^{21}\bar{\mathbf{S}}_{ji}$ . With this notation, we can write up  ${}^{11}\bar{\mathbf{S}}$ ,  ${}^{12}\bar{\mathbf{S}}$ ,  ${}^{21}\bar{\mathbf{S}}$  and  ${}^{22}\bar{\mathbf{S}}$  as:

$${}^{11}\bar{\mathbf{S}} = \underbrace{\begin{bmatrix} {}^{11}\bar{\mathbf{S}}_{11} & {}^{11}\bar{\mathbf{S}}_{12} & {}^{11}\bar{\mathbf{S}}_{13} \\ {}^{11}\bar{\mathbf{S}}_{21} & {}^{11}\bar{\mathbf{S}}_{22} & {}^{11}\bar{\mathbf{S}}_{23} \\ {}^{11}\bar{\mathbf{S}}_{31} & {}^{11}\bar{\mathbf{S}}_{32} & {}^{11}\bar{\mathbf{S}}_{33} \end{bmatrix}}_{3N_1}, \quad {}^{12}\bar{\mathbf{S}} = \underbrace{\begin{bmatrix} {}^{12}\bar{\mathbf{S}}_{11} & {}^{12}\bar{\mathbf{S}}_{12} & {}^{12}\bar{\mathbf{S}}_{13} \\ {}^{12}\bar{\mathbf{S}}_{21} & {}^{12}\bar{\mathbf{S}}_{22} & {}^{12}\bar{\mathbf{S}}_{23} \\ {}^{12}\bar{\mathbf{S}}_{31} & {}^{12}\bar{\mathbf{S}}_{32} & {}^{12}\bar{\mathbf{S}}_{33} \end{bmatrix}}_{3N_2} \quad (59)$$

$${}^{21}\bar{\mathbf{S}} = \underbrace{\begin{bmatrix} {}^{21}\bar{\mathbf{S}}_{11} & {}^{21}\bar{\mathbf{S}}_{12} & {}^{21}\bar{\mathbf{S}}_{13} \\ {}^{21}\bar{\mathbf{S}}_{21} & {}^{21}\bar{\mathbf{S}}_{22} & {}^{21}\bar{\mathbf{S}}_{23} \\ {}^{21}\bar{\mathbf{S}}_{31} & {}^{21}\bar{\mathbf{S}}_{32} & {}^{21}\bar{\mathbf{S}}_{33} \end{bmatrix}}_{3N_2}, \quad {}^{22}\bar{\mathbf{S}} = \underbrace{\begin{bmatrix} {}^{22}\bar{\mathbf{S}}_{11} & {}^{22}\bar{\mathbf{S}}_{12} & {}^{22}\bar{\mathbf{S}}_{13} \\ {}^{22}\bar{\mathbf{S}}_{21} & {}^{22}\bar{\mathbf{S}}_{22} & {}^{22}\bar{\mathbf{S}}_{23} \\ {}^{22}\bar{\mathbf{S}}_{31} & {}^{22}\bar{\mathbf{S}}_{32} & {}^{22}\bar{\mathbf{S}}_{33} \end{bmatrix}}_{3N_2} \quad (60)$$

The two object the Oseen tensor,  $\mathbf{S}$ , is therefore:

$$\mathbf{S} = \begin{bmatrix} {}^{11}\bar{\mathbf{S}} & {}^{12}\bar{\mathbf{S}} \\ {}^{21}\bar{\mathbf{S}} & {}^{22}\bar{\mathbf{S}} \end{bmatrix} \quad (61)$$

Let us now consider the left-hand side of Eq. (56). It consists of the velocity components of object 1 and object 2. For the first object:  ${}^1\mathbf{v}^k = {}^1\mathbf{u}^k + {}^1\mathbf{U} + {}^1\boldsymbol{\Omega} \times {}^1\mathbf{a}^k$ . Here  ${}^1\mathbf{v}^k$  is the velocity of the  $k$ 'th boundary element in the lab frame for object 1, while  ${}^1\mathbf{u}^k$  is the velocity of the  $k$ 'th boundary element in the squirmer frame for object 1.  ${}^1\mathbf{a}^k = ({}^1a_1^k, {}^1a_2^k, {}^1a_3^k)$  is the coordinate of the  $k$ 'th boundary element relative to the centre of sphere 1 (see Figure 3a).  ${}^1\mathbf{U}$  and  ${}^1\boldsymbol{\Omega}$  are the translational velocity and rotational velocity of object 1. Object 2 is similar:  ${}^2\mathbf{v}^k = {}^2\mathbf{u}^k + {}^2\mathbf{U} + {}^2\boldsymbol{\Omega} \times {}^2\mathbf{a}^k$ .  ${}^1\mathbf{U}$ ,  ${}^2\mathbf{U}$ ,  ${}^1\boldsymbol{\Omega}$  and  ${}^2\boldsymbol{\Omega}$  are unknowns. By rearranging and moving  $\mathbf{U}$  and  $\boldsymbol{\Omega}$  to the other side of Eq.

(56), we have:  ${}^1\mathbf{u}^k = {}^1\mathbf{v}^k - {}^1\mathbf{U} - {}^1\boldsymbol{\Omega} \times {}^1\mathbf{a}^k$  and  ${}^2\mathbf{u}^k = {}^2\mathbf{v}^k - {}^2\mathbf{U} - {}^2\boldsymbol{\Omega} \times {}^2\mathbf{a}^k$ . When we also require the forces and torques to sum to 0, the Oseen tensor at the right-hand side of Eq. (56) becomes (in the compacted notation):

$$\mathbf{S} = \begin{bmatrix} {}^{11}\mathbf{S} & {}^{12}\mathbf{S} & -{}^1\mathbf{C}^\top & \mathbf{0} \\ {}^{21}\mathbf{S} & {}^{22}\mathbf{S} & \mathbf{0} & -{}^2\mathbf{C}^\top \\ {}^1\mathbf{C} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^2\mathbf{C} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (62)$$

The boldface zeros represent a matrix full of zeros with the needed shape, and the constraint matrix  ${}^1\mathbf{C}$  for sphere 1 is:

$${}^1\mathbf{C} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & {}^1a_3^1 & {}^1a_3^2 & \cdots & {}^1a_3^{N_1} & -{}^1a_2^1 & -{}^1a_2^2 & \cdots & -{}^1a_2^{N_1} \\ -{}^1a_3^1 & -{}^1a_3^2 & \cdots & -{}^1a_3^{N_1} & 0 & 0 & \cdots & 0 & -{}^1a_1^1 & -{}^1a_1^2 & \cdots & -{}^1a_1^{N_1} \\ -{}^1a_2^1 & -{}^1a_2^2 & \cdots & -{}^1a_2^{N_1} & {}^1a_1^1 & {}^1a_1^2 & \cdots & {}^1a_1^{N_1} & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (63)$$

The first three rows in  ${}^1\mathbf{C}$  are for the zero-force condition and the last three rows are for the torque-free condition. The second sphere's constraint matrix,  ${}^2\mathbf{C}$ , is constructed similarly. To actually enforce the zero-force and zero-torque condition, 12 zeros are added at the bottom of the vector  $\mathbf{u}$ . The force vector will also be 12 elements longer. The first three of the additional force elements are the translational velocity of sphere 1,  ${}^1\mathbf{U}$ , the next three elements are the angular velocity of sphere 1,  ${}^2\mathbf{U}$ , and the last six elements are the translational velocity and angular velocity for sphere 2. The boundary values  ${}^1\mathbf{u} = {}^1u_1^1 \cdots {}^1u_3^{N_1}$  for sphere 1 is the velocity field discussed in the previous chapter, in equation (7) - (9). For the second sphere, which is a passive drifting sphere, the boundary values  ${}^2\mathbf{u} = {}^2u_1^1 \cdots {}^2u_3^{N_2}$  are set to zero, due to the no-slip condition. So the vectors  $\mathbf{u}$  and  $\mathbf{F}$  become:

$$\mathbf{u} = [{}^1u_1^1 \cdots {}^1u_3^{N_1} \ 0 \cdots 0 \ 000 \ 000 \ 000 \ 000]^\top \quad (64)$$

$$\mathbf{F} = [{}^1F_1^1 \cdots {}^1F_3^{N_1} \ {}^2F_1^1 \cdots {}^2F_3^{N_2} \ {}^1U_1 {}^1U_2 {}^1U_3 \ {}^2\omega_1 {}^2\omega_2 {}^2\omega_3 \ {}^2U_1 {}^2U_2 {}^2U_3 \ {}^2\omega_1 {}^2\omega_2 {}^2\omega_3]^\top \quad (65)$$

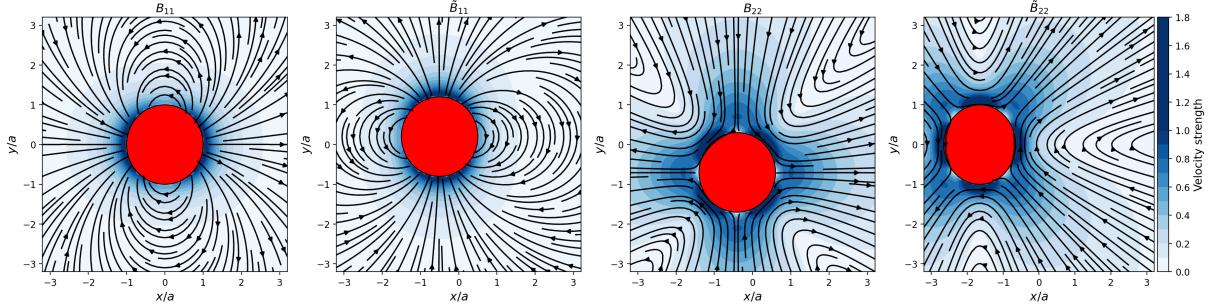
Letting  ${}^1\mathbf{F}$  denote  ${}^1F_1^1 \cdots {}^1F_3^{N_1}$  and  ${}^1\mathbf{U}$  denote  ${}^1U_1 {}^1U_2 {}^1U_3$  and so on, the set of linear equations we are solving is:

$$\begin{bmatrix} {}^1\mathbf{u} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \frac{\Delta A}{8\pi\mu} \begin{bmatrix} {}^{11}\mathbf{S} & {}^{12}\mathbf{S} & -{}^1\mathbf{C}^\top & \mathbf{0} \\ {}^{21}\mathbf{S} & {}^{22}\mathbf{S} & \mathbf{0} & -{}^2\mathbf{C}^\top \\ {}^1\mathbf{C} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^2\mathbf{C} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} {}^1\mathbf{F} \\ {}^2\mathbf{F} \\ {}^1\mathbf{U} \\ {}^1\boldsymbol{\Omega} \\ {}^2\mathbf{U} \\ {}^2\boldsymbol{\Omega} \end{bmatrix} \quad (66)$$

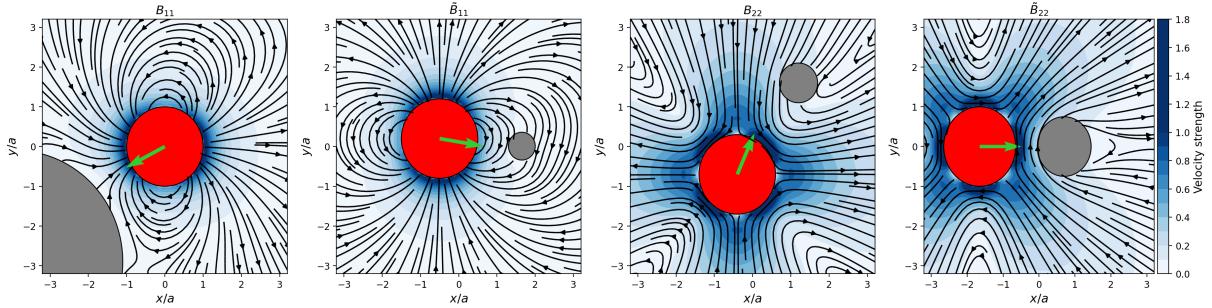
In Figure 4a and 4b the velocity field has been plotted for different modes, in the presence of only a squirmer and in the presence of both a squirmer and a passive particle. Both cases are solved using the boundary element method, using  $a = 1, N_1 = 500, \epsilon = 0.05$ .

### 3.2 Validation and choice of parameters values

We compute the translational velocity and field velocity numerically in the case of only a squirmer and compare them with the analytical expression. We set  $B_{11} = 1$  and  $\tilde{B}_{11} = 1$ , which means the squirmer moves diagonally in the xy-plane. To measure the accuracy of the numerical field, we find the average of the relative error between the numerically computed velocity field and the analytical



(a) Velocity field due to different modes solved with the Boundary Element Method in the presence of only the squirmer (red circle).



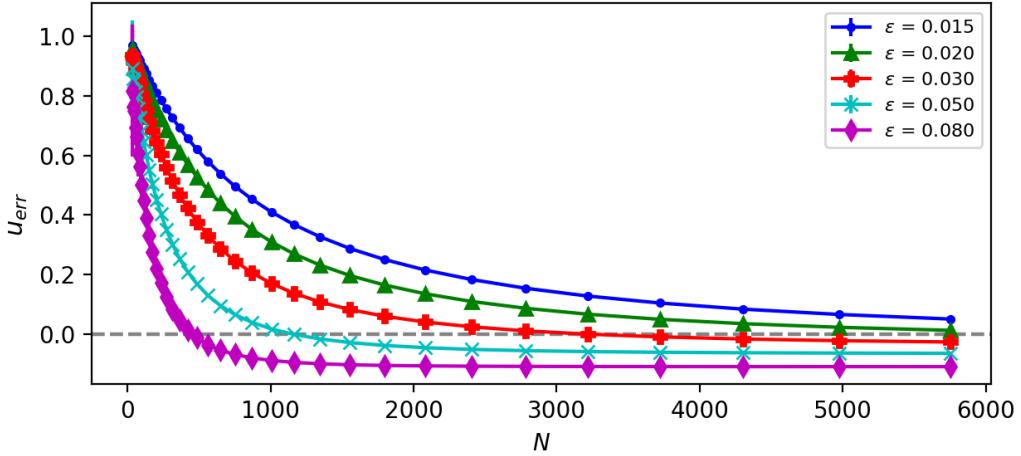
(b) Velocity field solved with the Boundary Element Method in the presence of both a squirmer (red circle) and another passive particle (grey circle). The green vector at the centre of the squirmer points in the direction of the average change in the difference between the force field with and without the passive particle.

**Figure 4:** The squirmer activates the same modes in (a) and (b). The continuous lines are streamlines and the colour density represents the magnitude of flow velocity.

velocity field in the  $xy$ -plane of size  $50a \times 50a$  and with  $100 \times 100$  grid points. We calculate the sample mean of the relative error component-wise:

$$\mathbf{u}_{err} = \text{mean}\left(\frac{\mathbf{u}_{ana} - \mathbf{u}_{num}}{\mathbf{u}_{ana}}\right) \quad (67)$$

We calculate it component-wise to ensure that not only is the size of the vector field compatible, but also the direction. In Figure 5 we have plotted this for both  $x$  and  $y$ -direction together with its uncertainty as error bars. The uncertainty has an order of magnitude of  $\mathcal{O}(10^{-4})$ , indicating that the relative error across the plane is almost constant for a given  $N$  and  $\epsilon$ . The graphs for  $x$  and  $y$ -direction for a given  $\epsilon$  have been coloured the same and lie almost on top of each other. This indicates that when there is a small relative error, not only does the magnitude of the vector field match but so does the directions. We interpret a higher  $N$  as higher precision and a lower  $\epsilon$  as being more accurate. Figure 5 shows that a large  $\epsilon$  gives an inaccurate result, even for high  $N$  values. The relative error line crosses the  $u_{err} = 0$  for high  $\epsilon$ . We interpret this as a numerical coincidence, and *not* that large  $\epsilon$  values are better at a low  $N$  value. The choice of  $N$  and  $\epsilon$  has a significant impact on the relative error of the numerical velocity field. Figure 5 shows that the higher the  $N$  and the lower the  $\epsilon$ , the closer the numerical is to the analytical field. However, a lower  $\epsilon$  requires a higher  $N$  to achieve the same relative error. The issue is, then, that a higher  $N$  requires a higher computation time, thus a proper balance between  $N, \epsilon$  is needed. We have, therefore, also investigated the computation time, which we found to increase exponentially with  $N$ . Taking all of the above into consideration, we have chosen  $\epsilon = 0.05$  and  $N = 700$ . We picked  $N = 700$  as this is the highest  $N$  that still yields an acceptable reinforcement learning training time, which will be introduced in the following section. Given that  $N = 700$ , we then chose  $\epsilon = 0.05$ , as this gives a relative error of  $u_{err} = 0.0788 \pm 0.0006$ , which we deem to be acceptable. A smaller  $\epsilon$  will have a greater relative error at  $N = 700$ , and a larger  $\epsilon$  may have a smaller relative error, but we consider it to be a worse approximation.



**Figure 5:** The average of the relative error between the numerically computed field and the analytical field as a function of  $N$  for different  $\epsilon$ . For big  $\epsilon$  values, there is a systematic error between the numerical and analytical field even for high  $N$ . The error bars show the standard deviation of the relative error. The error bars are however too small to be seen.

### 3.3 Constant power

From a physical or biological perspective, a squirmer cannot use an infinite amount of power. We will therefore unless stated otherwise, model the squirmer as having constant power. This means it cannot activate all its modes at full strength simultaneously. We choose this power constant to be  $P_{\text{const}} = \frac{64\pi\mu}{3a^5}$ , which, as seen in Eq. (10), means the lowest three modes after the transformation given in (68) satisfy:  $\max(|B_{01}|) = \max(|B_{11}|) = \max(|\tilde{B}_{11}|) = 1$ , as in Zhu et al., 2022 [3]. If one chooses the more straightforward choice of  $P_{\text{const}} = 1$ , the lowest three modes after the transformation given in (68) would satisfy:  $\max(|B_{01}|) = \max(|B_{11}|) = \max(|\tilde{B}_{11}|) < 1$ , and we would not be able to compare our results with [3]. The squirmer could in theory use less power than  $P_{\text{const}}$ , but in practice, it will always be suboptimal to do so, and thus our squirmer has constant power. To enforce the constant power, the modes chosen by the squirmer are normalized in the following way:

$$\begin{aligned} B_{mn} &\longrightarrow B_{mn} \sqrt{\frac{P_{\text{const}}}{P(B_{mn}, \tilde{B}_{mn}, C_{mn}, \tilde{C}_{mn})}} \\ \tilde{B}_{mn} &\longrightarrow \tilde{B}_{mn} \sqrt{\frac{P_{\text{const}}}{P(B_{mn}, \tilde{B}_{mn}, C_{mn}, \tilde{C}_{mn})}} \end{aligned} \quad (68)$$

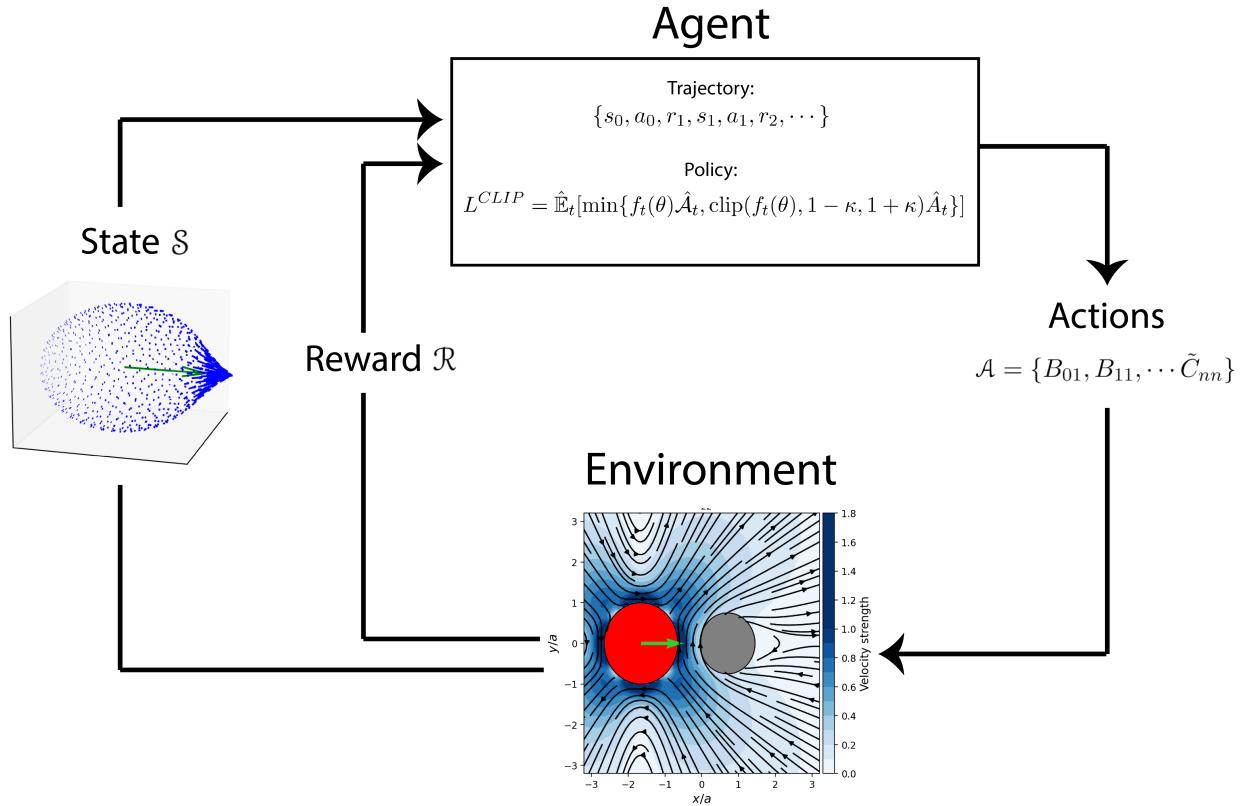
And likewise for  $C_{mn}$  and  $\tilde{C}_{mn}$ .  $P(B_{mn}, \tilde{B}_{mn}, C_{mn}, \tilde{C}_{mn})$  is the power given in equation (10). These new, normalized modes, are used to calculate the velocity field in Eq. (53)-(55). Because of the power constraint, the squirmer has to distribute its power amongst the different modes. The interesting part, which the next section is dedicated to, is now whether the squirmer can get away with choosing the cheap translation modes for sensing, or if it has to use the (generally) more costly higher modes, which would prevent it from moving efficiently.

## 4 Reinforcement Learning

### 4.1 Essential reinforcement learning and the squirmer model

Reinforcement learning ("RL") is a branch of machine learning, which through interactions between an agent and its environment learns an optimal course of action for said agent. This is illustrated in Figure 6, where at a time  $t$  the agent (the squirmer) chooses amongst a set of available actions  $a_t \in \mathcal{A}$  (the modes), which influence the environment (the velocity field with both squirmer and target), who provides the scalar reward  $R_{t+1} \in \mathcal{R}$  and state  $s_{t+1} \in \mathcal{S}$  (either position or average squirmer force field difference) that the chosen actions resulted in, which in turn lets the agent decide its next set of actions and so on. In this section, we first briefly explain the policy algorithm used and then dive into the two RL problems. The first investigates how a squirmer optimally catches its target with perfect sensing and the second how imperfect sensing affects mode choices. In section 5.1 the results of the RL training are presented, and in section 6.2.1 the case of catching the target with imperfect sensing will be discussed.

All coding was done in Python using the Gym library [12] for creating the agent-environment interaction and the Stable Baselines 3 ("SB3") library [13] for training. The hyperparameters used and library versions can be seen in Table 1 at the end of this section.



**Figure 6:** The agent-environment interaction for the sensing scenario. Inspired by Figure 3.1 in [1].

#### 4.1.1 Proximal Policy Optimization

Figure 6 does not give the policy quite enough credit, as in most cases an RL problem is considered solved when the optimal policy is found [1], which means tuning parameters until the maximum reward over time is found. More formally, the policy  $\pi$  is a mapping from the current agent state to action probabilities, so  $\pi(a|s)$  is the probability of taking the action  $a_t = a$  if the state is  $s_t = s$  [1]. We use the Proximal Policy Optimization "PPO" algorithm, which will now be explained in broad strokes. PPO is roughly a policy gradient optimization algorithm. The default policy gradient loss

$L^{PG}$  can be written in terms of expectations [14]:

$$L^{PG} = \hat{\mathbb{E}}_t[\log \pi_\Theta(a_t|s_t) \hat{A}_t] \quad (69)$$

The policy  $\pi_\Theta$ , with parameters  $\Theta$ , is a neural network, which as usual, takes the agent state and returns action probabilities.  $\hat{A}_t$  is the estimator of the *advantage function*, informally given by [15]:

$$\hat{A}_t = \text{Discounted rewards} - \text{Baseline estimate} \quad (70)$$

The discounted rewards look at the actual rewards gained throughout the episode, while the baseline estimate is a prediction of what reward the agent would get following the current policy in an episode. So  $\hat{A}_t$  is a measure of how good the total reward received was compared to the expected reward following  $\pi_\Theta$ . This means if  $\hat{A}_t$  is positive, the gradient (in its expectation form) in Eq. (69) is positive and the probability of choosing the action  $a_t$  will be increased, and vice versa. The issue that PPO attends to is that the baseline estimate is noisy, and the noise will accumulate with increasing time steps, which will eventually ruin the policy [15]. PPO does so by limiting how great the advantage function can be, directly in the objective function [14]:

$$L^{CLIP} = \hat{\mathbb{E}}_t[\min\{f_t(\Theta)\hat{A}_t, \text{clip}(f_t(\Theta), 1 - \kappa, 1 + \kappa)\hat{A}_t\}] \quad (71)$$

The log probability is replaced by  $f_t(\Theta) = \pi_\Theta(a_t|s_t)/\pi_{\Theta_{\text{old}}}(a_t|s_t)$ , which is how likely actions are in the new policy compared to the old. Thus, if both  $f_t(\Theta)$  and  $\hat{A}_t$  are large and positive, meaning the actions taken were both more probable and scored higher than the expected reward, the policy will be changed greatly towards favouring that action and vice versa. However, to prevent the aforementioned issue of noise, the clip function is introduced. Together with the hyperparameter  $\kappa$  (Schulman et al. suggest  $\kappa = 0.2$  [14]), the clipping both prevents updates from becoming too great (through  $1 + \kappa$ ), and the negative changes are allowed to be large but not 0 (through  $1 - \kappa$ ). Eq. (71), along with two other terms which promote exploration, is then what the PPO method optimizes.

## 4.2 Catching a point target with perfect sensing

The agent is given the target's exact location relative to itself in polar coordinates,  $\mathcal{S} = \{r, \theta\}$ , and access to two translation modes,  $\mathcal{A} = \{B_{01}, \tilde{B}_{11}\}$  in the  $yz$ -plane with  $B_{01}, \tilde{B}_{11} \in [-1, 1]$ , under constant power (see Section 3.3 for how the power influences the modes). Zhu et al [3] did similar *without* the power constraint, and we will compare our results to theirs in Section 6.3. The squirmer is rewarded for having a low distance  $r$  at that time step to the target, in a similar fashion to Eq. (2.9) in [3]:

$$R_t = \begin{cases} -1000, & r > r_0, \quad (\text{End episode}) \\ 200/(t - d_0/v_{\text{charac}}), & r < r_{\text{catch}}, \quad (\text{End episode}) \\ 3000, & r < r_{\text{catch}} \wedge t \leq d_0, \quad (\text{End episode}) \\ -r, & \text{otherwise} \end{cases} \quad (72)$$

Where  $r_0$  is the initial centre-centre distance,  $r_{\text{catch}} = a + \delta$  is the squirmer radius plus an additional small catch distance,  $d_0 = r_0 - r_{\text{catch}}$  is the initial distance between the squirmer surface and the target, and the characteristic velocity  $v_{\text{charac}} = 1$ . By defining the reward as such, the agent will quickly learn not to move away from the target, as it will be massively penalized with  $-1000$ . The remaining three other options encourage the agent to quickly catch the target, by either penalizing for being far away with  $-r$  or rewarding a quick catch with  $200/(t - d_0/v_{\text{charac}})$ . The  $r < r_{\text{catch}} \wedge t \leq d_0$  condition seems at first to be redundant, as  $t$  cannot be less than the time required to travel the initial shortest distance  $d_0$ , but when the power consumption is not restricted, both the horizontal and vertical velocity may be 1 simultaneously, yielding a total velocity of  $\sqrt{2}$ . If

one does not include this condition in the reward function, the agent will learn a time-inefficient strategy to avoid  $t - d_0/v_{\text{charac}}$  becoming negative. Zhu et al [3] did not include this condition. It should be noted that this scenario is already solved analytically with Eq. (53) - (55), so RL is not strictly necessary.

### 4.3 Imperfectly sensing a finite target

Moving on from the simple case, we now investigate how the squirmer finds a finite size target by sensing disturbances in its force field. The target is a passive spherical particle, that now disturbs the flow field created by the squirmer. In each training session, the target is kept at a constant angle and distance to the squirmer and only one time step is taken per episode. Unlike in the previous section, the flow field must now be numerically solved, which we do by using the boundary element method explained in section 3.1. The force disturbances are found by calculating the difference in the force field on the surface of the squirmer with and without the target and then averaging the differences to produce a single vector, which in the ideal case points directly towards the target. To do so, a weighted mean of the force magnitudes and their positions is calculated. The weight for the  $k$ 'th point on the surface of the squirmer is:

$$w^k = \sqrt{(\Delta F_x^k + \eta_x^k)^2 + \Delta(F_y^k + \eta_y^k)^2 + \Delta(F_z^k + \eta_z^k)^2} \quad (73)$$

Where  $\Delta F^k$  is the  $k$ 'th point's difference in the squirmer's force field with and without the target, the sensor noise is Gaussian  $\eta^k \sim \mathcal{N}(\mu = 0, \sigma = \sigma_{\text{noise}})$  and  $\sigma_{\text{noise}}$  is the noise strength hyperparameter. Let  $\mathbf{x}^k$  be the vector pointing from the centre of the squirmer to the  $k$ 'th point, and then the normalized observed vector is:

$$\mathbf{O} = \sum_{k=1}^N w^k \mathbf{x}^k \quad (74)$$

$$\Rightarrow \hat{\mathbf{O}} = \frac{\mathbf{O}}{\|\mathbf{O}\|} \quad (75)$$

And thus the agent state for the imperfect sensing squirmer is  $\mathcal{S} = \{\hat{O}_x, \hat{O}_y, \hat{O}_z\}$ . In Figure 4b the vector  $\hat{\mathbf{O}}$  has been plotted for different target sizes and modes. To combat the sensing noise  $\eta$  introduced in Eq. (73), the agent is given access to more modes, in addition to the three translating modes. We choose only to include up to  $n = 2, m = 2$  modes, so  $\mathcal{A} = \{B_{01}, B_{11}, \dots, \tilde{C}_{22}\}$  because of computation time. When sensing, the reward function is based on the normalized difference between the real and guessed angle:

$$R_t = 1 - \frac{|\theta - \theta_{\text{guess}}|_{\min}}{\pi} \quad (76)$$

Where  $|\theta - \theta_{\text{guess}}|_{\min}$  is the minimum angle difference, defined as:

$$|x - y|_{\min} = \min\{|x - y|, |x - y - 2\pi|, |x - y + 2\pi|\} \quad (77)$$

With a finite reward space  $\mathcal{R} = [0, 1]$ , the reward has a clear interpretation, with  $R_t = 1$  being a perfectly guessed angle and  $R_t \leq 0.5$  being as good as or worse than a random guess. This means we have a well defined upper limit for when the training becomes too hard for the agent, namely once the mean reward becomes  $R = 0.5$ . The upper limit in most of the graphs in the results section (5) are found this way.

Hyperparameter	Value
Surface Points $N$	700
Squirmer Radius $a$	1
Viscosity $\eta$	1
Max mode value	1
Regularization offset $\epsilon$	0.05
Extra catch distance $\delta$	0.1

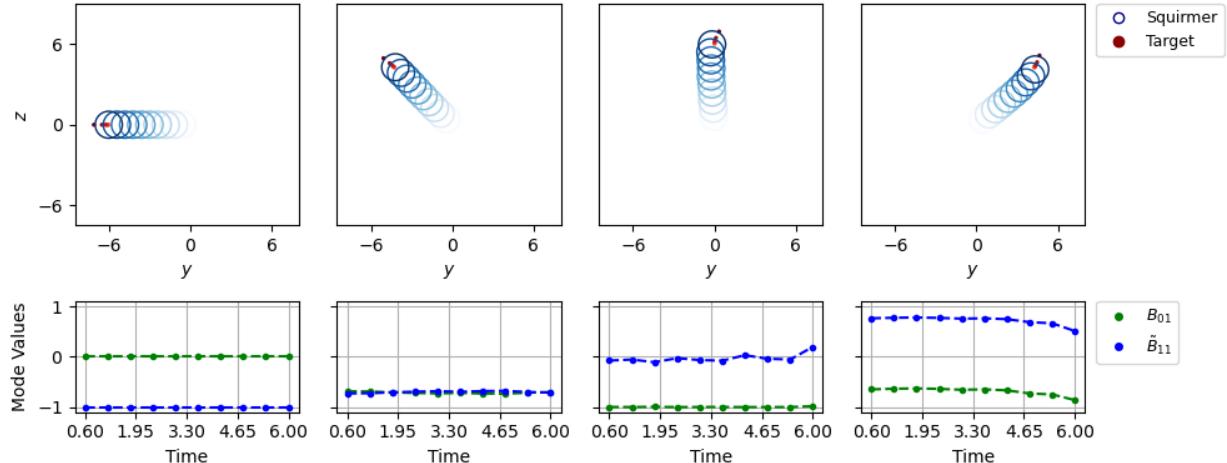
Policy	PPO [14]
Gym version	0.26.2
SB3 version	1.7.0

**Table 1:** Hyperparameters, policy and library version used in RL.

## 5 Results

### 5.1 Perfect sensing and constant power

We first consider the case where the target is a point mass moving with the flow produced by the squirmer. This means the target does not disturb the velocity field produced by the squirmer, and we can use the analytical solution of the velocity field given in Eq. (53) - (55). The squirmer has access to two modes:  $B_{01}$  and  $\tilde{B}_{11}$ , which are responsible for the squirmer moving along the  $\hat{\mathbf{z}}$  and  $\hat{\mathbf{y}}$ -direction, respectively (See Figure 2 for the flow field). Furthermore, we require that the squirmer has a constant power consumption of  $P_{const} = \frac{64\pi\mu}{3a^5}$ , as described in section (3.3). The



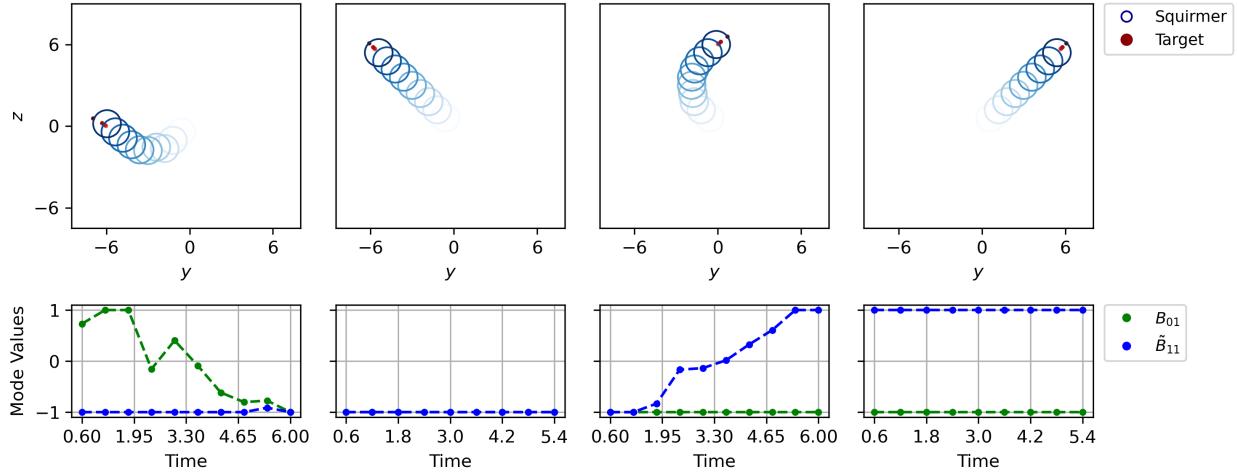
**Figure 7:** The first row shows the path taken by the squirmer and target. The colour of the markers indicates time, with darker values being later. The second row has the  $B_{01}$  and  $\tilde{B}_{11}$  values at each time step. The optimal trajectory is a straight line when the power consumption is constant  $P_{const} = \frac{64\pi\mu}{3a^5}$ .

plot in Figure 7 shows that the optimal strategy for the squirmer is to move in a straight line, independent of its initial angle  $\theta_0$ . The markers are coloured such that darker indicates later in time. Small irregularities to the straight lines can appear when the squirmer gets close enough that its velocity field moves the target significantly.

In all cases, the RL agent was trained until the reward converged, and this convergence was qualitatively judged from reward vs. training steps graphs.

## 5.2 Perfect sensing and no power limit

If we do not limit the squirmer's power consumption but only the maximum mode value, we then have the same setup as in Zhu et al [3]. In this case, the squirmer moves in an "L" shape when the target does not start in one of the corners, meaning the initial angle  $\theta_0 \notin \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$ . In Figure 8 we have plotted the path taken by the squirmer and target for different initial angles  $\theta_0$ . Figure 8 also shows the sequence of mode choices over time for each case. A discussion of why the squirmer moves in an "L" shape can be found in section 6.3. Sometimes, instead of an "L"-shape,



**Figure 8:** The first row shows the path taken by the squirmer and target. The colour of the markers indicates time, with darker values being later. The second row has the  $B_{01}$  and  $\tilde{B}_{11}$  values at each time step. The squirmer moves in an "L" shape for  $\theta_0 \notin \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$  if we do not require the power consumption to be constant.

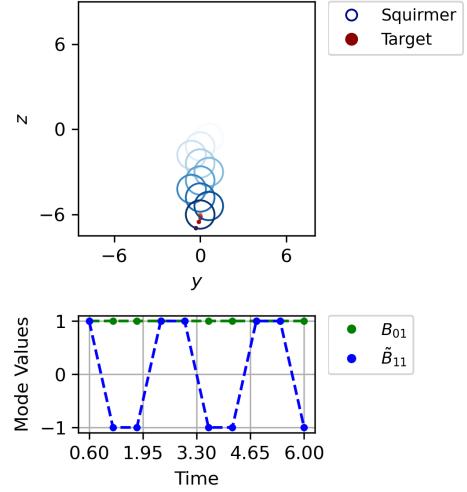
the agent learns a zigzag movement, frequently changing the sign of one of its modes. An example of this can be seen in Figure 9.

## 5.3 Imperfectly sensing a finite target

We investigate which modes are optimal for sensing the target's direction when the squirmer has a constant power consumption of  $P_{const} = \frac{64\pi\mu}{3a^5}$  and has access to all modes up to and including  $n = 2, m = 2$ . This means it has access to the 13 lowest modes. We have not included higher-order modes primarily because they would be more computationally heavy but also because higher-order modes cost more energy and have faster-decaying velocity fields. By activating different modes, the squirmer can then predict the direction towards the target. It does effectively not move, as only one time step is taken per episode.

In the following graphs, each data point is found by averaging 30 episodes from the same training data. In other words, the squirmer makes 30 predictions based on the same experience, and the average of those guesses is one data point. Ideally, each data point would also be averaged over points from different training sessions, as even though the RL agent received a satisfactory reward, there is still a possibility it had learned an unusual behaviour that multiple training sessions would reveal. This was not done because of the huge computation time it would demand. Furthermore, we choose to omit the modes that never have any significant activation in the graphs.

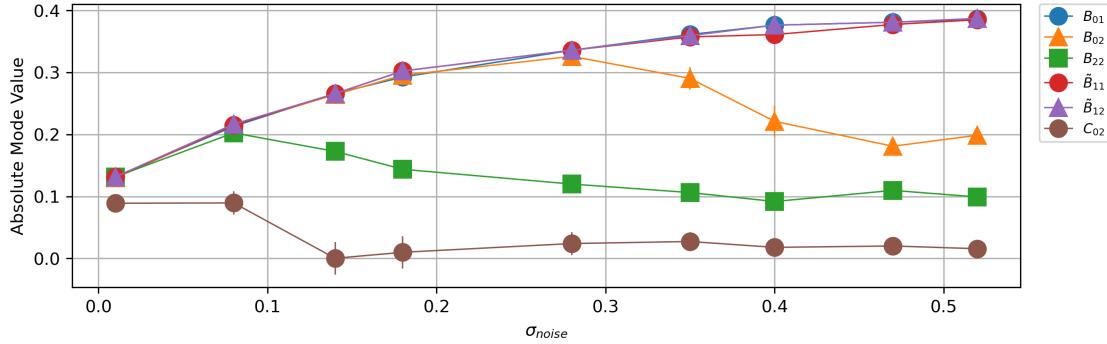
The following parameters are varied: sensor noise  $\sigma_{noise}$ , centre-centre distance  $r_0$ , target radius



**Figure 9:** Example of the zigzag motion utilizing diagonal velocity is  $\sqrt{2}$ .

$a_{\text{target}}$  and finally the angle in the  $xy$ -,  $xz$ - and  $yz$ -plane.

In Figure 10 **Sensor noise**  $\sigma_{\text{noise}}$  is varied and the other parameters are set to  $\phi_0 = \pi/2$ ,  $\theta_0 = \pi/4$ ,  $a = 1$ ,  $a_{\text{target}} = 0.4$ ,  $r_0 = 2$ . When the sensor noise is small it does not matter which modes are activated, as all modes are efficient at detecting. As  $\sigma_{\text{noise}}$  increases the bad modes are filtered out, and the best modes for sensing are  $B_{01}, \tilde{B}_{11}, \tilde{B}_{12}$ . Mode  $B_{01}$  and  $\tilde{B}_{11}$  are the translational modes in the  $\hat{\mathbf{z}}$  and  $\hat{\mathbf{y}}$  direction respectively, and thus translation is at least as efficient at sensing as other modes for all  $\sigma_{\text{noise}}$ .

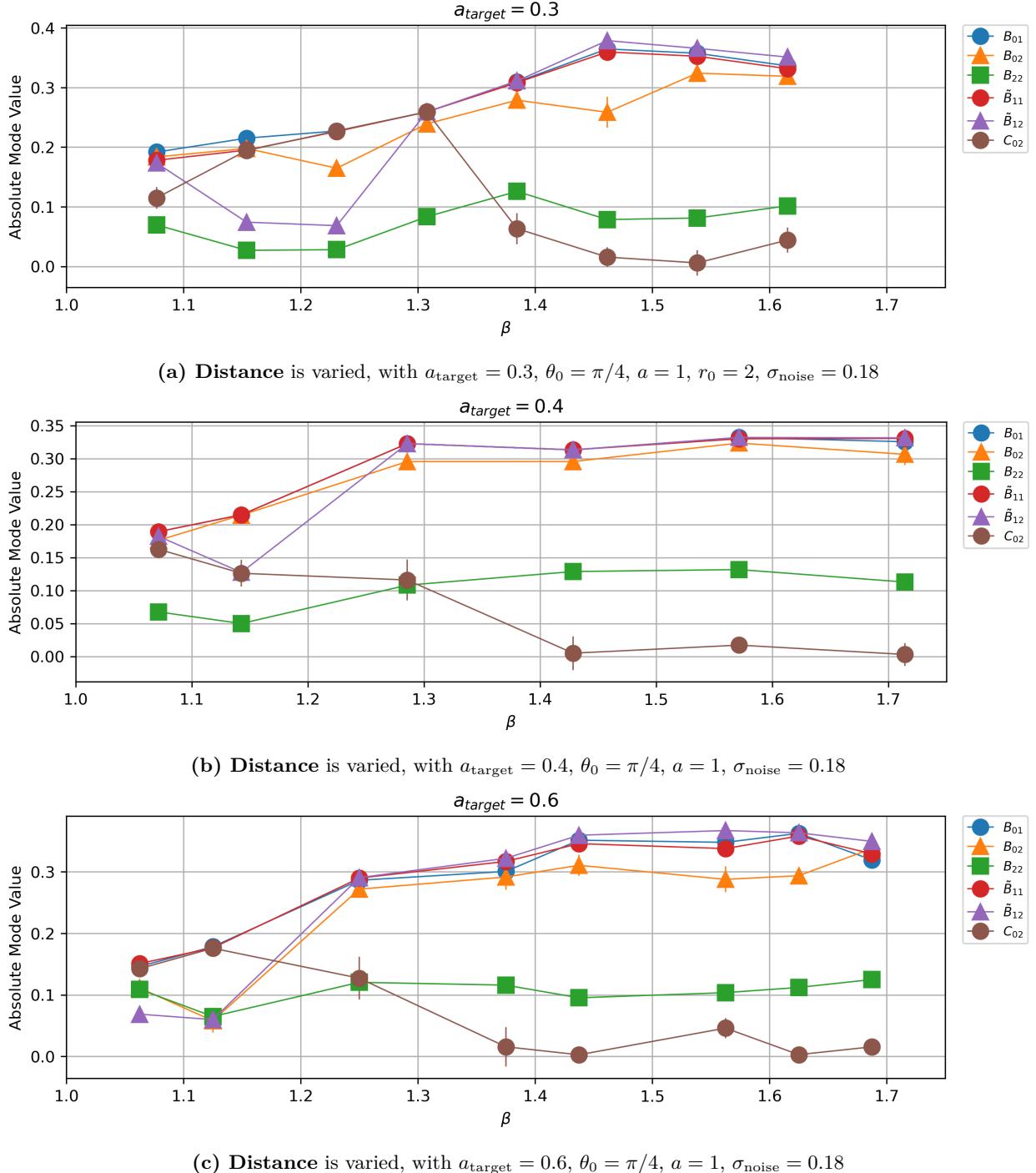


**Figure 10:** **Sensor noise** is varied. The other parameters are  $\phi_0 = \pi/2$ ,  $\theta_0 = \pi/4$ ,  $a = 1$ ,  $a_{\text{target}} = 0.4$ ,  $r_0 = 2$ . At low sensor noise, it does not matter which modes are activated. At high sensor noise modes  $B_{01}, \tilde{B}_{11}, \tilde{B}_{12}$  are best at sensing. Modes with no significant activation are omitted.

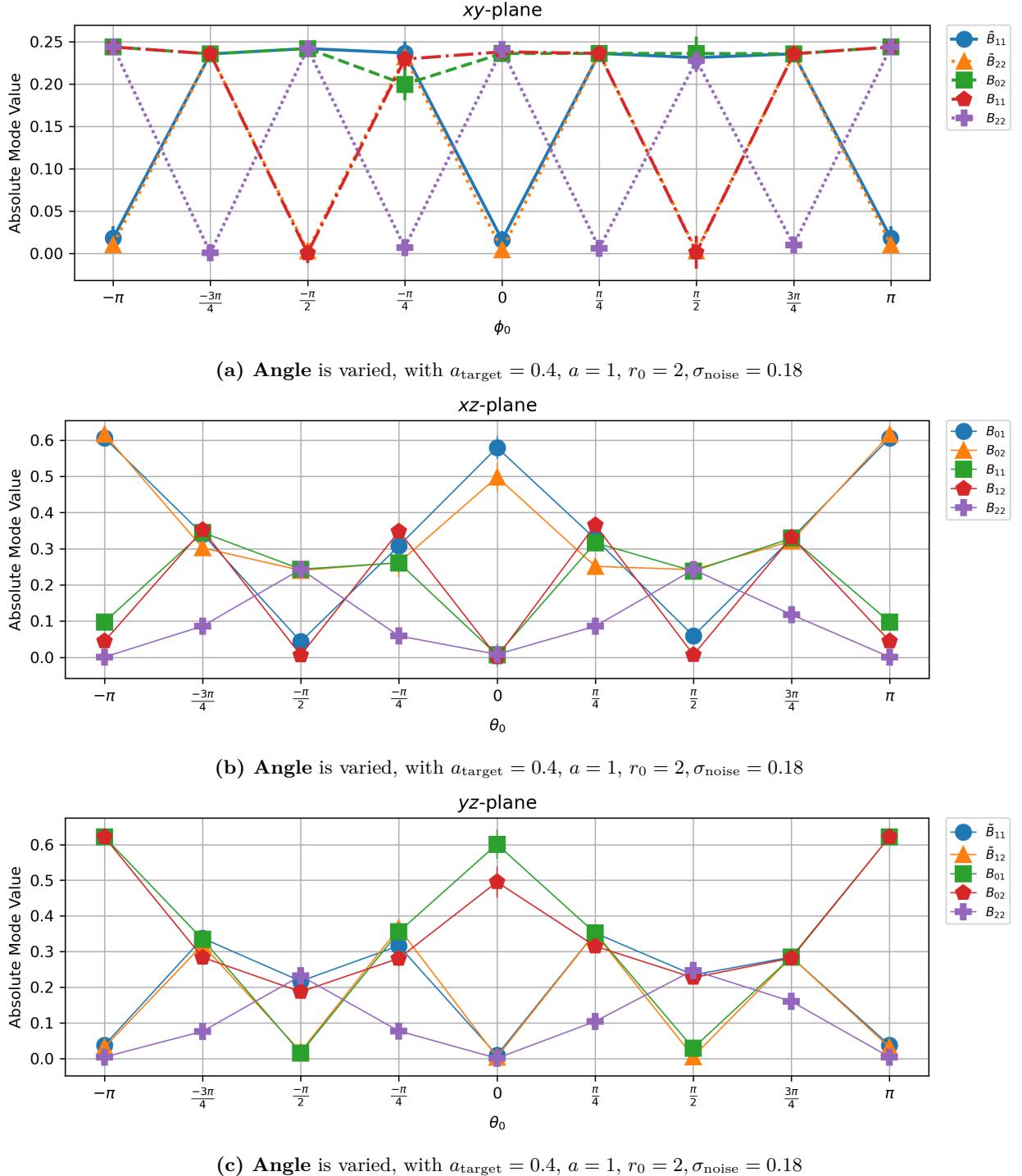
In Figure 11, we have varied the **centre-centre distance**,  $r_0$ . We have plotted the mode choice for three different  $a_{\text{target}}$  values as a function of  $\beta \equiv r_0/(a + a_{\text{target}})$ . This is done, so it is easier to compare the graphs for different  $a_{\text{target}}$ . When  $\beta = 1$  the target touches the surface of the squirmer. In each of the three plots, we see a similar picture as in the noise graph (Figure 10): When the target is close to the squirmer, it is not important which modes are activated, and as  $\beta$  increases, it becomes increasingly difficult to sense the target, and a noticeable preference for some modes over others is seen. In all three cases, the best modes for sensing are  $B_{01}, B_{02}, \tilde{B}_{11}, \tilde{B}_{12}$ , so the translational modes  $B_{01}$  and  $\tilde{B}_{11}$  are at least as efficient as other sensing modes.

Lastly, we investigate which modes are used for different **angles**  $\phi_0, \theta_0$ . For visual aid, all angle graphs have had the  $\pi$  data points also plotted at  $-\pi$ . From our previous results in Figure 10 and 11, we have chosen  $\sigma_{\text{noise}} = 0.18$ ,  $a_{\text{target}} = 0.4$ ,  $r_0 = 2$  for the angle training. These are chosen because they give a *stable mode choice*. We consider a mode choice stable when there is a clear preference for some modes over others. In the  $xy$ -plane,  $\theta_0 = \pi/2$ , and we vary  $\phi_0$ . This can be seen in Figure 12a, where the two translational modes  $B_{11}$  and  $\tilde{B}_{11}$  perform similarly to all other modes for all angles except those where its direction of travelling is perpendicular to the target. The  $B_{02}$  mode is doing exceptionally well here. We notice that the behaviour of the  $xy$ -plane graph is rather different from the two other planes. We suspect this is because we choose modes up and to  $n = 2, m = 2$ . This yields 13 modes, which means some directions have fewer available modes than others. However, we still find that the translational modes are efficient. The  $xz$ -plane is seen in Figure 12b. Here  $\phi_0 = 0$  and we vary  $\theta_0$ . Again it shows that, for all angles, the translational modes are, within uncertainties, at least as efficient as the non-translation modes. When the target is along the  $\hat{\mathbf{z}}$ -axis the translational mode  $B_{01}$  is best at sensing and likewise  $B_{11}$  is the best at sensing when the target is along the  $\hat{\mathbf{x}}$ -axis. Finally, the  $yz$ -plane is seen in Figure 12c, and repeats the pattern of the  $xz$ -plane. Here  $\phi_0 = \pi/2$  and  $\theta_0$  is varied. When the target is along the  $\hat{\mathbf{y}}$ -axis the translational mode  $\tilde{B}_{01}$  is best at sensing and as before  $B_{01}$  is the best at sensing when the target is along the  $\hat{\mathbf{z}}$ -axis.

We thus find that in each coordinate plane, the translational modes in that plane are at least as efficient at sensing as other modes. Moreover, we find that the  $B_{02}$  mode is one of the best modes for all  $\theta_0, \phi_0$ .  $B_{02}$  is the slowest decaying mode. All other modes, including the translational modes, decay faster.



**Figure 11:** Best modes for sensing the target's direction when varying  $r_0$  in  $\beta \equiv r_0/(a + a_{\text{target}})$  for three different  $a_{\text{target}}$  values.



**Figure 12:** Best modes for sensing the target's direction when the initial angle, either  $\theta_0$  or  $\phi_0$ , is varied, in the  $xy$ -,  $xz$ -, and  $yz$ -plane.

## 6 Discussion

### 6.1 Idealizations

We are aware that we have made many simplifications throughout this thesis. In this section, we go through them, and the impact they have on our results.

### 6.1.1 Constant force and Oseen tensor

In the velocity expression used for the BEM (56), we have assumed that both the force  $\mathbf{F}$  and the Oseen tensor  $\mathbf{S}$  are constant over the boundary area element,  $\Delta A$ . We have also assumed that all area elements are equal:  $\Delta A_1 = \dots = \Delta A_k = \dots \Delta A_N$ . According to Smith [10], assuming the force is constant over the area elements  $\Delta A$  is a reasonable approximation since the forces vary slowly, meaning we made the simplification that:

$$\int_{\Delta A_k} \mathbf{S}(k, q) dA_k \cdot \mathbf{F} \approx \mathbf{S}(k, q) \Delta A \cdot \mathbf{F} \quad (78)$$

However, assuming  $\mathbf{S}$  is constant over the boundary area element  $\Delta A$  is *not* a good approximation, as  $\mathbf{S}$  varies rapidly near  $\epsilon$ , and it is particularly not good when  $k = q$  [10]. This would explain the need for such a high number of surface point  $N$ , as evident in Figure 5. A larger  $N$  means smaller and more evenly distributed boundary elements and only for very large  $N$  will Eq. (78) become a good approximation. The better way is to numerically evaluate the integral on the left-hand side of Eq. (78) for each boundary element.

### 6.1.2 One object power equation

The power given in Eq. (10) is derived for one object, however, we have used this equation to limit the squirmer's modes in the RL scenarios with two objects. In [16], Lauga et al. show that the squirmer's power is changed if it is placed inside a tube, and it is, therefore, expectable that placing a target next to the squirmer also could affect the power. It is not an issue, however, if the power for all modes is affected equally. It is possible to find and numerically solve the power equation for two objects, at the cost of increasing the computation time.

## 6.2 Further improvements

Apart from the idealizations mentioned in the previous section, one could combine our results for optimal modes in imperfect sensing with catching a finite target.

### 6.2.1 Catching a finite target with imperfect sensing

In sections (4.2) and (4.3), the RL squirmer was first given the target's precise position to catch it, and in the next section, it had to guess their relative angle but did not catch the target. Combining these two scenarios such that the squirmer would need to guess the target's position and catch it would be a proper next step. It should be noted that the two previous examples can be done *without* RL, but when combined they require RL. The reward would be similar to when catching the point particle, so penalize the squirmer from being far away, and reward it for catching the target. The actions could in principle be the same as when sensing the target, that is all modes  $B_{01}, B_{11}, \dots, \tilde{C}_{nn}$  up to any  $n$ , but as we saw in section (5.3), many of the modes are never used for sensing and ought not to be included. The state is what would need the greatest change. For sensing to be usable, the agent would need to know what it sensed in (at least) the last time step. We propose that in addition to the average force difference vector  $\hat{\mathbf{O}}$  defined in Eq. (75) the agent would be given the angle it sensed last time step  $\theta_{t-1}$ , along with how good the guessed angle was  $R(\theta_{t-1})$ . The quality of the guessed angle would be the normalized difference between the real and guessed angle, so identical to the reward in the imperfect sensing scenario in Eq. (76). The previous time step's angle would need to be updated to match the agent's new position. The agent state is then:

$$\mathcal{S} = \{\hat{O}_x, \hat{O}_y, \hat{O}_z, \theta_{t-1}, R(\theta_{t-1})\} \quad (79)$$

With our results that even for high noise and distance between squirmer and target, the translational modes remained efficient, we predict that not much would change in the squirmer's catching strategy

whether it could perfectly sense a point target or imperfectly sense a finite target. The optimal strategy would most likely still be to move in a straight, albeit now probably jagged, line.

### 6.3 Comparing our results to the Zhu 2022 article

In section 5.2 we found that the squirmer moves in an "L" shape if we do not restrict its power consumption and the initial angle between squirmer and target  $\theta_0 \notin \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$ . This is also what Zhu et al., 2022 [3] found. They proposed that the reason for the squirmer moving in an "L" shape when  $\theta_0 \notin \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$  is due to the squirmer pushing the target away as it approaches the target. By moving in an "L" shape the squirmer will not push the target away: "[a] predator swimming straight toward the prey generates a flow field that always repels the prey away from it [...] the predator adjusts its position (with respect to the prey) and surface actuation for best exploiting its disturbance flow field to attract the prey, leading to the initially left-upward movement." [3]. We however disagree with their explanation. We propose that the reason the squirmer moves in an "L" shape when there is no restriction on its power consumption and  $\theta_0 \notin \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$ , is because it can maximize its speed by moving diagonally. This is also why the squirmer does not move in an "L" shape when its initial angle places the target in a corner:  $\theta_0 = \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$ , as the shortest distance here is the diagonal distance. When we require that the power consumption is constant, the squirmer does not move in an "L" shape for all  $\theta_0$ , as seen in Figure 7. Furthermore, the swimming strategy should not depend on the coordinate system. If swimming in an "L"-shape were the most optimal strategy, it should do so for all  $\theta_0$ .

## 7 Conclusion

By first deriving the squirmer's swimming kinematics, we find that only three modes are responsible for translation. Then introducing the boundary element method, we have shown using reinforcement learning that the translation modes are at least as efficient at detecting the target as other modes, in other words: *optimal sensing* is optimal swimming. More explicitly, this was found by varying the sensor noise, the target radius, the centre-centre distance and the angle between the squirmer and the target. The angle investigations showed that for all planes, no matter the angle, at least one mode is simultaneously efficient at translation and sensing. Furthermore, the  $B_{02}$  mode was found to be efficient at sensing for all angles. From the other parameter investigations, it was similarly found that translation towards the target was at least as efficient at sensing as the other modes, and we conclude that in the stable mode regions, it will always be optimal to pick the translational mode that moves the squirmer in the direction of the target.

We have also argued that limiting the squirmer's power is crucial for getting realistic behaviour. If not, the squirmer will learn to take either an "L"-shaped or zigzag path if the initial angle is not diagonal.

## Acknowledgement

We would like to extend our deepest thanks to Julius Bier Kirkegaard, in whose office many enjoyable, fruitful hours have been spent from the very start of the project to its end. Julius was able to shape our wild ideas into something useful while still keeping the elements we found interesting.

## References

1. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).
2. Colabrese, S., Gustavsson, K., Celani, A. & Biferale, L. Flow navigation by smart microswimmers via reinforcement learning. *Physical review letters* **118**, 158004 (2017).

3. Zhu, G., Fang, W.-Z. & Zhu, L. Optimizing low-Reynolds-number predation via optimal control and reinforcement learning. *Journal of Fluid Mechanics* **944**, A3 (2022).
4. Lauga, E. *The Fluid Dynamics of Cell Motility* (Cambridge University Press, 2020).
5. Happel, J. & Brenner, H. *Low Reynolds Number Hydrodynamics. with special applications to particulate media* (Martinus Nijhoff Publishers, 1973).
6. Lamb, H. *Hydrodynamics* Fourth (Cambridge University Press, 1916).
7. Pak, O. S. & Lauga, E. Generalized squirming motion of a sphere. *Journal of Engineering Mathematics* **88**, 1–28. <https://doi.org/10.1007/s10665-014-9690-9> (2014).
8. Pijush Kundu, I. C. & Dowling, D. *Fluid Mechanics* Sixth (Elsevier Academic Press, 2016).
9. Cortez, R. The Method of Regularized Stokeslets. *Siam Journal on Scientific Computing* **23** (Dec. 2001).
10. Smith, D. J. A boundary element regularized Stokeslet method applied to cilia-and flagella-driven flow. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **465**, 3605–3626 (2009).
11. Roberts, M. *How to evenly distribute points on a sphere more effectively than the canonical Fibonacci Lattice* <http://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/> (2023).
12. Brockman, G. *et al.* OpenAI Gym 2016. eprint: [arXiv:1606.01540](https://arxiv.org/abs/1606.01540).
13. Raffin, A. *et al.* Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research* **22**, 1–8. <http://jmlr.org/papers/v22/20-1364.html> (2021).
14. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal Policy Optimization Algorithms. <https://arxiv.org/abs/1707.06347> (2017).
15. Steenbrugge, X. *An introduction to Policy Gradient methods - Deep Reinforcement Learning* <https://youtu.be/5P7I-xPq8u8> (2023).
16. Zhu, L., Lauga, E. & Brandt, L. Low-Reynolds-number swimming in a capillary tube. *Journal of Fluid Mechanics* **726**, 285–311 (2013).

## A Appendix

### A.1 Derivation of derivatives of spherical harmonics

**SH 1** *Radial derivative of SH*

$$\frac{\partial}{\partial r}(\Gamma_{-n-1}) = (-n-1)r^{-n-2} \sum_{m=0}^n P_n^m [K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi] \quad (80)$$

$$= \frac{-n-1}{r} r^{-n-1} \sum_{m=0}^n P_n^m [K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi] \quad (81)$$

$$= -\frac{n+1}{r} \cdot \Gamma_{-n-1} \quad (82)$$

**SH 2** *Radial derivative of gradient of SH*

For a function  $f$  (not necessarily a spherical harmonic), the radial derivative of the gradient of  $f$  in spherical coordinates is:

$$\frac{\partial}{\partial r} \nabla f = \frac{\partial}{\partial r} \left( \frac{\partial f}{\partial r} \hat{\mathbf{r}} + \frac{1}{r} \frac{\partial f}{\partial \theta} \hat{\boldsymbol{\theta}} + \frac{1}{r \sin \theta} \frac{\partial f}{\partial \phi} \hat{\boldsymbol{\phi}} \right) \quad (83)$$

$$= \frac{\partial^2 f}{\partial r^2} \hat{\mathbf{r}} - \frac{1}{r^2} \frac{\partial f}{\partial \theta} \hat{\boldsymbol{\theta}} + \frac{1}{r} \frac{\partial^2 f}{\partial r \partial \theta} \hat{\boldsymbol{\theta}} - \frac{1}{r^2 \sin \theta} \frac{\partial f}{\partial \phi} \hat{\boldsymbol{\phi}} + \frac{1}{r \sin \theta} \frac{\partial^2 f}{\partial \phi \partial r} \hat{\boldsymbol{\phi}} \quad (84)$$

$$= \left( \frac{\partial^2 f}{\partial r^2} \hat{\mathbf{r}} + \frac{1}{r} \frac{\partial^2 f}{\partial r \partial \theta} \hat{\boldsymbol{\theta}} + \frac{1}{r \sin \theta} \frac{\partial^2 f}{\partial \phi \partial r} \hat{\boldsymbol{\phi}} \right) - \frac{1}{r} \left( \frac{1}{r} \frac{\partial f}{\partial \theta} \hat{\boldsymbol{\theta}} + \frac{1}{r \sin \theta} \frac{\partial f}{\partial \phi} \hat{\boldsymbol{\phi}} \right) \quad (85)$$

The term in the first parenthesis is recognized as  $\nabla \frac{\partial f}{\partial r}$ , and the term inside the second as  $\nabla f - \frac{\partial f}{\partial r} \hat{\mathbf{r}}$ , so:

$$\frac{\partial}{\partial r} \nabla f = \nabla \frac{\partial f}{\partial r} - \frac{1}{r} (\nabla f - \frac{\partial f}{\partial r} \hat{\mathbf{r}}) \quad (86)$$

Using this result with  $\Gamma_{-n-1}$  and applying SH 1:

$$\frac{\partial}{\partial r} (\nabla \Gamma_{-n-1}) = \nabla \left( -\frac{n+1}{r} \Gamma_{-n-1} \right) - \frac{1}{r} \left( \nabla \Gamma_{-n-1} - \frac{-(n+1)}{r} \Gamma_{-n-1} \hat{\mathbf{r}} \right) \quad (87)$$

$$= -(n+1) \left( -\frac{1}{r^2} \Gamma_{-n-1} \hat{\mathbf{r}} + \frac{1}{r} \nabla (\Gamma_{-n-1}) \right) - \frac{1}{r} \nabla \Gamma_{-n-1} - \frac{(n+1)}{r^2} \Gamma_{-n-1} \hat{\mathbf{r}} \quad (88)$$

$$= -\frac{n+1}{r} \nabla \Gamma_{-n-1} - \frac{1}{r} \nabla \Gamma_{-n-1} \quad (89)$$

$$= -\frac{n+2}{r} \nabla \Gamma_{-n-1} \quad (90)$$

**SH 3** *Radial derivative of curl of SH multiplied by radial vector*

Let  $\mathbf{A} = \mathbf{r} \Gamma_{-n-1}$ . As  $\mathbf{A}$  only has a radial component, the curl reduces to:

$$\nabla \times \mathbf{A} = \frac{1}{r} \left( \frac{1}{\sin \theta} \frac{\partial A_r}{\partial \phi} \hat{\boldsymbol{\theta}} - \frac{\partial A_r}{\partial \theta} \hat{\boldsymbol{\phi}} \right) \quad (91)$$

The radial derivative of this is:

$$\frac{\partial}{\partial r} (\nabla \times \mathbf{A}) = -\frac{1}{r^2} \left( \frac{1}{\sin \theta} \frac{\partial A_r}{\partial \phi} \hat{\boldsymbol{\theta}} - \frac{\partial A_r}{\partial \theta} \hat{\boldsymbol{\phi}} \right) + \frac{1}{r} \left( \frac{1}{\sin \theta} \frac{\partial^2 A_r}{\partial \phi \partial r} \hat{\boldsymbol{\theta}} - \frac{\partial^2 A_r}{\partial \theta \partial r} \hat{\boldsymbol{\phi}} \right) \quad (92)$$

$$= -\frac{1}{r} \nabla \times \mathbf{A} + \nabla \times \frac{\partial \mathbf{A}}{\partial r} \quad (93)$$

Now inserting  $\mathbf{A} = \mathbf{r}\Gamma_{-n-1}$  and using SH1:

$$\frac{\partial}{\partial r}(\nabla \times (\mathbf{r}\Gamma_{-n-1})) = -\frac{1}{r}\nabla \times (\mathbf{r}\Gamma_{-n-1}) + \nabla \times \left(\Gamma_{-n-1}\frac{\partial \mathbf{r}}{\partial r} + \mathbf{r}\frac{\partial \Gamma_{-n-1}}{\partial r}\right) \quad (94)$$

$$= -\frac{1}{r}\nabla \times (\mathbf{r}\Gamma_{-n-1}) + \nabla \times \left(\Gamma_{-n-1}\frac{\mathbf{r}}{r} - \mathbf{r}\frac{n+1}{r}\Gamma_{-n-1}\right) \quad (95)$$

$$= -\frac{1}{r}\nabla \times (\mathbf{r}\Gamma_{-n-1}) - \frac{n}{r}\nabla \times (\mathbf{r}\Gamma_{-n-1}) \quad (96)$$

$$= -\frac{n+1}{r}\nabla \times (\mathbf{r}\Gamma_{-n-1}) \quad (97)$$

Where it was used that the curl of  $\mathbf{r}\Gamma_{-n-1}$  is constant over  $\mathbf{r}$ .

## A.2 Derivation of cross product of spherical harmonics

### SH 4 Cross product rule 1

$$\mathbf{r} \times (\nabla \times (\mathbf{r}\Gamma_{-n-1})) = \mathbf{r} \times \left(\frac{1}{r} \frac{1}{\sin \theta} \frac{\partial}{\partial \phi} (r\Gamma_{-n-1}) \hat{\theta} - \frac{1}{r} \frac{\partial}{\partial \theta} (r\Gamma_{-n-1}) \hat{\phi}\right) \quad (98)$$

$$= \frac{r}{\sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \hat{\phi} + r \frac{\partial \Gamma_{-n-1}}{\partial \theta} \hat{\theta} \quad (99)$$

$$= \frac{r}{\sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \hat{\phi} + r \frac{\partial \Gamma_{-n-1}}{\partial \theta} \hat{\theta} - (n+1)\mathbf{r}\Gamma_{-n-1} + (n+1)\mathbf{r}\Gamma_{-n-1} \quad (100)$$

$$= \underbrace{\frac{r}{\sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \hat{\phi} + r \frac{\partial \Gamma_{-n-1}}{\partial \theta} \hat{\theta}}_{=r^2 \nabla \Gamma_{-n-1}} + r^2 \frac{\partial \Gamma_{-n-1}}{\partial r} \hat{\mathbf{r}} + (n+1)\mathbf{r}\Gamma_{-n-1} \quad (101)$$

$$= r^2 \nabla \Gamma_{-n-1} + (n+1)\mathbf{r}\Gamma_{-n-1} \quad (102)$$

### SH 5 Cross product rule 2

$$\mathbf{r} \times \nabla \Gamma_{-n-1} = \mathbf{r} \times \left(\frac{\partial \Gamma_{-n-1}}{\partial r} \hat{\mathbf{r}} + \frac{1}{r} \frac{\partial \Gamma_{-n-1}}{\partial \theta} \hat{\theta} + \frac{1}{r \sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \hat{\phi}\right) \quad (103)$$

$$= \frac{\partial \Gamma_{-n-1}}{\partial \theta} \hat{\phi} - \frac{1}{\sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \hat{\theta} \quad (104)$$

$$= -\nabla \times (\mathbf{r}\Gamma_{-n-1}) \quad (105)$$

## A.3 Derivation of surface integral theorems

### A.3.1 Evaluating $\iint_S \Gamma_{-n-1} dS$

Let  $\Gamma_{-n-1}$  be an arbitrary solid spherical harmonics defined in equation (19) with  $0 < n$  and  $0 \leq m \leq n$ . Then:

$$\iint_S \Gamma_{-n-1} dS = \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \Gamma_{-n-1} a^2 \sin \theta d\theta d\phi \quad (106)$$

$$= a^{-n+1} \int_0^\pi P_n^m \sin \theta d\theta \underbrace{\int_0^{2\pi} (K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi) d\phi}_{\begin{aligned} &= 2\pi K_{0n} \text{ for } m=0 \\ &= 0 \text{ otherwise} \end{aligned}} \quad (107)$$

From the  $\phi$  integral, we can conclude that  $m$  must be 0 in order to have a nonzero integral. The  $\theta$  integral can be evaluated using the orthogonality relationship:

$$\int_0^\pi P_n^m P_{n'}^m \sin \theta d\theta = \frac{2(n+m)!}{(2n+1)(n-m)!} \delta_{nn'} \quad (108)$$

$\delta_{nn'}$  is the kronecker delta function. In our case we use  $1 = P_0^0$ , so:

$$\iint_S \Gamma_{-n-1} dS = 2\pi K_{0n} \cdot a^{-n+1} \underbrace{\int_0^\pi P_0^0 \cdot P_n^0 \sin \theta d\theta}_{\begin{array}{l} = 2 \text{ for } n=0 \\ = 0 \text{ otherwise} \end{array}} \quad (109)$$

We see from the  $\theta$  integral that all  $n > 0$  gives 0. Hence:

$$\iint_S \Gamma_{-n-1} dS = 0, \text{ for all } n > 0 \text{ and } n \geq m \geq 0. \quad (110)$$

### A.3.2 Evaluating $\iint_S a\hat{\mathbf{r}}\Gamma_{-n-1} dS$

For the integral  $\iint_S a\hat{\mathbf{r}}\Gamma_{-n-1} dS$  it is useful to write the vector  $a\hat{\mathbf{r}}$  in Cartesian coordinates as

$$\begin{pmatrix} a \sin \theta \cos \phi \\ a \sin \theta \sin \phi \\ a \cos \theta \end{pmatrix}:$$

$$\iint_S a\hat{\mathbf{r}}\Gamma_{-n-1} dS = \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \begin{pmatrix} a \sin \theta \cos \phi \\ a \sin \theta \sin \phi \\ a \cos \theta \end{pmatrix} \Gamma_{-n-1} a^2 \sin \theta d\theta d\phi \quad (111)$$

$$\begin{aligned} &= a^{-n+2} \int_0^\pi P_n^m \sin \theta \begin{pmatrix} \sin \theta \\ \sin \theta \\ \cos \theta \end{pmatrix} d\theta \underbrace{\int_0^{2\pi} (K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi) \begin{pmatrix} \cos \phi \\ \sin \phi \\ 1 \end{pmatrix} d\phi}_{\begin{array}{l} = \begin{pmatrix} \pi K_{1n} \\ \pi \tilde{K}_{1n} \\ 0 \end{pmatrix} \text{ for } m=1 \\ = \begin{pmatrix} 0 \\ 0 \\ 2\pi K_{0n} \end{pmatrix} \text{ for } m=0 \\ = \mathbf{0} \text{ otherwise} \end{array}} \\ &\quad (112) \end{aligned}$$

The integral over  $\phi$  the in  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{y}}$  direction is only nonzero when  $m = 1$ . While is it only nonzero in the  $\hat{\mathbf{z}}$  direction for  $m = 0$ . When  $m$  is 1 the integral over  $\theta$  in the  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{y}}$  direction is the same. And by using the orthogonality relationship (108) and  $\sin \theta = -P_1^1$ :

$$\int_0^\pi P_n^1 \sin \theta \sin \theta d\theta = - \int_0^\pi P_n^1 P_1^1 \sin \theta d\theta = -\frac{4}{3} \delta_{n1} \quad (113)$$

When  $m = 0$  the integral over  $\theta$  in the  $\hat{\mathbf{z}}$  direction can be evaluated by using the orthogonality relationship (108) and  $\cos \theta = P_1^0$ :

$$\int_0^\pi P_n^1 \cos \theta \sin \theta d\theta = \int_0^\pi P_n^0 P_1^0 \sin \theta d\theta = \frac{2}{3} \delta_{n1} \quad (114)$$

Hence we get:

$$\iint_S a\hat{\mathbf{r}}\Gamma_{-n-1} dS = \begin{cases} \frac{4}{3}\pi a K_{01} \hat{\mathbf{z}}, & \text{for } n = 1, m = 0 \\ -\frac{4}{3}\pi a K_{11} \hat{\mathbf{x}} - \frac{4}{3}\pi a \tilde{K}_{11} \hat{\mathbf{y}}, & \text{for } n = 1, m = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (115)$$

### A.3.3 Evaluating $\iint_S \nabla \Gamma_{-n-1} dS$

Here it is helpful to take the gradient in Cartesian coordinates, but write the gradient  $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}$  in terms of  $\frac{\partial}{\partial r}, \frac{\partial}{\partial \theta}, \frac{\partial}{\partial \phi}$ :

$$\iint_S \nabla \Gamma_{-n-1} dS = \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \left( \begin{array}{c} \sin \theta \cos \phi \frac{\partial \Gamma_{-n-1}}{\partial r} + \frac{1}{r} \cos \phi \cos \theta \frac{\partial \Gamma_{-n-1}}{\partial \theta} - \frac{1}{r \sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \\ \sin \theta \sin \phi \frac{\partial \Gamma_{-n-1}}{\partial r} + \frac{1}{r} \sin \phi \cos \theta \frac{\partial \Gamma_{-n-1}}{\partial \theta} - \frac{1}{r \sin \theta} \frac{\partial \Gamma_{-n-1}}{\partial \phi} \\ \cos \theta \frac{\partial \Gamma_{-n-1}}{\partial r} - \frac{1}{r} \sin \theta \frac{\partial \Gamma_{-n-1}}{\partial \theta} \end{array} \right) r^2 \sin \theta d\theta d\phi \Big|_{r=a} \quad (116)$$

We consider the  $\hat{\mathbf{x}}$ -direction first:

$$\begin{aligned} & r^2 \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \left( \sin \theta^2 \cos \phi \frac{\partial \Gamma_{-n-1}}{\partial r} + \frac{1}{r} \sin \theta \cos \phi \cos \theta \frac{\partial \Gamma_{-n-1}}{\partial \theta} - \frac{1}{r} \sin \phi \frac{\partial \Gamma_{-n-1}}{\partial \phi} \right) d\phi d\theta \Big|_{r=a} \\ &= a^{-n+2} \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \left( (-n-1) \sin^2 \theta P_n^m (K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi) \cos \phi \right. \\ &\quad \left. - \cos \theta \sin^2 \theta \frac{dP_n^m}{d \cos \theta} (K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi) \cos \phi \right. \\ &\quad \left. - P_n^m (-K_{mn} \sin m\phi + \tilde{K}_{mn} \cos m\phi) \cdot m \sin \phi \right) d\theta d\phi \end{aligned} \quad (117)$$

The  $\phi$  integral for all three terms are only non-zero when  $m = 1$ . In the second term we rewrite  $\sin^2 \theta \cdot \frac{dP_n^m}{d \cos \theta} = (n+1) \cos \theta P_n^m - (1+n-m) P_{n+1}^m$ . The  $\hat{\mathbf{x}}$ -direction integral when  $m = 1$  then becomes:

$$= a^{-n+2} \cdot \pi K_{1n} \int_{\theta=0}^{\theta=\pi} \left( -(n+1) \sin^2 \theta P_n^1 - (n+1) \cos^2 \theta P_n^1 + n \cos \theta P_{n+1}^1 + P_n^1 \right) d\theta \quad (119)$$

$$= a^{-n+2} \cdot \pi K_{1n} n \cdot \int_{\theta=0}^{\theta=\pi} \left( -P_n^1 + \cos \theta \cdot P_{n+1}^1 \right) d\theta \quad (120)$$

The integral in (120) is always zero. To prove this we first use integration by substitution with  $\cos \theta = x$ . We remind the reader again that the argument in  $P_n^m$  is  $\cos \theta$ . So the above integral becomes:

$$= a^{-n+2} \cdot \pi K_{1n} n \cdot \int_{-1}^1 \frac{(-P_n^1(x) + x \cdot P_{n+1}^1(x))}{\sqrt{1-x^2}} dx \quad (121)$$

Now we will use the definition of Associated Legendre polynomials:  $P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x)$ . Here  $P_n$  is the Legendre polynomial of degree  $n$ . With this the integral becomes:

$$= a^{-n+2} \cdot \pi K_{1n} n \cdot \int_{-1}^1 \left( \frac{dP_n}{dx} - x \cdot \frac{dP_{n+1}}{dx} \right) dx \quad (122)$$

Using integration by parts, we have:

$$= a^{-n+2} \cdot \pi K_{1n} n \cdot \left\{ \left[ P_n \right]_{-1}^1 - \left[ xP_{n+1} \right]_{-1}^1 + \underbrace{\int_{-1}^1 P_{n+1} dx}_{= 0 \text{ for all } n \geq 0, \text{ due to the Orthogonality principle}} \right\} \quad (123)$$

$$= a^{-n+2} \cdot \pi K_{1n} n \cdot \left\{ 1 - 1 - (-1)^n - (-1)^{n+1} \right\} \quad (124)$$

$$= 0 \text{ for all } n \quad (125)$$

Here we have used the fact that  $P_n(1) = 1$ ,  $P_n(-1) = (-1)^n$  and  $P_{n+1}(1) = 1$ ,  $P_{n+1}(-1) = (-1)^{n+1}$ . We see that the integral for all  $m$  and  $n$  is zero. Hence there is no  $\hat{x}$ -direction contribution from this integral. The integral in the  $\hat{y}$ -direction is also zero and looks almost identical to what we have went through now. We have therefore not included it here. Now for the  $\hat{z}$ -direction:

$$\int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=\pi} \left( \cos \theta \frac{\partial \Gamma_{-n-1}}{\partial r} - \frac{1}{r} \sin \theta \frac{\partial \Gamma_{-n-1}}{\partial \theta} \right) r^2 \sin \theta d\theta d\phi \Big|_{r=a} \quad (126)$$

$$= a^{-n+2} \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} ((-n-1)P_n^m \cos \theta \sin \theta + \sin \theta^3 \frac{dP_n^m}{d \cos \theta}) \cdot (K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi) d\theta d\phi \quad (127)$$

Here we will again use that  $\sin^2 \theta \cdot \frac{dP_n^m}{d \cos \theta} = (n+1) \cos \theta P_n^m - (1+n-m)P_{n+1}^m$ . And note the  $\phi$  integral is only non-zero when  $m = 0$ :

$$= a^{-n+2} \cdot 2\pi K_{0n} \int_{\theta=0}^{\theta=\pi} (-(n+1)P_n^0 \cos \theta \sin \theta + (n+1)P_n^0 \cos \theta \sin \theta - (n+1)P_{n+1}^0 \sin \theta) d\theta \quad (128)$$

$$= a^{-n+2} \cdot 2\pi K_{0n} \int_{\theta=0}^{\theta=\pi} -(n+1)P_{n+1}^0 \sin \theta d\theta \quad (129)$$

$$= -a^{-n+2} \cdot 2\pi K_{0n} \cdot (n+1) \underbrace{P_0^0 \cdot P_{n+1}^0 \sin \theta d\theta}_{\begin{array}{l} = 2 \text{ for } n = -1 \\ = 0 \text{ otherwise} \end{array}} \quad (130)$$

We see from the  $\theta$  integral that all  $n$  gives 0. The final results for  $\hat{x}$ ,  $\hat{y}$  and  $\hat{z}$  direction is therefore:

$$\iint_S \nabla \Gamma_{-n-1} dS = 0, \text{ for all } n > 0 \text{ and } n \geq m \geq 0 \quad (131)$$

### A.3.4 Evaluating $\iint_S \nabla \times (\mathbf{r} \Gamma_{-n-1}) dS$

For this integral  $\iint_S \nabla \times (\mathbf{r} \Gamma_{-n-1}) dS$ , it is also useful to work in Cartesian coordinates. Meaning we take the curl in Cartesian coordinates, but express  $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}$  in terms of  $\frac{\partial}{\partial r}, \frac{\partial}{\partial \theta}, \frac{\partial}{\partial \phi}$ . In the  $\hat{z}$ -direction the integral reads:

$$\int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \left( -\sin \theta \cdot \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} (r \Gamma_{-n-1}) \right) r^2 \sin \theta d\theta d\phi \Big|_{r=a} \quad (132)$$

$$= -a^{-n} \int_{\theta=0}^{\theta=\pi} P_n^m \sin \theta d\theta \underbrace{\int_{\phi=0}^{\phi=\pi} (-K_{mn} \sin m\phi + \tilde{K}_{mn} \cos m\phi) \cdot m d\phi}_{=0 \text{ for all } m} \quad (133)$$

The  $\phi$  integral is always zero for all  $m$ . Hence the integral in the  $\hat{\mathbf{z}}$ -direction is always zero. Now for the  $\hat{\mathbf{x}}$ -direction:

$$\int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} \left( \cos \theta \cos \phi \frac{1}{r \sin \theta} \frac{\partial}{\partial \phi} (r \Gamma_{-n-1}) - \sin \phi \cdot \left( -\frac{1}{r} \frac{\partial}{\partial \theta} (r \Gamma_{-n-1}) \right) \right) \sin \theta r^2 d\theta d\phi \Big|_{r=a} \quad (134)$$

$$= a^{-n} \int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} (\cos \theta P_n^m (-K_{mn} \sin m\phi + \tilde{K}_{mn} \cos m\phi) \cdot m \cos \phi \quad (135)$$

$$- \sin^2 \theta \frac{dP_n^m}{d \cos \theta} (K_{mn} \cos m\phi + \tilde{K}_{mn} \sin m\phi) \sin \phi) d\theta d\phi \quad (136)$$

Here we will again use that  $\sin^2 \theta \cdot \frac{dP_n^m}{d \cos \theta} = (n+1) \cos \theta P_n^m - (1+n-m) P_{n+1}^m$ . And note the  $\phi$  integral is only non-zero when  $m=1$ . The above integral becomes:

$$= a^{-n} \pi \tilde{K}_{1n} \int_0^\pi (\cos \theta P_n^1 - (n+1) \cos \theta P_n^1 + n P_{n+1}^1) d\theta \quad (137)$$

$$= a^{-n} \pi \tilde{K}_{1n} \int_0^\pi (-n \cos \theta P_n^1 + n P_{n+1}^1) d\theta \quad (138)$$

We now use integration by substitution with  $\cos \theta = x$ . We remind the reader again that the argument in  $P_n^m$  is  $\cos \theta$ . So the above integral becomes:

$$= a^{-n} \pi \tilde{K}_{1n} n \cdot \int_{-1}^1 \frac{(-x \cdot P_n^1(x) + P_{n+1}^1(x))}{\sqrt{1-x^2}} dx \quad (139)$$

Now we will use the definition of Associated Legendre polynomials:  $P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x)$ . Here  $P_n$  is the Legendre polynomial of degree  $n$ . With this the integral becomes:

$$= a^{-n} \pi \tilde{K}_{1n} n \cdot \int_{-1}^1 \left( x \cdot \frac{dP_n}{dx} - \frac{dP_{n+1}}{dx} \right) dx \quad (140)$$

Using integration by parts, we have:

$$= a^{-n} \pi \tilde{K}_{1n} n \cdot \left\{ \left[ x P_n \right]_{-1}^1 + \underbrace{\int_{-1}^1 P_n dx}_{= 0 \text{ for all } n \geq 1, \text{ due to the Orthogonality principle}} - \left[ P_{n+1} \right]_{-1}^1 \right\} \quad (141)$$

$$= a^{-n} \pi \tilde{K}_{1n} n \cdot \left\{ 1 - 1 + (-1)^n + (-1)^{n+1} \right\} \quad (142)$$

$$= 0 \text{ for all } n \quad (143)$$

Here we have used the fact that  $P_n(1) = 1$ ,  $P_n(-1) = (-1)^n$  and  $P_{n+1}(1) = 1$ ,  $P_{n+1}(-1) = (-1)^{n+1}$ . The integral  $\iint_S \nabla \times (\mathbf{r} \Gamma_{-n-1}) dS$  in the  $\hat{\mathbf{y}}$ -direction is also zero and is almost identical to the integral in the  $\hat{\mathbf{x}}$ -direction. We have therefore not included it here. The overall result for this integral is therefore:

$$\iint_S \nabla \times (\mathbf{r} \Gamma_{-n-1}) dS = 0, \text{ for all } n > 0 \text{ and } n \geq m \geq 0 \quad (144)$$