

The Hydrodynamics of Active
Particles Inside of a Porous
Container and the Galerkin
Boundary Element Discretization
of Stokes Flow

Caltech

Kevin J. Marshall

Department of Chemical Engineering
California Institute of Technology
Pasadena, California

In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

2018
Defended October 4, 2017

© 2018
Kevin J. Marshall
ORCID: 0000-0001-6025-7674
All rights reserved

Acknowledgements

It has been a pleasure and privilege to work with my advisor, Professor John Brady. John has supported me over the past several years with, essentially, unrestricted funding. This is a special right that very few graduate students are ever granted. Early on he believed in me as an independent scientist. He never demanded that I study a particular problem, and only ever provided interesting and thoughtful suggestions for research directions. This gave me a wonderful level of autonomy and independence in choosing exactly what I wanted to learn about and study. This independence allowed me to develop my own ideas and research tastes. In particular, I have become fascinated with the Boundary Element Method.

I have also found a great deal of enjoyment in working with Professor Alan Barr. We connected during his computer graphics course, CS 171, and quickly found that we shared a diverse set of research interests. Over the past several years, Al has provided me with unofficial advisement in both research and life. He also helped me get accepted into the Summer Physiology Course of 2015 at the Marine Biological Laboratory. Al and I tend to think about research problems in very similar ways. I consider him to be a great friend.

Lisa Gilmore deserves a great amount of recognition in providing me with incredible home support. She has been by my side throughout this entire PhD. During times of struggle, she has always been there to provide reason and direction. She is a driven and motivated woman, and seeing her success has always motivated me to work towards obtaining mine.

Abstract

In this thesis I formulate and present a novel and new framework for simulating the dynamics of arbitrarily shaped active or passive particles immersed in a Stokesian fluid and evolving under confinement by a porous container or in free space. I use a completed double layer boundary integral equation to model the particle's dynamics and combine this with a new formulation that uses a second kind integral equation for describing the motion of the porous container. This newly formulated porous container model permits fluid to pass through the container's surface at a velocity in proportion to a discontinuous jump in stress across the container's surface. This jump in stress is induced by the active particle's motion. The proposed porous container model is general in the sense that it allows fluid to pass through the membrane with finite tangential and normal velocity components. I obtain the exact analytical solution to this model when the active particle and porous container are perfectly concentric. In addition, I numerically solve this system of boundary integral equations for arbitrary particle positions, and fully characterize the particle and container dynamics by performing a vast number of trajectory studies. Both the container and particle are seen to move in complicated ways owing to their self and pairwise hydrodynamic interactions. This system is studied over a vast parameter space, for multiple container to particle size ratios, multiple types of active particles, and various permeability parameterizations. This coupled set of particle and container boundary integral equations is discretized using a novel interpretation and new extension of the Galerkin Boundary Element discretization to multi-body particle systems in Stokes flow. I also implement and extend an h -adaptive conformal mesh refinement algorithm to accurately resolve near-contact particle and container interactions. In addition, I perform all Galerkin BEM calculations on CUDA enabled GPUs, allowing for these simulations to be run on modern desktop computers in seconds. I combine all of these techniques in a modern C++ Galerkin Boundary Element Method computational framework called GPUGBEM.

Table of contents

Acknowledgments	iii
Abstract	iv
List of figures	ix
List of tables	xxii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	2
1.3 Thesis Overview	3
2 Basic Concepts and Tools	5
2.1 Frames of Reference and Transformations	5
2.2 Rigid Body Dynamics	7
2.3 Rigid Body Position and Orientation	9
2.4 Linear Velocity	9
2.5 Angular Velocity	9
2.6 Velocity of a Particle (or point)	10
2.7 Centroids and Centers of Mass	11
2.8 Force and Torque	12
2.9 Linear Momentum	13
2.10 Angular Momentum	13
2.11 Inertia Tensor	14
2.12 Rigid Body Equations of Motion	15
2.13 Rigid Body Dynamics Algorithm	15
2.14 Quaternion Representation of $R(t)$	16
2.15 Updating the Rigid Body Mesh	17
2.16 Integration in Time	17

2.17	Linear Multistep Methods	18
2.18	Important Notes and Future Work	19
3	Boundary Integral Methods for Viscous Flows	20
3.1	The Stokes Equations	20
3.2	Notation	21
3.3	Singularity Solutions	22
3.3.1	Important Properties	25
3.4	Lorentz reciprocal relation	25
3.4.1	Boundary Integral Equations	26
3.4.2	Conventions for Normal Vectors and Interior / Exterior Problems	27
3.4.3	The Calderón Projection	28
3.5	The Completed Double Layer Representation	30
3.5.1	Completion Schemes	32
3.5.1.1	The Resistance Formulation	33
3.5.1.2	The Mobility Formulation	34
3.5.2	The CDL Representation of an Active Particle	35
3.6	Conclusions	36
4	The GPUGBEM Framework and the Galerkin Discretization	38
4.1	Boundary Element Methods	39
4.1.1	Symmetric Galerkin Formulation	42
4.1.2	Boundary Approximation and Discretization	43
4.1.2.1	Linear Quadrilateral Shape Functions	46
4.1.2.2	Linear Triangle Shape Functions	47
4.1.3	Galerkin Discretization	47
4.1.4	Transformation of Integrals to Reference Elements	48
4.1.5	Discrete Forms of Galerkin Boundary Element Integrals	49
4.1.6	Mass Matrix	50
4.2	The Galerkin Discretization Applied to Linearized Viscous Flows	51
4.2.1	Discretization of the Completed Double Layer Representation	52
4.2.1.1	Single Particle Solution	54
4.2.2	Multiple Particles and General Matrix Representations	54
4.3	Mobility and Resistance Formulation for Particle-in-Cell Models	55
4.3.1	Particle Contribution	56
4.3.2	Container Contribution	57
4.3.3	Formulation and Solution of the System	57
4.3.4	Mobility Problem	58

4.3.5	Resistance Problem	58
4.4	The GPUGBEM Framework	59
4.4.1	GPU GBEM Computations Using CUDA	62
4.4.2	h -Adaptive Meshing in GPUGBEM	63
5	The Problem of an Active Particle Inside of a Porous Container	65
5.1	Contributions	66
5.2	Background	68
5.2.1	Tangential Squirmers	69
5.3	Analytical Solutions	71
5.3.1	The Rigid Non-Porous Container	73
5.3.2	The Rigid Porous Container	74
5.3.3	Resolving Flow in the Porous Region	77
5.3.3.1	Limiting Behaviors	79
5.3.3.2	Velocity Fields	80
5.4	Boundary Integral Formulation	82
5.4.1	Squirmer Contribution	83
5.4.2	Container Contribution	85
5.5	Galerkin Discretization	86
5.5.1	Continuous Boundary Element Spaces	87
5.5.2	Galerkin BEM	89
5.5.3	Matrix Representation of Galerkin BEM Integrals	90
5.5.4	Global Linear System	93
5.6	Numerical Solution and Comparison with the Analytical Model	95
5.6.1	Parameterization	95
5.6.2	Concentric Geometry BEM Results	96
5.7	Mobility Field Calculations	97
5.7.1	Basic Trajectory Calculations and Improvements	98
5.7.2	Interpolation Based Mobility Solutions and Trajectory Sampling	99
5.7.3	Mobility Solution Sampling Grid	100
5.7.4	Mobility Solution Fields: Up-sampling to a Finer Grid	102
5.7.4.1	Container Velocity Field, \mathbf{U}^m Scalings	103
5.7.4.2	Effect of Permeability Resistances on \mathbf{U}^m and \mathbf{U}^{sq}	104
5.7.4.3	Effect of Squirmer Type on \mathbf{U}^m and \mathbf{U}^{sq}	105
5.7.4.4	Effect of b/a on \mathbf{U}^m and \mathbf{U}^{sq}	107
5.8	Squirmer and Container Trajectory Calculations	107
5.8.1	Squirmer Container Collisions	109
5.8.2	3D Squirmer Trajectories	110

5.8.3	3D Container Trajectories	111
5.8.4	Net and Total Distance, Fastest Squirmer Translocation . . .	113
5.9	Discussion and Conclusions	114
	References	118

List of figures

2.1	The world space (space fixed) and body space frames of reference are shown for a group of spherical particles in a 3D computation cell. Also shown is the formalism of vector notation for particle position, $\boldsymbol{x}(t)$, and orientation $\boldsymbol{e}_o(t)$	8
5.1	The container-squirmer geometry is shown with a Cartesian coordinate system. The fluid region is partitioned into interior, porous, and exterior regions, respectively Ω_i , Ω_p , and Ω_e	68
5.2	The center of mass translation velocities of the squirmer, U_z^{sq} and a forced particle U_z^{fp} are plotted as a function of the container particle size ratio, b/a . The squirmer translational velocity approaches its free space squirming velocity fairly rapidly in comparison to the forced particle as $b/a \rightarrow \infty$	73
5.3	The normalized z-component of the squirmer's and container's translational velocity, \boldsymbol{U}_z^{sq} and \boldsymbol{U}_z^m , is shown versus the container to squirmer size ration b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$	78
5.4	The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	81

- 5.5 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized squirmer's translational velocity, $U_z^{sq}/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model. . . . 97
- 5.6 In this plot all regular Cartesian (query) grid points are circled in red. Grid points that are inside (outside) of the region defined by $r < r_{cp}^{max}$ are shown in green (blue). The mobility solution is solved for a squirmer placed at all green points (841 query points and 385 mobility solves). A thick black circle shows the container boundary at $r = b$. An inner red circle shows where $r = b - a\delta$. The pink or fuchsia colored circle shows where $r = r_{cp}^{max}$. The inner gray circle is sized according to $r = a$ giving a sense of the squirmer's size. 101
- 5.7 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. . . 105
- 5.8 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the tangential direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. . . 106
- 5.9 In this figure, an example output of the h-adaptive meshing algorithm, based on the nearly singular distance ration $d_{xy} \leq 1$ is shown for two spheres. Colors are related to triangle diameters. This application of NVB to BEM results in a conformal mesh (i.e. there are no hanging nodes or so-called T-edges). 123
- 5.10 The normalized z-component of the squirmer's translational velocity, U_z^{sq} , is shown versus the container to squirmer size ration b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} > R_{\perp}$ 124

5.11	The normalized z-component of the container's translational velocity, U_z^m , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} > R_{\perp}$	125
5.12	The normalized z-component of the squirmer's translational velocity, U_z^{sq} , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} < R_{\perp}$	126
5.13	The normalized z-component of the container's translational velocity, U_z^m , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} < R_{\perp}$	127
5.14	The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	128
5.15	The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	129
5.16	The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	130
5.17	The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	131
5.18	The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	132
5.19	The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	133

5.20	The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	134
5.21	The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.	135
5.22	The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	136
5.23	The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	137
5.24	The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	138
5.25	The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	139
5.26	The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	140
5.27	The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	141
5.28	The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.	142

- 5.29 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction. 143
- 5.30 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction. 144
- 5.31 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized squirmer's translational velocity, $U_z^{sq}/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model. . . . 145
- 5.32 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized squirmer's translational velocity, $U_z^{sq}/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow is made to dominating (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model. . . . 146
- 5.33 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized container's translational velocity, $U_z^m/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model. . . . 147

- 5.34 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized container's translational velocity, $U_z^m/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model. . . . 148
- 5.35 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized container's translational velocity, $U_z^m/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model. . . . 149
- 5.36 In this plot all regular Cartesian (query) grid points are circled in red. Grid points that are inside (outside) of the region defined by $r < r_{cp}^{max}$ are shown in green (blue). The mobility solution is solved for a squirmer placed at all green points (289 query points and 96 mobility solves). A thick black circle shows the container boundary at $r = b$. An inner red circle shows where $r = b - a\delta$. The pink or fuchsia colored circle shows where $r = r_{cp}^{max}$. The inner gray circle is sized according to $r = a$ giving a sense of the squirmer's size. 150
- 5.37 In this plot all regular Cartesian (query) grid points are circled in red. Grid points that are inside (outside) of the region defined by $r < r_{cp}^{max}$ are shown in green (blue). The mobility solution is solved for a squirmer placed at all green points (3249 query points and 2040 mobility solves). A thick black circle shows the container boundary at $r = b$. An inner red circle shows where $r = b - a\delta$. The pink or fuchsia colored circle shows where $r = r_{cp}^{max}$. The inner gray circle is sized according to $r = a$ giving a sense of the squirmer's size. 151

- 5.38 This plot shows an up-sampled grid for $b/a = 3$ and $\kappa = 3$. Up-sampled grid points are shown in orange. Original grid points are shown in green (inside the circle mask defined by $r = r_{cp}^{max}$) and blue outside the circle mask. 152
- 5.39 This plot shows an up-sampled grid for $b/a = 5$ and $\kappa = 3$. Up-sampled grid points are shown in orange. Original grid points are shown in green (inside the circle mask defined by $r = r_{cp}^{max}$) and blue outside the circle mask. 153
- 5.40 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 154
- 5.41 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 155
- 5.42 This plot shows the squirmer velocities that would result if a $\beta = 0$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 156
- 5.43 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 157
- 5.44 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 158

- 5.45 This plot shows the squirmer velocities that would result if a $\beta = 0$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 159
- 5.46 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. . . 160
- 5.47 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 161
- 5.48 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. . . 162
- 5.49 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 163
- 5.50 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. . . 164
- 5.51 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 165

- 5.52 This plot shows the container velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 166
- 5.53 This plot shows the container velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively. 167
- 5.54 Several squirmer trajectories are shown for $\beta = -5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. 168
- 5.55 Several squirmer trajectories are shown for $\beta = 0$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. 169
- 5.56 Several squirmer trajectories are shown for $\beta = 5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. 170
- 5.57 Several squirmer trajectories are shown for $\beta = -5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. 171
- 5.58 Several squirmer trajectories are shown for $\beta = 0$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. 172
- 5.59 Several squirmer trajectories are shown for $\beta = 5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. 173
- 5.60 Trajectories for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially outward (different from the $\beta = +5$ type squirmer) as they swim upward in z 174

- 5.61 Trajectories for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to swim along z showing only small radial migrations as they near the container wall (different from the $\beta = \pm 5$ type of squirmers). 175
- 5.62 Trajectories for $\beta = +5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially inward as they swim upward in z . . . 176
- 5.63 Trajectories for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially outward (different from the $\beta = +5$ type squirmer) as they swim upward in z . This effect is much weaker than was seen for the weak tangential resistance case. 177
- 5.64 Trajectories for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to swim along z showing only small radial migrations as they near the container wall (different from the $\beta = \pm 5$ type of squirmers). 178
- 5.65 Trajectories for $\beta = +5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially inward as they swim upward in z . This effect is much weaker than was seen for the weak tangential resistance case. 179
- 5.66 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz plane. 180
- 5.67 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz plane. 181

- 5.68 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane. 182
- 5.69 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane. 183
- 5.70 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane. 184
- 5.71 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane. 185
- 5.72 Container trajectories are shown for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$ 186
- 5.73 Container trajectories are shown for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$ 187
- 5.74 Container trajectories are shown for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$ 188
- 5.75 Container trajectories are shown for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$ 189

5.76	Container trajectories are shown for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	190
5.77	Container trajectories are shown for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	191
5.78	The total and net distance that a $\beta = -5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	192
5.79	The total and net distance that a $\beta = 0$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	193
5.80	The total and net distance that a $\beta = 5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	194
5.81	The total and net distance that a $\beta = -5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	195
5.82	The total and net distance that a $\beta = 0$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	196
5.83	The total and net distance that a $\beta = 5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_Z$	197

- 5.84 The difference between the net distance that a pusher ($\beta = -5$) and puller ($\beta = +5$) swim is shown as a function of time. In this case the trajectory dynamics are shown under weak tangential resistance to permeable flow. Pushers are seen to move relative distances greater than pullers at short times. At long times, pullers move greater relative distances. Roughly speaking, this is due to the fact that pushers tend to be attracted towards the container walls. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$ 198
- 5.85 The difference between the net distance that a pusher ($\beta = -5$) and puller ($\beta = +5$) swim is shown as a function of time. In this case the trajectory dynamics are shown under weak normal resistance to permeable flow. Pushers are seen to move relative distances greater than pullers at short times. At long times, pullers move greater relative distances. Roughly speaking, this is due to the fact that pushers tend to be attracted towards the container walls. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$ 199
- 5.86 The difference between the net distance that a pusher ($\beta = -5$) swims in the weak tangential and weak normal resistance regimes is plotted as a function of time. Pushers are seen to favor motion in weak tangential resistance at short times and weak normal resistance at long times. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$ 200
- 5.87 The difference between the net distance that a puller ($\beta = +5$) swims in the weak tangential and weak normal resistance regimes is plotted as a function of time. Pullers are seen to favor motion in weak normal resistance for almost all times. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. . . 201

List of tables

5.1	Net squirmer and container swimming directions are shown for the \mathbf{P}^0 parameterization relative to z . An up arrow, \uparrow indicates net motion in $+z$. A down arrow, \downarrow , indicates net motion in $-z$. The first and second arrows represent the net directional motion of the squirmer and container respectively. Arrows are to be understood as the overall direction of the net motion of the squirmer and container for sufficiently long trajectories.	112
-----	---	-----

Chapter 1

Introduction

1.1 Motivation

Physics was the science of the 20th century, and few disagree that biology will be the science of the 21st century. Discoveries about our living world are brought about daily by advances in molecular biology, neuroscience, cell biology, and other biological disciplines. Enough is currently known about individual molecules and their interactions inside cells that cellular life-cycles and cellular behaviors can be simulated. Due to the advent of genomics and other high throughput techniques, it has recently become possible to construct a full-cell computational model [29] dubbed cells *in silico*, Latin for cells *in silicon*. Although this model recapitulates a broad set of experimental data, the mathematics and perspectives of the model are not directly connected to an explicit interpretation of cellular particle dynamics, specifically from a fluid mechanical perspective.

There is currently great interest in understanding how hydrodynamic interactions affect particle motions inside biological cells. A great deal of effort and work has been put into performing molecular simulations that aim to ultimately develop quantitative comparisons with *in vivo* molecular motion inside biological cells, [12]. A fair amount of this work is on understanding mechanisms for the self-assembly of biological molecules and sub-structures [3], [14], [39] (e.g. protein folding and lipid bilayer self-assembly). It is also generally accepted and understood that hydrodynamic interactions play a significant role in determining the dynamical properties of concentrated macromolecular systems [2], [1]. However, all of these studies simulate concentrated dense suspensions of stick or slip-stick spheres and completely abstract away the complex surface geometries of biological structures. In these dense suspensions of spherical particles there are also immense amounts of hydrodynamic screening, which makes it very hard to develop intuition about individual particle motions and behaviors. In addition, these

simulations are usually performed with periodic boundary conditions where there is no inclusion of container confinement effects. A notable exception is in the recent work by Nazockdast and Shelley, [43], where fiber assemblies are simulated inside of a realistic cellular geometry. This thesis aims to explore basic cellular particle dynamics, in confinement, and from a rigorous fluid mechanical prospective, but does so using a method that remains faithful to particle geometries. No attempt is made at simulating dense suspensions, rather effort is put into understanding the strikingly complicated dynamics of a single active particle that moves as a force and torque free body inside of a porous container. The simulation of larger numbers of active particles is left as future work.

1.2 Contributions

In this thesis we rigorously formulate and solve an applied problem where an actively propelled particle, immersed in a viscous fluid, moves inside of a geometry that is reminiscent of the most basic cell imaginable: a spherical container. The container is special in that it is porous and leaks fluid at a velocity proportional to a jump in stress across its surface. This jump in stress is induced by the active particle's motion. Exact analytical solutions for the dynamics of this problem are obtained when the particle and container geometry is perfectly concentric. With this analytical solution, the problem serves as an important benchmark for testing numerical schemes, which ultimately may be geared towards solving more complicated problems, with arbitrarily shaped geometries and flexible or charged surfaces. The particle and container geometries are spherical for convenience. There is no assumption in the numerical methods or restrictions on the algorithms that forces the use of this geometry.

In order to solve this problem, a very special new numerical framework is needed. We develop a computational framework, called GPUGBEM, which formulates multi-body hydrodynamics problems in a very efficient way. In order to build this framework, this thesis presents the detailed theoretical construction of Stokes flow boundary element operators under the Galerkin discretization. This construction has not been seen before for multi-body colloidal hydrodynamics problems. Its use in obtaining the presented numerical results is both novel and new. This framework expresses boundary integral equations using an intuitive language that is wrapped deeply in object oriented design. Multi-body hydrodynamics problems may be solved in a very straightforward manner with black-box controls over accuracy and spatial mesh adaptivity, automatic formulation of Galerkin boundary element matrices, and automatic cross coupling of particle-particle pairwise hydrodynamic interactions. Implementation may easily

be switched between the mobility and resistance formulations. The framework also very efficiently models the dynamics of a special class of active particles known as *Blake squirmers*. In addition, GPUGBEM is equipped with interfaces to the NVIDIA CUDA API and performs all boundary element integrations on the GPU using efficient splitting techniques and load balancing optimizations. Without GPUGBEM and its object oriented design structure, solution of the particle porous container problem would be incredibly error prone. The GPUGBEM framework is coupled to a novel conformal adaptive meshing algorithm. This algorithm operates in time $O(N \log N)$ by utilizing advanced spatial querying data structures.

1.3 Thesis Overview

The remainder of this thesis is organized as follows:

Chapter 2 reviews relevant ideas and formulas that are used in all subsequent chapters. A detailed description is given for the appropriate design and implementation of rigid body dynamics simulations and its coupling to the fluid-structure interaction problem in Stokes flow.

Chapter 3 reviews important ideas from the boundary integral formulation of linearized viscous flows. The Stokes equations are introduced together with notation that is used throughout this thesis. Singularity solutions of Stokes flows are reviewed and the boundary integral equations of linearized viscous flows are derived. The chapter concludes with the formulation of the completed double layer boundary integral representation. This method is described in detail since it is ultimately used to find solutions for the dynamics of the particle porous container problem.

Chapter 4 discusses the theoretical formulation of the Galerkin boundary integral method. Explicit constructions for the Galerkin discretization of Stokes BEM operators are given and explained in detail. We discuss how the Galerkin discretization maps onto multi-body problems in Stokes flow. Several constructions of the resistance and mobility problems are provided as examples. Next, several important object oriented design principles are discussed in the context of the GPUGBEM framework. Insight is given into how GBEM calculations map to GPGPU computation. Finally the implementation of a conformal h -adaptive meshing algorithm is described. This h -adaptive meshing algorithm is an extension of the newest vertex bisection method

to triangle 2-manifolds. Novelty is found in how the algorithm is coupled with the Galerkin Boundary Element Method.

Chapter 5 presents the particle porous container problem. The active particle is represented as a tangential *Blake squirmer*. The problem where a squirmer moves inside of a non-porous container is first studied. Exact analytical solutions are obtained for this problem and are compared with the solution for the well-known forced particle concentric sphere problem. Next, a porous container model is proposed that derives in part from Darcy's law, but is more general in the sense that it allows for both normal and tangential stress jumps to exist across the container surface. This model is parameterized by a set of normal and tangential permeable resistances. An exact solution for this porous model is obtained, and is shown to agree with the non-porous container solution in appropriate limits. The flow fields for these analytical models show vortical regions of flow that are axisymmetric to the particles swim orientation and in both the anterior and posterior regions of the container relative to its center of mass. Next the full boundary integral formulation is given for the system. The set of boundary integral equations is solved using the Galerkin discretization of Chapter 4 and the GPUGBEM framework. The solution is seen to compare well with the analytical models. A full trajectory analysis of the particle and container is given. A novel interpolation algorithm is formulated and used to evaluate these trajectories using pre-resolved fluid mechanics calculations. Finally, squirmer efficiencies are quantified by examining which squirmer type has the ability to translocate across the container the fastest.

Chapter 2

Basic Concepts and Tools

In this chapter we describe various principles and techniques that are needed to simulate the rigid body dynamics of particles. This formulation loosely follows [4]. Ultimately these techniques are coupled with the Stokes flow mobility solution for the particle container system (Chapter 5), and are used to accurately resolve particle and container trajectories.

2.1 Frames of Reference and Transformations

In this thesis, the world space (WS), synonymous with the space fixed (SF) frame of reference, is defined using a Cartesian coordinate system with basis vectors, $\mathbf{e}_X, \mathbf{e}_Y, \mathbf{e}_Z$, that are taken to span a computational cell in 3D. Inside of the world space, one may embed other orthonormal, rotated coordinate systems. These rotated coordinate systems, when attached to a physical object (surface, volume, set of points), define the object's body space (BS) and have basis vectors which will be denoted by $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$. These basis vectors span the body space, and are collectively referred to as the body fixed frame of reference. In general there is always a so-called *passive* transformation that takes the world space to the body space. This is also known more commonly as a change of basis assuming that both frames of reference share the same origin. Otherwise there is always a *passive* affine transformation that moves between a translated body and world frame of reference. Under passive transformations a given vector remains fixed while the basis is transformed. The construction is simple and is given by solving for the transformation matrix, \mathbf{R}_W^B , such that $\mathbf{e}_{B:i} = \mathbf{R}_{W:ij}^B \mathbf{e}_{W:j}$,

$$\begin{bmatrix} \mathbf{e}_{B:1} \\ \mathbf{e}_{B:2} \\ \mathbf{e}_{B:3} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{B:1} \cdot \mathbf{e}_{W:1} & \mathbf{e}_{B:1} \cdot \mathbf{e}_{W:2} & \mathbf{e}_{B:1} \cdot \mathbf{e}_{W:3} \\ \mathbf{e}_{B:2} \cdot \mathbf{e}_{W:1} & \mathbf{e}_{B:2} \cdot \mathbf{e}_{W:2} & \mathbf{e}_{B:2} \cdot \mathbf{e}_{W:3} \\ \mathbf{e}_{B:3} \cdot \mathbf{e}_{W:1} & \mathbf{e}_{B:3} \cdot \mathbf{e}_{W:2} & \mathbf{e}_{B:3} \cdot \mathbf{e}_{W:3} \end{bmatrix} \begin{bmatrix} \mathbf{e}_{W:1} \\ \mathbf{e}_{W:2} \\ \mathbf{e}_{W:3} \end{bmatrix} . \quad (2.1)$$

The coordinates of a vector change under a passive transformation. For a WS vector \mathbf{a}_W , the coordinates transform as

$$\begin{aligned} (a_{B:i} \mathbf{e}_{B:i}) \cdot \mathbf{e}_{B:k} &= (a_{W:i} \mathbf{e}_{W:i}) \cdot \mathbf{e}_{B:k} \\ a_{B:i} \delta_{ik} &= (a_{W:i} \mathbf{R}_{B:ij}^W \mathbf{e}_{B:j}) \cdot \mathbf{e}_{B:k} \\ a_{B:k} &= a_{W:i} \mathbf{R}_{B:ij}^W \delta_{jk} \\ a_{B:k} &= a_{W:i} \mathbf{R}_{B:ik}^W \end{aligned} \quad , \quad (2.2)$$

where $\mathbf{R}_B^W \equiv (\mathbf{R}_W^B)^{-1}$. The active transformation is different in the sense that the the basis is left fixed but the vectors are transformed. Under an active transformation, the coordinates transform as

$$\begin{aligned} (a_{B:i} \mathbf{e}_{W:i}) \cdot \mathbf{e}_{W:k} &= (a_{W:i} \mathbf{e}_{B:i}) \cdot \mathbf{e}_{W:k} \\ a_{B:i} \delta_{ik} &= (a_{W:i} (\mathbf{R}_{W:ij}^B \mathbf{e}_{W:j})) \cdot \mathbf{e}_{W:k} \\ a_{B:k} &= a_{W:i} \mathbf{R}_{W:ik}^B \end{aligned} \quad . \quad (2.3)$$

The active transformation actually changes the physical location of a point or rigid body. A passive transformation changes the coordinate system in which the object is described. These two types of transformation are used throughout this thesis. They are very important and are extensively used when describing rigid body dynamics. Usually the dynamics of a rigid body are most easily described in a world space coordinate system. On the other hand, the dynamics of a deforming body are often much more easily described in the body space. However, there are many situations where one needs a representation of these dynamics in the world space. Take for example the description of an active swimming particle whose surface deforms with non-reciprocal motion in time. Usually one either knows or explicitly defines a surface parameterization, $\mathbf{x}(t)$, in a body frame that is attached to the active particle at some world space location $\mathbf{x}_c(t)$. The WS position vector of a point on the body's surface is then almost always most conveniently described by

$$x(t)_{W:i} \mathbf{e}_{W:i} = x(t)_{c:i} \mathbf{e}_{W:i} + x(t)_{B:i} \mathbf{e}(t)_{B:i} \quad . \quad (2.4)$$

However, Eqn. 2.2 explicitly tells us how to write this expression completely in the world frame,

$$x(t)_{W:i} \mathbf{e}_{W:i} = x(t)_{c:i} \mathbf{e}_{W:i} + (\mathbf{R}_{W:ik}^B(t) x(t)_{B:k}) \mathbf{e}_{W:i} \quad . \quad (2.5)$$

Equations 2.4 and 2.5 must yield the same point $\mathbf{x}(t)$, and are therefore useful for debugging numerical code and in producing and debugging visualizations. Conventions describing particle locations in the world space and body space are shown in Fig. 2.1.

2.2 Rigid Body Dynamics

Simulating the motion of a rigid body is almost the same as simulating the motion of a particle. Therefore, we will start by understanding particle simulation. Since we are interested in understanding dynamic evolution in time, the spatial variable \mathbf{x} is now identified as being a function of time, $\mathbf{x}(t)$ such that $\mathbf{x} : \mathbb{R} \rightarrow \mathbb{R}^3$. This position is in world space and can be written more explicitly as

$$\mathbf{x}(t) = x_i(t)\mathbf{e}_{W:i} \quad , \quad (2.6)$$

where the basis vectors $\{\mathbf{e}_{W:1}, \mathbf{e}_{W:2}, \mathbf{e}_{W:3}\}$ span the space that all particles occupy during the simulation. The function $\mathbf{U}(t) = \dot{\mathbf{x}}(t)$ is identified as the velocity, also in world space. This can be written more explicitly as

$$\mathbf{U}(t) = \dot{x}_i(t)\mathbf{e}_{W:i} \quad . \quad (2.7)$$

For notational purposes, it is convenient to group these two quantities into a stacked state vector, \mathbf{X} ,

$$\mathbf{X}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{pmatrix} \quad . \quad (2.8)$$

Normally we also need to know the force acting on the particle at time t . We identify this force as $\mathbf{F}(t)$, also in world space. If the particle has constant mass m , then the change of $\mathbf{X}(t)$ over time is given by

$$\dot{\mathbf{X}}(t) = \begin{pmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{u}}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{u}(t) \\ \mathbf{F}(t)/m \end{pmatrix} \quad . \quad (2.9)$$

Given some $\mathbf{X}(t)$, we see that Eqn. 2.9 describes the evolution of $\mathbf{X}(t)$ over time. The force on a particle at the k 'th discrete simulation time t_k is generally obtained from the numerical solution of differential equations based on Newton's second law. State variable evolution is obtained by evolving the particle position using the known force at a series of discrete times. Simulating rigid bodies proceeds in almost exactly the same way except $\mathbf{X}(t)$ now holds more information, specifically related to the body's size.

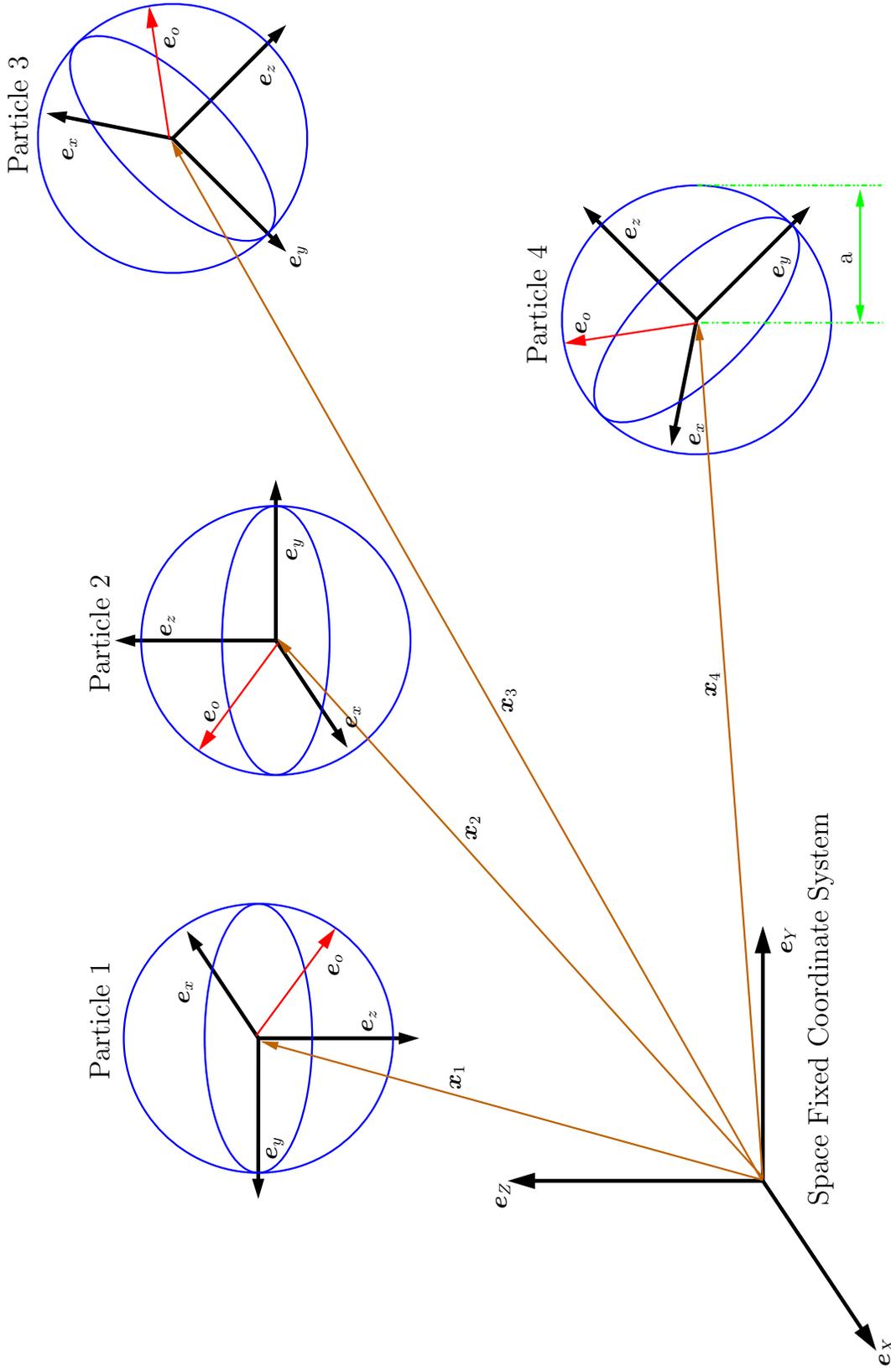


Fig. 2.1 The world space (space fixed) and body space frames of reference are shown for a group of spherical particles in a 3D computation cell. Also shown is the formalism of vector notation for particle position, $\mathbf{x}(t)$, and orientation $\mathbf{e}_o(t)$.

2.3 Rigid Body Position and Orientation

To locate a rigid body in world space we will use the vector $\mathbf{x}(t)$ which will describe the translation of the body. We must also describe the rotation of the body. For now we use a 3×3 rotation matrix $\mathbf{R}(t)$. To simplify this construction we require that the description of the body in the body space be such that the center of mass of the body lies at the body space origin, $(0, 0, 0)$. This identifies the world-space location of the center of mass as $\mathbf{x}(t) \equiv \mathbf{x}_c(t)$. For a point \mathbf{x}_B in the body space, the world space location, \mathbf{x}_W , is given as a result of rotating \mathbf{x}_B and then translating it by \mathbf{x}_c . For rigid bodies, this is an affine transform and is given as

$$x_i(t)\mathbf{e}_{W:i} = x_{c:i}(t)\mathbf{e}_{W:i} + \left(\mathbf{R}_{W:ik}^B(t)x_{B:k}(t)\right)\mathbf{e}_{W:i}(t) \quad . \quad (2.10)$$

A simple interpretation of $\mathbf{R}_{W:ik}^B(t)$ can be obtained if one notes that $\mathbf{e}_{B:i}(t)$ is equal to the i 'th column of $\mathbf{R}_{W:ik}^B(t)$. Thus the columns of $\mathbf{R}_{W:ik}^B(t)$ give the set of coefficients of the $\mathbf{e}_{W:i}$ basis vectors which explicitly represents the direction that the rigid body's axes point in when transformed to world space at time t . In what follows we refer to $\mathbf{x}_c(t)$ and $\mathbf{R}_{W:ik}^B(t) \equiv \mathbf{R}(t)$ as the position and orientation of the body at time t .

2.4 Linear Velocity

We now define how the position and orientation change over time. Expressions are needed for $\dot{\mathbf{x}}_c(t)$ and $\dot{\mathbf{R}}(t)$. We define the linear velocity $\mathbf{U}(t)$ as

$$\mathbf{U}(t) = \dot{\mathbf{x}}_c(t) \quad . \quad (2.11)$$

If the orientation of the body is fixed, then the only movement the body can undergo is pure translation. The quantity $\mathbf{U}(t)$ gives the velocity of this translation. In this thesis we continually refer to the linear velocity as the *translational velocity* or as a component of the body's *rigid body motion*.

2.5 Angular Velocity

In addition to this translation, the body can also spin. If we freeze the position of the center of mass in space, any movement of points on the body must be due to the body spinning about some axis that passes through the center of mass. This spin can be described by the vector $\boldsymbol{\Omega}(t)$ and we call this the body's angular velocity. The direction of $\boldsymbol{\Omega}(t)$ gives the direction of the axis about which the body is spinning. Its

magnitude tells how fast the body is spinning in rev/time. Thus $|\boldsymbol{\Omega}(t)|$ relates the angle through which the body will rotate over a given period of time if the angular velocity is constant.

How then are $\mathbf{R}(t)$ and $\boldsymbol{\Omega}(t)$ related? We have already understood the columns of $\mathbf{R}(t)$ as being the world space representation of the body space spanning vectors. Thus $\dot{\mathbf{R}}(t)$ intuitively represents the velocity at which these vectors change their directions in the world space. To gain further insight, we consider a vector in world space that is translating with the body,

$$\mathbf{r}(t) = \mathbf{x}(t) - \mathbf{x}_c(t) \quad . \quad (2.12)$$

We can decompose $\mathbf{r}(t)$ into a sum of two vectors \mathbf{a} and \mathbf{b} that are parallel and perpendicular to $\boldsymbol{\Omega}$. Since this vector translates with the body it is independent of \mathbf{U} . Then the instantaneous velocity of $\mathbf{r}(t)$ must be perpendicular to both \mathbf{b} and $\boldsymbol{\Omega}$. Thus one may write $\dot{\mathbf{r}}(t) = \boldsymbol{\Omega}(t) \times \mathbf{b}$. However, if one notes that $\mathbf{r}(t) = \mathbf{a} + \mathbf{b}$ and that $\mathbf{a} \parallel \boldsymbol{\Omega}(t) \implies \boldsymbol{\Omega} \times \mathbf{a} = 0$, then

$$\dot{\mathbf{r}}(t) = \boldsymbol{\Omega} \times \mathbf{b} = \boldsymbol{\Omega} \times (\mathbf{b} + \mathbf{a}) = \boldsymbol{\Omega} \times \mathbf{r}(t) \quad . \quad (2.13)$$

From Eqn. 2.13 we see that any world space position vector of a point on the body transforms as $\boldsymbol{\Omega}(t) \times (\cdot)$. We just claimed that the world space representation of the body axes are the columns of $\mathbf{R}(t)$ and hence $\dot{\mathbf{R}}(t)$ can be written as,

$$\dot{\mathbf{R}}(t) = \left(\boldsymbol{\Omega} \times \mathbf{R}_{i1}(t), \boldsymbol{\Omega} \times \mathbf{R}_{i2}(t), \boldsymbol{\Omega} \times \mathbf{R}_{i3}(t) \right) \quad . \quad (2.14)$$

This $\boldsymbol{\Omega}$ can be factored out and we find

$$\begin{aligned} \dot{\mathbf{R}}(t) &= \boldsymbol{\Omega}^*(t) \mathbf{R}(t) \\ \boldsymbol{\Omega}^* &= [\epsilon_{ijk} \Omega_j]_{3 \times 3} \end{aligned} \quad . \quad (2.15)$$

2.6 Velocity of a Particle (or point)

If we now proceed to differentiate Eqn. 2.10, we find that

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \dot{\mathbf{R}}(t) \mathbf{x}_B(t) + \mathbf{R}(t) \dot{\mathbf{x}}_B(t) + \dot{\mathbf{x}}_c(t) \\ &= \boldsymbol{\Omega}^*(t) \mathbf{R}(t) \mathbf{x}_B(t) + \mathbf{0} + \mathbf{U}(t) \end{aligned} \quad , \quad (2.16)$$

where we note that $\dot{\mathbf{x}}_B = \mathbf{0}$ since the body is assumed to be rigid. We can rewrite this as

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \Omega^*(t) (\mathbf{R}(t)\mathbf{x}_B(t) + \mathbf{x}_c(t) - \mathbf{x}_c(t)) + \mathbf{U}(t) \\ &= \Omega^*(t) (\mathbf{x}(t) - \mathbf{x}_c(t)) + \mathbf{U}(t) \\ &= \boldsymbol{\Omega}(t) \times (\mathbf{x}(t) - \mathbf{x}_c(t)) + \mathbf{U}(t)\end{aligned}\quad . \quad (2.17)$$

We see that Eqn. 2.17 separates the velocity of a point on a rigid body into two components: a linear component $\mathbf{U}(t)$, and an angular component $\boldsymbol{\Omega}(t) \times (\mathbf{x}(t) - \mathbf{x}_c(t))$. This is the normal usage most authors jump to when specifying boundary-velocity conditions in fluid mechanics problems. However, the interpretation shown here is much richer and shows how angular velocity is connected to the time derivatives of a body's rotation matrix (or orientation). Especially when studying or modeling actively swimming or deforming particles, $\dot{\mathbf{x}}_B \neq \mathbf{0}$, and this term must be accounted for in Eqn. 2.17. In later chapters, we shall see that a nonzero body space deformation function gives rise to a mathematical mechanism by which microorganisms and other particles may be understood to generate propulsive motion in Stokes flow.

2.7 Centroids and Centers of Mass

The primary purpose of defining the centroid or center of mass of a body is so that we may separate the dynamics of bodies into linear and angular components. The geometric centroid of a body in world space is defined to be,

$$\mathbf{x}_c(t) = \frac{\int_V \mathbf{x}(t) dV(\mathbf{x})}{\int_V dV(\mathbf{x})} \quad . \quad (2.18)$$

In what follows, quantities solely dependent on the geometry of the object are differentiated from their mass quantity counter parts under the conversion,

$$\int_V (\cdot) dV(\mathbf{x}) \rightarrow \int_V \rho(\mathbf{x}, t) (\cdot) dV(\mathbf{x}) \quad . \quad (2.19)$$

An interesting and useful alternative formula follows by using the divergence theorem to convert the volume integrals to surface integrals. We note the following conversions:

$$\begin{aligned}\nabla \cdot \left(\frac{\mathbf{x} \cdot \mathbf{x}}{2} \right) &= \mathbf{x} \\ \int_V \mathbf{x} dV(\mathbf{x}) &= \int_V \nabla \cdot \left(\frac{\mathbf{x} \cdot \mathbf{x}}{2} \right) dV(\mathbf{x}) = \frac{1}{2} \int_{\Gamma} x^2 \mathbf{n}(\mathbf{x}) dS(\mathbf{x}) \\ \nabla \cdot \frac{\mathbf{x}}{3} &= 1 \\ \int_V dV(\mathbf{x}) &= \frac{1}{3} \int_V \nabla \cdot \mathbf{x} dV(\mathbf{x}) = \frac{1}{3} \int_{\Gamma} \mathbf{x} \cdot \mathbf{n}(\mathbf{x}) dS(\mathbf{x})\end{aligned}\tag{2.20}$$

Thus we find that Eqn. 2.18 rewritten in terms of surface integrals becomes

$$\mathbf{x}_c(t) = \frac{3}{2} \frac{\int_{\Gamma} x^2 \mathbf{n}(\mathbf{x}) dS(\mathbf{x})}{\int_{\Gamma} \mathbf{x} \cdot \mathbf{n}(\mathbf{x}) dS(\mathbf{x})}\tag{2.21}$$

By a center of mass coordinate system we mean that in body space,

$$\frac{\int_V \mathbf{x}_B(t) dV(\mathbf{x})}{\int_V dV(\mathbf{x})} = \mathbf{0}\tag{2.22}$$

Another useful equation follows from Eqn. 2.22,

$$\int_V (\mathbf{x}(t) - \mathbf{x}_c(t)) dV(\mathbf{x}) = \int_V (\mathbf{R}(t) \mathbf{x}_B + \mathbf{x}_c - \mathbf{x}_c) dV(\mathbf{x}) = \mathbf{R}(t) \int_V \mathbf{x}_B dV(\mathbf{x}) = \mathbf{0}\tag{2.23}$$

or, from the conversions in Eqn. 2.20, that

$$\int_{\Gamma} r^2(t) \mathbf{n}(\mathbf{x}) dS(\mathbf{x}) = \mathbf{0}\tag{2.24}$$

All of the formulas shown in this section are highly useful in debugging numerical algorithms and discrete differential geometry code. These identities must at least approximately hold on discrete (triangulated) versions of 2-manifolds and should converge as the discrete resolution of the problem is increased (triangles are refined).

2.8 Force and Torque

For a particle or point that makes up a rigid body we can imagine an external force $\mathbf{F}_i(t)$ that denotes the total external force acting on the i 'th particle at time t . The total external torque on this particle can then be defined as

$$\mathbf{T}_i(t) = (\mathbf{x}(t) - \mathbf{x}_c(t)) \times \mathbf{F}_i(t)\tag{2.25}$$

These concepts are generalized to rigid bodies by integrating over the body volume,

$$\mathbf{F}(t) = \int_V \mathbf{f}_V(\mathbf{x}, t) dV(\mathbf{x}) \quad (2.26)$$

$$\mathbf{T}(t) = \int_V (\mathbf{x}(t) - \mathbf{x}_c(t)) \times \mathbf{f}_V(\mathbf{x}, t) dV(\mathbf{x}) \quad , \quad (2.27)$$

where $\mathbf{f}_V(t)$ is a force density (i.e. a force per unit volume). We note that in general, for uniform external forces, $\mathbf{F}(t)$ will convey no information about where the forces are acting on the body. However, $\mathbf{T}(t)$ does tell us about this distribution. Equations 2.26 and 2.27 can be converted to surface integrals using the divergence theorem, where it can be assumed that $\mathbf{f}_V(t)$ can be represented as $\mathbf{f}_V(t) = \nabla_{\mathbf{x}} \cdot \sigma(\mathbf{x}, t)$ and

$$\mathbf{F}(t) = \int_S \sigma(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) dS(\mathbf{x}) \quad (2.28)$$

$$\mathbf{T}(t) = \int_S (\mathbf{x}(t) - \mathbf{x}_c(t)) \times (\sigma(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x})) dS(\mathbf{x}) \quad . \quad (2.29)$$

In coming chapters, $\sigma(\mathbf{x}, t)$ will be described by the Newtonian stress tensor, and Eqns. 2.28 and 2.29 will be used as integral constraints on the hydrodynamic force and torque that a body or particle may experience.

2.9 Linear Momentum

The total linear momentum simplifies in a straightforward manner. This is because we are using a center of mass coordinate system where we can apply Eqn. 2.23. The linear momentum of our rigid body becomes

$$\begin{aligned} \mathbf{P}(t) &= \int_V \rho \dot{\mathbf{x}}(t) dV(\mathbf{x}) \\ &= \int_V \rho (\boldsymbol{\Omega}(t) \times (\mathbf{x}(t) - \mathbf{x}_c(t)) + \mathbf{U}(t)) dV(\mathbf{x}) \\ &= \int_V \rho \mathbf{U}(t) dV(\mathbf{x}) + \boldsymbol{\Omega}(t) \times \int_V \rho (\mathbf{x}(t) - \mathbf{x}_c(t)) dV(\mathbf{x}) \quad . \quad (2.30) \\ &= \int_V \rho \mathbf{U}(t) dV(\mathbf{x}) \\ &= m\mathbf{U}(t) \end{aligned}$$

2.10 Angular Momentum

Angular momentum is more unintuitive. Angular momentum is conserved in nature but angular velocity is not. This means that if we have a body floating through space with no net torque acting on it, its angular momentum would be constant.

However, the angular velocity may not be constant. A body's angular velocity can vary even when no force acts on the body. The angular momentum of a body is defined as $\mathbf{L}(t) = I(t)\boldsymbol{\Omega}(t)$ where $I(t)$ is a rank 2 tensor called the inertia tensor. This tensor describes how the mass of a body is distributed relative to its center of mass. The tensor depends on the orientation of the body but not on the body's absolute translation. Angular momentum and total torque are related by $\dot{\mathbf{L}}(t) = \mathbf{T}(t)$ analogous to the relation $\dot{\mathbf{P}}(t) = \mathbf{F}(t)$.

2.11 Inertia Tensor

The inertia tensor scales the angular momentum, $\mathbf{L}(t)$ and the angular velocity $\boldsymbol{\Omega}(t)$. The tensor is expressed in terms of $\mathbf{r}(t) = \mathbf{x}(t) - \mathbf{x}_c(t)$ as the symmetric matrix,

$$I(t) = \begin{pmatrix} \int_V \rho(r_k r_k - r_1^2) dV(\mathbf{x}) & - \int_V \rho r_1 r_2 dV(\mathbf{x}) & - \int_V \rho r_1 r_3 dV(\mathbf{x}) \\ - \int_V \rho r_2 r_1 dV(\mathbf{x}) & \int_V \rho(r_k r_k - r_2^2) dV(\mathbf{x}) & - \int_V \rho r_2 r_3 dV(\mathbf{x}) \\ - \int_V \rho r_3 r_1 dV(\mathbf{x}) & - \int_V \rho r_3 r_2 dV(\mathbf{x}) & \int_V \rho(r_k r_k - r_3^2) dV(\mathbf{x}) \end{pmatrix} \quad (2.31)$$

Or by the equivalent element-element relation,

$$I_{ij}(t) = \int_V \rho(\delta_{ij} r_k r_k - r_i r_j) dV(\mathbf{x}) \quad . \quad (2.32)$$

One can develop this expression and find that the inertia tensor in the body space, I_B , is given by

$$I_B = \int_V \rho(\mathbf{x}_B \cdot \mathbf{x}_B \mathbb{I} - \mathbf{x}_B \mathbf{x}_B) dV(\mathbf{x}) \quad . \quad (2.33)$$

From which it follows that

$$I(t) = \mathbf{R}(t) I_B \mathbf{R}(t)^T \quad . \quad (2.34)$$

Since I_B is specified in the body-space, it is constant throughout the simulation. It is pre-computed and then one can easily compute $I(t)$ from I_B and $\mathbf{R}(t)$. The inverse of $I(t)$ is given by

$$\begin{aligned} I^{-1}(t) &= \left(\mathbf{R}(t) I_B \mathbf{R}(t)^T \right)^{-1} \\ &= \left(\mathbf{R}(t)^T \right)^{-1} I_B^{-1} \mathbf{R}(t)^{-1} \quad . \\ &= \mathbf{R}(t) I_B^{-1} \mathbf{R}(t)^T \end{aligned} \quad (2.35)$$

Since for rotation matrices it holds that $\mathbf{R}(t)^T = \mathbf{R}(t)^{-1}$ and $(\mathbf{R}(t)^T)^T = \mathbf{R}(t)$. Note also that I_B^{-1} remains constant during the simulation.

2.12 Rigid Body Equations of Motion

The stacked state vector $\mathbf{X}(t)$ for the rigid body becomes

$$\mathbf{X}(t) = \begin{pmatrix} \mathbf{x}_c(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} . \quad (2.36)$$

Thus, the state of a rigid body is its position and orientation (describing spatial information), and its linear and angular momentum (describing velocity information). We know m and I_B at the beginning of the simulation and can compute the auxiliary quantities $I(t)$, $\boldsymbol{\Omega}(t)$, and $\mathbf{U}(t)$ via

$$\mathbf{U}(t) = \frac{\mathbf{P}(t)}{m}, \quad I(t) = \mathbf{R}(t)I_B\mathbf{R}(t)^T, \quad \boldsymbol{\Omega}(t) = I(t)^{-1}\mathbf{L}(t) . \quad (2.37)$$

The derivative of the state vector $\dot{\mathbf{X}}(t)$ is then

$$\dot{\mathbf{X}}(t) = \begin{pmatrix} \mathbf{U}(t) \\ \boldsymbol{\Omega}^*(t)\mathbf{R}(t) \\ \mathbf{F}(t) \\ \mathbf{T}(t) \end{pmatrix} . \quad (2.38)$$

2.13 Rigid Body Dynamics Algorithm

The rigid body dynamics algorithm proceeds by the following steps:

1. Initialization:

- (a) Determine Body Constants: m , I_B , and I_B^{-1}
- (b) Determine Initial Conditions: $\mathbf{X}(t = 0)$
- (c) Compute Initial Auxiliary Quantities: $I(0) = \mathbf{R}(0)I_B\mathbf{R}(0)$, $\mathbf{U}(0) = \mathbf{P}(0)/m$,
 $\boldsymbol{\Omega}(0) = I^{-1}(0)\mathbf{L}(0)$

2. Simulation, $t > 0$:

- (a) Compute Individual Forces and Application Points: \mathbf{F}_i , \mathbf{r}_i
- (b) Compute Total Forces and Torques: $\mathbf{F}(t) = \sum_i \mathbf{F}_i(t)$, $\mathbf{T}(t) = \sum_i \mathbf{r}_i \times \mathbf{F}_i(t)$

(c) Integrate State Vector Quantities:

$$\mathbf{X}(t+1) = \mathbf{X}(t) + \dot{\mathbf{X}}(t)\Delta t$$

(d) Re-orthogonalize $\mathbf{R}(t)$ (e.g. using modified Gram-Schmidt)

(e) Compute Auxiliary Quantities: $I(t+1) = \mathbf{R}(t+1)I_B\mathbf{R}(t+1)$, $\mathbf{U}(t+1) = \mathbf{P}(t+1)/m$, $\boldsymbol{\Omega}(t+1) = I(t+1)^{-1}\mathbf{L}(t+1)$

(f) Update Geometry:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t \\ &= \mathbf{x}(t) + [\boldsymbol{\Omega}(t) \times (\mathbf{x}(t) - \mathbf{x}_c(t)) + \mathbf{U}(t)] \Delta t \end{aligned} \quad (2.39)$$

In practice this algorithm is much more stable if $\mathbf{R}(t)$ is represented by a unit quaternion $q(t)$ and this quaternion is integrated in time and appropriately re-orthonormalized.

2.14 Quaternion Representation of $\mathbf{R}(t)$

In practice $\mathbf{R}(t)$ experiences a fair amount of numerical drift. The post re-orthogonalization step also requires an additional algorithm (e.g. modified Gram-Schmidt). It proves useful to use a quaternion representation of $\mathbf{R}(t)$. A quaternion is an ordered pair $[s, \mathbf{v}]$ such that $\mathbf{q} = s + v_i \mathbf{e}_{W:i}$. Quaternion multiplication follows the rule,

$$\begin{aligned} \mathbf{q}_1 \mathbf{q}_2 &= [s_1, \mathbf{v}_1][s_2, \mathbf{v}_2] \\ &= [s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2] \end{aligned} \quad (2.40)$$

A unit quaternion represents a rotation matrix. The evolution formula for $\mathbf{q}(t)$ is

$$\dot{\mathbf{q}}(t) = \frac{1}{2}[0, \boldsymbol{\Omega}(t)]\mathbf{q}(t) \quad (2.41)$$

It also holds that the $\mathbf{q}(t) \rightarrow \mathbf{R}(t)$ conversion follows,

$$[\mathbf{q}(t) = [s, \mathbf{v}]] \rightarrow \left[\mathbf{R}(t) = \begin{pmatrix} 1 - 2\mathbf{v}_2^2 - 2\mathbf{v}_3^2 & 2\mathbf{v}_1\mathbf{v}_2 - 2s\mathbf{v}_3 & 2\mathbf{v}_1\mathbf{v}_3 - 2s\mathbf{v}_2 \\ 2\mathbf{v}_1\mathbf{v}_2 - 2s\mathbf{v}_3 & 1 - 2\mathbf{v}_1^2 - 2\mathbf{v}_3^2 & 2\mathbf{v}_2\mathbf{v}_3 - 2s\mathbf{v}_1 \\ 2\mathbf{v}_1\mathbf{v}_3 - 2s\mathbf{v}_2 & 2\mathbf{v}_2\mathbf{v}_3 - 2s\mathbf{v}_1 & 1 - 2\mathbf{v}_1^2 - 2\mathbf{v}_2^2 \end{pmatrix} \right] \quad (2.42)$$

When using a quaternion representation in the rigid body dynamics algorithm, $\mathbf{q}(t)$ replaces $\mathbf{R}(t)$ as a state variable and $\mathbf{R}(t)$ is computed as an auxiliary quantity following Eqn. 2.42.

The norm of a quaternion is defined by $\mathbf{q}\mathbf{q}^c$ where $\mathbf{q}^c = [s, -\mathbf{v}]$. For a unit quaternion \mathbf{q} , the mapping of a vector from body space to world space obeys the relation,

$$\mathbf{x}_W = \mathbf{q}\mathbf{x}_B\mathbf{q}^c \quad , \quad (2.43)$$

where $\mathbf{x}_W = [0, \mathbf{x}_W]$, $\mathbf{x}_B = [0, \mathbf{x}_B]$, and $\mathbf{q} = \mathbf{q}(t)$ represents $\mathbf{R}(t)$.

2.15 Updating the Rigid Body Mesh

It is much more efficient to update the mesh geometry by computing an overall affine transformation matrix describing the net rotation and translation of the body. This affine transformation matrix is the composition of a translation back to last known center of mass $\mathbf{x}_c(t)$, inverse rotation $\mathbf{R}^{-1}(t)$, a rotation by $\mathbf{R}(t + \Delta t)$, and a translation to $\mathbf{x}_c(t + \Delta)$. This method which uses an overall transformation matrix avoids numerical surface distortions which would result if individual points were evolved separately.

2.16 Integration in Time

In dynamics problems, we have to solve an initial value problem which takes the vector form,

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{X}(0) = \mathbf{X}_0 \quad , \quad (2.44)$$

where $\mathbf{X}(t)$ is a time dependent vectorized state variable, subject to an initial condition \mathbf{X}_0 . In the context of Stokes flow with N_p rigid particles, this system of equations expands as

$$\frac{d}{dt} \begin{pmatrix} \mathbf{x}_c^1(t) \\ \mathbf{R}^1(t) \\ \vdots \\ \mathbf{x}_c^{N_p}(t) \\ \mathbf{R}^{N_p}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{U}^1(t, \mathbf{x}) \\ \boldsymbol{\Omega}^1(t, \mathbf{x}) \times \mathbf{R}^1(t) \\ \vdots \\ \mathbf{U}^{N_p}(t, \mathbf{x}) \\ \boldsymbol{\Omega}^{N_p}(t, \mathbf{x}) \times \mathbf{R}^{N_p}(t) \end{pmatrix} . \quad (2.45)$$

There are multiple methods to integrate Eqn. 2.45 ranging from fully explicit to IMEX and fully implicit schemes. Although one can directly evolve material points \mathbf{x} using pointwise evaluations of $\mathbf{u}(\mathbf{x})$, we want to avoid distortion in the shape of a rigid body, and so it is much better to compute \mathbf{U} and $\boldsymbol{\Omega}$ and then evolve \mathbf{x}_c and \mathbf{R} in time.

2.17 Linear Multistep Methods

Explicit Adams methods are called *Adams-Bashforth methods*. An s -step Adams-Bashforth method is $O(h^s)$ accurate for time step h . Adams-Bashforth methods up to order $s = 5$ are listed below:

$$y_{n+1} = y_n + hf(t_n, y_n), \quad (\text{This is the Euler method}) \quad (2.46)$$

$$y_{n+2} = y_{n+1} + h \left(\frac{3}{2}f(t_{n+1}, y_{n+1}) - \frac{1}{2}f(t_n, y_n) \right), \quad (2.47)$$

$$y_{n+3} = y_{n+2} + h \left(\frac{23}{12}f(t_{n+2}, y_{n+2}) - \frac{4}{3}f(t_{n+1}, y_{n+1}) + \frac{5}{12}f(t_n, y_n) \right), \quad (2.48)$$

$$y_{n+4} = y_{n+3} + h \left(\frac{55}{24}f(t_{n+3}, y_{n+3}) - \frac{59}{24}f(t_{n+2}, y_{n+2}) + \frac{37}{24}f(t_{n+1}, y_{n+1}) - \frac{3}{8}f(t_n, y_n) \right), \quad (2.49)$$

$$y_{n+5} = y_{n+4} + h \left(\frac{1901}{720}f(t_{n+4}, y_{n+4}) - \frac{1387}{360}f(t_{n+3}, y_{n+3}) + \frac{109}{30}f(t_{n+2}, y_{n+2}) \right) \quad (2.50)$$

$$- \frac{637}{360}f(t_{n+1}, y_{n+1}) + \frac{251}{720}f(t_n, y_n) \Big). \quad (2.51)$$

Implicit Adams methods are called *Adams-Moulton methods*. An s -step Adams-Moulton method is $O(h^{s+1})$ accurate for time step h . Adams-Moulton methods up to order $s = 5$ are listed below:

$$y_n = y_{n-1} + hf(t_n, y_n), \quad (\text{This is the backward Euler method}) \quad (2.52)$$

$$y_{n+1} = y_n + \frac{1}{2}h(f(t_{n+1}, y_{n+1}) + f(t_n, y_n)), \quad (\text{This is the trapezoidal rule}) \quad (2.53)$$

$$y_{n+2} = y_{n+1} + h \left(\frac{5}{12}f(t_{n+2}, y_{n+2}) + \frac{2}{3}f(t_{n+1}, y_{n+1}) - \frac{1}{12}f(t_n, y_n) \right), \quad (2.54)$$

$$y_{n+3} = y_{n+2} + h \left(\frac{3}{8}f(t_{n+3}, y_{n+3}) + \frac{19}{24}f(t_{n+2}, y_{n+2}) - \frac{5}{24}f(t_{n+1}, y_{n+1}) + \frac{1}{24}f(t_n, y_n) \right), \quad (2.55)$$

$$y_{n+4} = y_{n+3} + h \left(\frac{251}{720}f(t_{n+4}, y_{n+4}) + \frac{646}{720}f(t_{n+3}, y_{n+3}) - \frac{264}{720}f(t_{n+2}, y_{n+2}) \right) \quad (2.56)$$

$$+ \frac{106}{720}f(t_{n+1}, y_{n+1}) - \frac{19}{720}f(t_n, y_n) \Big). \quad (2.57)$$

Adams-Moulton methods are more expensive to compute, especially if evaluations of f are expensive. Since the method is implicit it requires an additional method for solving nonlinear equations iteratively (e.g. fixed-point iteration). This must be applied during each step.

An interesting alternative is to pair an Adams-Bashforth method with an Adams-Moulton method to obtain an *Adams-Moulton predictor-corrector method*. These

methods are normally abbreviated “PECE”, which stands for predict-evaluate-correct-evaluate. More generally the corrector may be run multiple times, giving the k PECE method commonly abbreviated as $P(EC)^kE$. The PECE method proceeds as follows:

1. Predict: Use the Adams-Bashforth method to compute a first approximation to y_{n+1} , which we call \tilde{y}_{n+1} .
2. Evaluate: Evaluate f at this value computing $f(t_{n+1}, \tilde{y}_{n+1})$.
3. Correct: Use the Adams-Moulton method to compute y_{n+1} where $f(t_{n+1}, \tilde{y}_{n+1})$ is used in place of $f(t_{n+1}, y_{n+1})$. This effectively turns the Adams-Moulton method into an explicit method.
4. Evaluate: Evaluate f at the newly computed value of y_{n+1} computing $f(t_{n+1}, y_{n+1})$ to use during the next time step.

PECE methods are viable alternatives to fully implicit schemes. To the authors knowledge, they have not been appropriately applied to fluid-structure interaction problems in Stokes flow.

2.18 Important Notes and Future Work

Because multi-step methods rely on previous time steps they cannot be used during the first $(s - 1)$ time steps because not enough data is available. A single step with the same order of accuracy must be used to compute enough starting values. A candidate one-step method is the Runge-Kutta method. For example, one may use a fourth order RK method to compute y_0, y_1, y_2 . Then an Adams-Bashforth three step method may be used to compute y_3 given y_0, y_1, y_2 .

In this thesis, all dynamics problems are solved with Euler integration. The reason for this is that the solution of the mobility problem for particles in linearized viscous flow (Stokes flow) is computationally demanding. Also, the use of linear explicit multi-step methods does not readily solve any of the problems in Stokes flow related to particle-particle collisions. For this, one at least needs adaptive resolution space. In other words, the Stokes equations must be solved on an adaptive mesh. Adaptivity in time is also necessary but is not considered here. In the coming chapters a scheme for adaptive spatial resolution is developed for solving the fluid-structure interaction problem in Stokes flow. Maintaining adaptivity in time remains an interesting research avenue. Adaptive time-stepping techniques such as Richardson extrapolation may provide appropriate solutions without significant modification.

Chapter 3

Boundary Integral Methods for Viscous Flows

3.1 The Stokes Equations

In this thesis all fluid mechanical calculations are derived from the Stokes equations. We present a derivation of these equations for completeness following [53]. We consider the flow of an incompressible Newtonian fluid. The motion of the fluid is governed by the continuity equation, $\nabla \cdot \mathbf{u} = 0$, which tells us about mass conservation, and the Navier-Stokes equations,

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \quad , \quad (3.1)$$

which simply express Newton's second law for a small volume element of fluid. In Eqn. 3.1 ρ and μ are the fluid density and viscosity, and \mathbf{g} is a body force (which we assume is constant). The Stokes equations may be non-dimensionalized using $\bar{\mathbf{u}} = \mathbf{u}/U$, $\bar{\mathbf{x}} = \mathbf{x}/L$, $\bar{t} = t/T$, and $\bar{p} = p/(\mu U/L)$, where we have identified the viscous pressure scale as $\mu U/L$. In dimensionless form, Eqn. 3.1 becomes

$$\beta \frac{\partial \bar{\mathbf{u}}}{\partial \bar{t}} + Re \bar{\mathbf{u}} \cdot \bar{\nabla} \bar{\mathbf{u}} = -\bar{\nabla} \bar{p} + \bar{\nabla}^2 \bar{\mathbf{u}} + \frac{Re}{Fr} \frac{\mathbf{g}}{|\mathbf{g}|} \quad . \quad (3.2)$$

Three dimensionless parameters appear in Eqn. 3.2, namely the frequency parameter $\beta = L^2/\nu T$, the Reynolds number $Re = UL/\nu$, and the Froude number $Fr = U^2/|\mathbf{g}|L$, where $\nu = \mu/\rho$ is the kinematic viscosity. The frequency parameter is a ratio between inertial acceleration forces relative to viscous forces. The Reynolds number is a ratio between the inertial convective forces and viscous forces. In this thesis we assume that

the both β and Re are small. Under these assumptions, the Navier Stokes equations reduce to the body-force-free Stokes equations,

$$-\nabla P(\mathbf{x}) + \mu \nabla^2 \mathbf{u}(\mathbf{x}) = \mathbf{0}, \quad \nabla \cdot \mathbf{u}(\mathbf{x}) = 0, \quad \mathbf{x} \in \tilde{\Omega} \quad , \quad (3.3)$$

where the body force has been combined with the pressure [53].

For a small Reynolds number and frequency parameter the Stokes equations model linearized viscous flows where the viscous forces are large compared with inertial forces. For colloidal dynamics problems, the particle length scale is often very small and the Stokes equations govern the relevant fluid mechanics. This is not to say that the fluid need be viscous, rather from the particle's perspective, the fluid is viscous over the particle's small length scale. Although the Stokes equations are linear and instantaneous, finding particle trajectories and or nodal trajectories of a surface mesh is in general difficult and numerically expensive because particle and fluid motions are coupled through specification of boundary conditions on the particle's surface.

3.2 Notation

Throughout this thesis, the following standard notation is used to describe vectors, tensors, and various fluid mechanics quantities:

Definition 1 ($\tilde{\Omega}$) *We are interested in solving the Stokes equations in a fluid domain, $\tilde{\Omega}$. In general $\tilde{\Omega} \subseteq \mathbb{R}^d$ where d is either two or three.*

Definition 2 ($\tilde{\Gamma}$) *$\tilde{\Gamma}$ is the boundary of the fluid domain $\tilde{\Omega}$.*

Definition 3 (Ω) *In general we can identify a closed domain Ω that generates the domain of the fluid, $\tilde{\Omega}$, and its boundary, $\tilde{\Gamma}$. In exterior problems we will identify $\tilde{\Omega} \equiv \mathbb{R}^d \setminus \bar{\Omega}$. In interior problems $\tilde{\Omega} \equiv \Omega$. In words, we have identified the domain closure with its normal topological definition, where we allow for points on the boundary and have taken the interior to mean Ω .*

Definition 4 (Γ_∞) *When the domain is infinite, i.e. in the exterior Stokes problem, the boundary $\tilde{\Gamma} = \Gamma \cup \Gamma_\infty$ where $\Gamma_\infty \equiv \lim_{R \rightarrow \infty} \partial B_R(0)$, and $B_R(0)$ is the ball of radius R centered at the origin (in world space).*

Definition 5 ((\mathbf{x}, \mathbf{y}) pairing) *$(\mathbf{x}, \mathbf{y}) \equiv \mathbf{y} - \mathbf{x} \equiv \mathbf{r}$. Note that \mathbf{y} is synonymous with a "Field Point", sometimes labeled as \mathbf{Q} , and \mathbf{x} is synonymous with a "Source Point", sometimes labeled \mathbf{P} .*

Definition 6 (*r* distance vector) $r \equiv |\mathbf{r}| \equiv |\mathbf{y} - \mathbf{x}| \equiv |\mathbf{x} - \mathbf{y}|$

Definition 7 (*r_i* component derivative of *r* w.r.t. *y*) $r_{,i} \equiv (y_i - x_i)/r \equiv (\mathbf{y} - \mathbf{x})_i/r$

Definition 8 (*G_{ij}*, Single Layer Potential)

$$G_{ij} \equiv \frac{1}{r} (\delta_{ij} + r_{,i} r_{,j}) \quad (3.4)$$

Definition 9 (*G_{ij}* symmetry) $G_{ij}(\mathbf{x}, \mathbf{y}) \equiv G_{ji}(\mathbf{x}, \mathbf{y})$

Definition 10 (*K_{ij}* double layer potential)

$$K_{ij}(\mathbf{x}, \mathbf{y}) \equiv T_{ijk}(\mathbf{x}, \mathbf{y}) n_k(\mathbf{y}) \equiv -\frac{6}{r^2} (r_{,i} r_{,j} r_{,k} n_k(\mathbf{y})) \quad (3.5)$$

Definition 11 (*K_{ij}* symmetry) $K_{ij}(\mathbf{x}, \mathbf{y}) \equiv K_{ji}(\mathbf{x}, \mathbf{y})$

Unless prefactors are explicitly given, we will absorb $1/8\pi\mu$ into G_{ij} , that is redefine $G_{ij} = (1/8\pi\mu)G_{ij}$, and also absorb $2/8\pi$ into K_{ij} , writing it as $K_{ij}(\mathbf{x}, \mathbf{y}) = -(2/3\pi)(r_{,i} r_{,j} r_{,k} n_k(\mathbf{y}))/r^2$. The additional factor of two is used to shift the eigenvalues of K to lie in $[-1, 1]$.

The general position of a fluid element or particle element is expressed using the vector \mathbf{y} and is referred to as the *field* position or *field* variable. The *source* points are expressed using the vector \mathbf{x} . The source and field points are interpreted based on how they are presented as arguments in the Green's function. The position of the center of mass of a particle is expressed as \mathbf{x}_c or $\mathbf{x}_{\{a-z, 1-N\}}$ (i.e. a sub-scripted \mathbf{x} where the particle label is the subscript). Unless it is specified, the particle orientation is expressed by a vector \mathbf{e}_o or \mathbf{e}_s . When writing out equations, the independent variables are explicitly expressed for each dependent field variable. If this dependence is obvious, this explicit indication will be omitted. A representation of a model system in this notation is shown in Fig. 2.1.

3.3 Singularity Solutions

Using our notation, we may consider the Stokes equations in free-space associated with a forcing term due to a Dirac mass centered at \mathbf{x} and weighted with the force vector \mathbf{g} . From this point of view the relevant equations are,

$$\begin{aligned}
-\nabla p(\mathbf{y}) + \mu \Delta \mathbf{u}(\mathbf{y}) &= \nabla \cdot \boldsymbol{\sigma} = -\mathbf{g} \delta(\mathbf{y} - \mathbf{x}), \quad \text{in } \mathbb{R}^d \\
\nabla \cdot \mathbf{u}(\mathbf{y}) &= 0 \\
\lim_{|\mathbf{y}| \rightarrow \infty} |\mathbf{u}(\mathbf{y})| &= 0, \quad \lim_{|\mathbf{y}| \rightarrow \infty} |p(\mathbf{y})| = 0 \\
\mathbf{u}(\mathbf{y}) &= \mathbf{g}_D \in H^{1/2}(\Gamma_D) \\
\mathbf{f}(\mathbf{y}) &= \mathbf{g}_N \in H^{-1/2}(\Gamma_N), \quad \forall \mathbf{y} \in \Gamma = \Gamma_D \cup \Gamma_N
\end{aligned} \tag{3.6}$$

Although we have written Eqn. 3.6 for the exterior problem, the interior problem is simply obtained by neglecting the decay conditions on \mathbf{u} and p . We note here that \mathbf{u} and p are the desired velocity and pressure fields of a viscous incompressible fluid flow in the domain \mathbb{R}^3 and \mathbf{f} is the hydrodynamic traction at the boundary defined by,

$$\mathbf{f}(\mathbf{y}) = \boldsymbol{\sigma}(\mathbf{u}(\mathbf{y}), p(\mathbf{y})) \cdot \mathbf{n}(\mathbf{y}) \quad , \tag{3.7}$$

where

$$\sigma_{ij}(\mathbf{u})(\mathbf{y}) = -p(\mathbf{y})\delta_{ij} + \mu \left(\frac{\partial u_i(\mathbf{y})}{\partial x_j} + \frac{\partial u_j(\mathbf{y})}{\partial x_i} \right) \quad , \tag{3.8}$$

is the Newtonian stress tensor. From equation system 3.6, G_{ij} is generated as the ‘‘Oseen-Burgers’’ tensor, also know as point force solution or Stokeslet, of Stokes flow. The stress tensor, T_{ijk} can be found by performing several manipulations of G_{ij} . The Stokeslet, G , and the pressure vector Π must satisfy the divergence free condition,

$$\frac{\partial G_{ij}(\mathbf{x}, \mathbf{y})}{\partial x_i} = 0 \quad , \tag{3.9}$$

and

$$-\frac{\partial \Pi_j(\mathbf{x}, \mathbf{y})}{\partial x_i} + \Delta_{\mathbf{y}} G_{ij}(\mathbf{x}, \mathbf{y}) = -\delta_{ij} \delta(\mathbf{y} - \mathbf{x}) \quad . \tag{3.10}$$

Also, let T_{ijk} be the stress tensor associated with G , having components,

$$T_{ijk}(\mathbf{x}, \mathbf{y}) \equiv T_{ijk}(\mathbf{y} - \mathbf{x}) = -\Pi_j(\mathbf{y} - \mathbf{x})\delta_{ik} + \frac{\partial G_{ij}(\mathbf{y} - \mathbf{x})}{\partial x_k} + \frac{\partial G_{kj}(\mathbf{y} - \mathbf{x})}{\partial x_i} \quad . \tag{3.11}$$

The tensors G , Π , and T are given by

$$G_{ij}(\mathbf{x}, \mathbf{y}) = \frac{\delta_{ij}}{|\mathbf{x} - \mathbf{y}|} + \frac{(y_i - x_i)(y_j - x_i)}{|\mathbf{x} - \mathbf{y}|^3} \quad (3.12)$$

$$\Pi_i(\mathbf{x}, \mathbf{y}) = 2 \frac{y_i - x_i}{|\mathbf{x} - \mathbf{y}|^3} \quad (3.13)$$

$$T_{ijk}(\mathbf{x}, \mathbf{y}) = -6 \frac{(y_i - x_i)(y_j - x_j)(y_k - x_k)}{|\mathbf{x} - \mathbf{y}|^5} \quad (3.14)$$

$$D_{ijkl}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{x} - \mathbf{y}|^3} ([-6(\delta_{ij}r_{,k} + \delta_{ik}r_{,j}) + 60r_{,i}r_{,j}r_{,k}]r_{,l} - 6(\delta_{jl}r_{,i}r_{,k} + \delta_{kl}r_{,i}r_{,j}) - 4\delta_{jk}\delta_{il}) \quad (3.15)$$

The pressure tensor $\Lambda = \Lambda_{ik}$ corresponds to the stress tensor, T , and satisfies

$$-\frac{\partial \Lambda_{ik}(\mathbf{y} - \mathbf{x})}{\partial y_j} + \Delta_{\mathbf{y}} T_{ijk}(\mathbf{y} - \mathbf{x}) = 0, \quad \text{for } \mathbf{x} \neq \mathbf{y} \quad , \quad (3.16)$$

where T satisfies the divergence free condition,

$$\frac{\partial T_{ijk}(\mathbf{y} - \mathbf{x})}{\partial x_k} = 0 \quad . \quad (3.17)$$

Λ is given by,

$$\Lambda_{ik}(\mathbf{y} - \mathbf{x}) = 4 \frac{\delta_{ik}}{|\mathbf{x} - \mathbf{y}|^3} - 12 \frac{(y_i - x_i)(y_k - x_k)}{|\mathbf{x} - \mathbf{y}|^5} \quad . \quad (3.18)$$

We note that the tensors G_{ij} , Π_i , and T_{ijk} generate the point force velocity, pressure, and stress solutions of Eqn. 3.6 upon contraction with the point force \mathbf{g} . These point force solutions are given by,

$$u_i(\mathbf{y}) = \frac{1}{8\pi\mu} G_{ij}(\mathbf{x}, \mathbf{y}) g_j \quad (3.19)$$

$$p(\mathbf{y}) = \frac{1}{8\pi} \Pi_i(\mathbf{x}, \mathbf{y}) g_i \quad (3.20)$$

$$\sigma_{ik}(\mathbf{y}) = \frac{1}{8\pi} T_{ijk}(\mathbf{x}, \mathbf{y}) g_j \quad (3.21)$$

There exists similar constructions using any singularity solution of the Stokes equations (half-space, rotlet, image systems, quadruple). However, higher order solutions become increasingly complicated and exhibit higher order singularities as $\mathbf{x} \rightarrow \mathbf{y}$ which tends to limit their use in numerical methods. The next useful higher order solution is the

stresslet construction which gives velocity, pressure, and stress fields,

$$u_i(\mathbf{y}) = \frac{1}{8\pi\mu} T_{ijk}(\mathbf{x}, \mathbf{y}) s_{jk} \quad (3.22)$$

$$p(\mathbf{y}) = \frac{\mu}{8\pi} \Lambda_{ij}(\mathbf{x}, \mathbf{y}) s_{ij} \quad (3.23)$$

$$\sigma_{il}(\mathbf{y}) = \frac{\mu}{8\pi} D_{ijkl}(\mathbf{x}, \mathbf{y}) s_{jk} \quad , \quad (3.24)$$

where $s_{ij} \equiv \phi_i n_j$.

3.3.1 Important Properties

From continuity it can be shown that

$$\int_{\Gamma} G_{ij}(\mathbf{x}, \mathbf{y}) n_i(\mathbf{x}) dS(\mathbf{y}) = 0 \quad , \quad (3.25)$$

and that

$$\frac{1}{8\pi} \int_{\Gamma} T_{ijk}(\mathbf{x}, \mathbf{y}) n_k(\mathbf{y}) dS(\mathbf{y}) = \begin{cases} \delta_{ij}, & \text{when } \mathbf{x} \text{ is inside } \Gamma \\ \frac{1}{2} \delta_{ij}, & \text{when } \mathbf{x} \text{ is on } \Gamma \\ 0, & \text{when } \mathbf{x} \text{ is outside } \Gamma \end{cases} . \quad (3.26)$$

Also it should be noted that (required by the symmetry of the stress tensor)

$$\begin{aligned} T_{ijk}(\mathbf{x}, \mathbf{y}) &= T_{kji}(\mathbf{x}, \mathbf{y}) = T_{jik}(\mathbf{x}, \mathbf{y}) = -T_{ijk}(\mathbf{y}, \mathbf{x}) \\ G_{ij}(\mathbf{x}, \mathbf{y}) &= G_{ji}(\mathbf{x}, \mathbf{y}) = G_{ij}(\mathbf{y}, \mathbf{x}) = G_{ji}(\mathbf{y}, \mathbf{x}) \quad . \end{aligned} \quad (3.27)$$

3.4 Lorentz reciprocal relation

Take two unrelated Newtonian flows with velocity fields \mathbf{u} and \mathbf{u}' , pressure fields p and p' and stress tensors σ and σ' . The counter part to Green's second identity can be written as [54]

$$u'_i \frac{\partial \sigma_{ij}}{\partial x_j} - u_i \frac{\partial \sigma'_{ij}}{\partial x_j} = \frac{\partial}{\partial x_j} (u'_i \sigma_{ij} - u_i \sigma'_{ij}) \quad . \quad (3.28)$$

If the flows satisfy the Stokes equation without a body force ($\nabla \cdot \sigma = \mathbf{0}$), then the *Lorentz reciprocal relationship* is immediately found,

$$\frac{\partial}{\partial x_j} (u'_i \sigma_{ij} - u_i \sigma'_{ij}) = 0 \quad . \quad (3.29)$$

It is important to note the following convention. In this thesis the *del* operator in an orthonormal basis is defined in the following way:

$$\nabla \circ (\cdot) \equiv \frac{\partial(\cdot)}{\partial x_j} \circ \mathbf{e}_j \quad , \quad (3.30)$$

where (\cdot) stands for an arbitrary differentiable tensor quantity (scalar, vector, higher-order tensor), and the \circ stands for any tensor product operation, for example the inner product, outer product, or cross product. The gradient, divergence, and curl can then be written as

$$\text{grad}(\cdot) \equiv \nabla \otimes (\cdot) = \nabla(\cdot), \quad \text{div}(\cdot) \equiv \nabla \cdot (\cdot), \quad \text{curl}(\cdot) \equiv -\nabla \times (\cdot) \quad . \quad (3.31)$$

From this we can see that the Eqn. 3.29, in vector form reads

$$\nabla \cdot (\mathbf{u}' \cdot \sigma - \mathbf{u} \cdot \sigma') = 0 \quad . \quad (3.32)$$

3.4.1 Boundary Integral Equations

Equation 3.28 can be evaluated using using an arbitrary Stokes flow (\mathbf{u}, σ) and a Stokes flow (\mathbf{u}', σ') associated with a point force. This yields,

$$\delta_3(\mathbf{y} - \mathbf{x})g_m u_m(\mathbf{x}) = \frac{\partial}{\partial x_j} \left(\frac{1}{8\pi\mu} G_{im}(\mathbf{x}, \mathbf{y})g_m \sigma_{ij} - \frac{1}{8\pi} u_i T_{imj}(\mathbf{x}, \mathbf{y})g_m \right) \quad . \quad (3.33)$$

The Lorentz reciprocal theorem is highly useful in the sense that one may attempt to express an arbitrary Stokes solution (\mathbf{u}, σ) in terms of a simpler solution (\mathbf{u}', σ') . This is how the boundary integral representation is constructed though the method is by no means limited to using an associated singularity solution (point force) as the *known* Stokes solution. For example, it would be interesting to attempt a representation where the Stokes flow (\mathbf{u}', σ') may be associated, to some extent, with slender body theory.

The following manipulations may be performed, all in one step: (1) discard the arbitrary constant body force g_m , (2) integrate over volume $\tilde{\Omega}$ for $\mathbf{x} \in \tilde{\Omega}$, and (3) convert the volume integrals to surface integrals over the boundary of $\tilde{\Omega}$ given by Γ ,

$$u_m(\mathbf{x}) = \frac{1}{8\pi\mu} \int_{\Gamma} G_{im}(\mathbf{x}, \mathbf{y})\sigma_{ij}(\mathbf{y})n_j(\mathbf{y})dS(\mathbf{y}) - \frac{1}{8\pi} \int_{\Gamma} u_i(\mathbf{y})T_{imj}(\mathbf{x}, \mathbf{y})n_j(\mathbf{y})dS(\mathbf{y}) \quad . \quad (3.34)$$

Now Eqn. 3.34 can be re-label by making $m \rightarrow j$, $j \rightarrow i$ and $i \rightarrow k$ to give,

$$u_j(\mathbf{x}) = \frac{1}{8\pi\mu} \int_{\Gamma} G_{ij}(\mathbf{x}, \mathbf{y}) \sigma_{ki}(\mathbf{y}) n_i(\mathbf{y}) dS(\mathbf{y}) - \frac{1}{8\pi} \int_{\Gamma} u_i(\mathbf{y}) T_{kji}(\mathbf{x}, \mathbf{y}) n_i(\mathbf{y}) dS(\mathbf{y}) \quad . \quad (3.35)$$

Using both of the properties from Eqn. 3.27 and noting that $\sigma_{ki}(\mathbf{y}) n_k(\mathbf{y}) = f_i(\mathbf{y})$, it can be shown that

$$u_j(\mathbf{x}) = \frac{1}{8\pi\mu} \int_{\Gamma} G_{ji}(\mathbf{x}, \mathbf{y}) f_i(\mathbf{y}) dS(\mathbf{y}) - \frac{1}{8\pi} \int_{\Gamma} u_i(\mathbf{y}) T_{ijk}(\mathbf{x}, \mathbf{y}) n_k(\mathbf{y}) dS(\mathbf{y}) \quad . \quad (3.36)$$

With additional careful mathematics, a BIE for three different situations can be derived, where $\mathbf{x} \in \tilde{\Omega}$, $\mathbf{x} \in \Gamma$, and $\mathbf{x} \in \tilde{\Omega}^c$. The final expression is

$$\frac{1}{8\pi\mu} \int_{\Gamma} G_{ji}(\mathbf{x}, \mathbf{y}) f_i(\mathbf{y}) dS(\mathbf{y}) - \frac{1}{8\pi} \int_{\Gamma} u_i(\mathbf{y}) T_{ijk}(\mathbf{x}, \mathbf{y}) n_k dS(\mathbf{y}) = \begin{cases} u_j(\mathbf{x}), & \mathbf{x} \in \tilde{\Omega} \\ \frac{1}{2}u_j(\mathbf{x}), & \mathbf{x} \in \Gamma \\ 0, & \text{else} \end{cases} \quad . \quad (3.37)$$

It is important to note the following two ideas, which reflect solution goals of Eqn. 3.37 in the context of fluid-structure interaction and in understanding particle dynamics:

1. **The Resistance Problem:** Using Eqn. 3.37, either the rigid body motions $\mathbf{U}, \mathbf{\Omega}$ or the velocity surface distribution $\{\mathbf{u}(\mathbf{y}), \forall \mathbf{y} \in \Gamma\}$ is given, and the total hydrodynamic force (or surface traction distribution) is solved for.
2. **The Mobility Problem:** Using Eqn. 3.37 either the total external force is known or the distribution of the surface traction is given, and the velocity distribution or rigid body motions are solved for.

In this thesis we formulate both of these problems in the context of the Galerkin Boundary Element discretization. We will solve the mobility problem in Chapter 5 for the dynamics of a particle and porous container.

3.4.2 Conventions for Normal Vectors and Interior / Exterior Problems

The normal vectors $\mathbf{n}(\mathbf{y})$ in Eqn. 3.37 point outward from the surface, Γ , and that the physics are being resolved inside of this surface, where there is viscous fluid. Even though the normals are exterior to the surface, Γ , this equation is understood to represent the *interior* flow problem. The same procedure can be use to derive the *exterior* flow problem where the fluid mechanics is resolved exterior to, for example,

particle surfaces. If this same construction were used for the *exterior* problem, the normal vectors would again point out of the fluid domain, and thus into the particle interiors (inwards into the particle bodies). However, in this thesis we use the geometric convention, which is that surface normals should always point outside of a closed geometric surface. This simply means that for the exterior problem, the normals need to be reversed. Thus the single layer and double layer integral contributions change mathematical sign. Some authors use the alternative convention that the normal vector always points into the fluid.

3.4.3 The Calderón Projection

The boundary integral equations and representations may be constructed for any linear elliptic partial differential equation(s). Today, there are even integral transform techniques and constructions for parabolic and hyperbolic PDEs. There are general ways to rigorously construct these representations, and they are always expressed as integrals over a surface distribution of singularity solutions. The general constructions for these representations are described in the excellent text by Steinbach, [64].

The construction shown in Eqn. 3.37 is only for one of the two BIEs that describes Stokes flow. A BIE for the traction field may also be formulated by taking appropriate derivatives of Eqn. 3.37 and using the Newtonian stress constitutive law. The factor of $1/2$ for $\boldsymbol{x} \in \Gamma$ arises only in the case where Γ is smooth and has a tangent plane defined everywhere. More generally it is expressed as a tensor C_{ij} . We do not derive the traction BIE here since this thesis does not use it. This additional traction BIE is potentially useful in Stokes flow problems when one must couple unknown surface tractions to constraints over velocity surface distributions. For example, when considering constant power squirmers or active particles a constraint must be made on the power or viscous dissipation as a function of a squirming parameter. Since power is force \times velocity, tractions depend on the squirmer's velocity surface distribution through a power integral constraint.

The velocity field and traction field boundary integral equations may be written in a more pedagogical and mathematically rigorous way. Following similar notation in

[64], the boundary integral equations of Stokes flow read,

$$C_{ij}(\mathbf{x})(\gamma_0^{\text{ext/int}} u_j)(\mathbf{x}) \pm \int_{\Gamma} K_{ij}(\mathbf{x}, \mathbf{y})(\gamma_0^{\text{ext/int}} u_j)(\mathbf{y}) dS(\mathbf{y}) \quad (3.38)$$

$$= \mp \int_{\Gamma} G_{ij}(\mathbf{x}, \mathbf{y})(\gamma_1^{\text{ext/int}} u_j)(\mathbf{y}) dS(\mathbf{y}) \quad (3.39)$$

$$C_{ij}(\mathbf{x})(\gamma_1^{\text{ext/int}} u_j)(\mathbf{x}) \pm \int_{\Gamma} K'_{ij}(\mathbf{x}, \mathbf{y})(\gamma_1^{\text{ext/int}} u_j)(\mathbf{y}) dS(\mathbf{y}) \quad (3.40)$$

$$= \mp \int_{\Gamma} D_{ij}(\mathbf{x}, \mathbf{y})(\gamma_0^{\text{ext/int}} u_j)(\mathbf{y}) dS(\mathbf{y}) \quad . \quad (3.41)$$

The boundary integral equation describing pressure is,

$$P(\mathbf{x}) \pm \int_{\Gamma} Q_i(\mathbf{x}, \mathbf{y}) u_i(\mathbf{y}) dS(\mathbf{y}) = \mp \int_{\Gamma} \Pi_i(\mathbf{x}, \mathbf{y}) t_i(\mathbf{y}) dS(\mathbf{y}) \quad . \quad (3.42)$$

In these equations, the integral kernels are given by,

$$\begin{aligned} K_{ij}(\mathbf{x}, \mathbf{y}) &= T_{ijk}(\mathbf{y} - \mathbf{x}) n_k(\mathbf{y}), & Q_i(\mathbf{x}, \mathbf{y}) &= \mu \Lambda_{ik}(\mathbf{y} - \mathbf{x}) n_k(\mathbf{y}) \\ K'_{ij}(\mathbf{x}, \mathbf{y}) &\equiv K_{ji}(\mathbf{y}, \mathbf{x}) = T_{jik}(\mathbf{x} - \mathbf{y}) n_k(\mathbf{x}) \end{aligned} \quad , \quad (3.43)$$

where all prefactors $1/8\pi\mu$, $1/8\pi$ and $1/4\pi$ have been respectively absorbed into the definitions of G , K , K' , D , Π and Q . The notions of the boundary trace and co-normal derivative trace operators are defined as,

$$\gamma_0^{\text{int}} \mathbf{u}(\mathbf{x}) = \lim_{\Omega \ni \tilde{\mathbf{x}} \rightarrow \mathbf{x} \in \Gamma} \mathbf{u}(\tilde{\mathbf{x}}) \quad (3.44)$$

$$\gamma_0^{\text{ext}} \mathbf{u}(\mathbf{x}) = \lim_{\mathbb{R}^d \setminus \bar{\Omega} \ni \tilde{\mathbf{x}} \rightarrow \mathbf{x} \in \Gamma} \mathbf{u}(\tilde{\mathbf{x}}) \quad (3.45)$$

$$\gamma_1^{\text{int}} \mathbf{u}(\mathbf{x}) = \lim_{\Omega \ni \tilde{\mathbf{x}} \rightarrow \mathbf{x} \in \Gamma} \sigma(\mathbf{u}, \tilde{\mathbf{x}}) \cdot \mathbf{n}(\tilde{\mathbf{x}}) \quad (3.46)$$

$$\gamma_1^{\text{ext}} \mathbf{u}(\mathbf{x}) = \lim_{\mathbb{R}^d \setminus \bar{\Omega} \ni \tilde{\mathbf{x}} \rightarrow \mathbf{x} \in \Gamma} \sigma(\mathbf{u}, \tilde{\mathbf{x}}) \cdot \mathbf{n}(\tilde{\mathbf{x}}) \quad . \quad (3.47)$$

Again, the normal vector is kept consistent in the geometric sense where it is defined to point out of closed surfaces or convex bodies. The normal vector, in the definition(s) above, does however point out of the fluid region for interior problems and into the fluid region for exterior problems.

Equations 3.38 and 3.40 may be written more compactly using shorthand matrix operator notation. We subsequently define the Calderón projections. The interior Calderón projection, C^{int} , and corresponding system of two boundary integral equations

can be written for $\mathbf{x} \in \Gamma$ as,

$$\begin{pmatrix} \gamma_0^{\text{int}} \mathbf{u} \\ \gamma_1^{\text{int}} \mathbf{u} \end{pmatrix} = \begin{pmatrix} (1-C)\mathbb{I} - K & G \\ D & C\mathbb{I} + K' \end{pmatrix} \begin{pmatrix} \gamma_0^{\text{int}} \mathbf{u} \\ \gamma_1^{\text{int}} \mathbf{u} \end{pmatrix} + \begin{pmatrix} N_0 f \\ N_1 f \end{pmatrix} \quad (3.48)$$

$$C^{\text{int}} = \begin{pmatrix} (1-C)\mathbb{I} - K & G \\ D & C\mathbb{I} + K' \end{pmatrix} . \quad (3.49)$$

where \mathbb{I} is the identity, K is the double layer potential, G is the single layer potential, D is the hypersingular potential, and γ_0 , γ_1 are respectively the boundary trace and co-normal derivative trace operators. Newton potentials N_0 and N_1 are left in for generality since they allow for coupling with the Finite Element Method. However in this thesis we are not concerned with volume body forced defined in the fluid.

Convention has it that, for problems with bounded domains, the normal vector points out of the interior domain. With the distinction of integration domains, Ω and $\mathbb{R}^d \setminus \bar{\Omega}$, it is only natural to also have an exterior Calderón projector for exterior boundary value problems. This is given as,

$$C^{\text{ext}} = \begin{pmatrix} C\mathbb{I} + K & -G \\ -D & [1-C]\mathbb{I} - K' \end{pmatrix} . \quad (3.50)$$

The corresponding exterior boundary value problem is written as,

$$\begin{pmatrix} \gamma_0^{\text{ext}} \mathbf{u} \\ \gamma_1^{\text{ext}} \mathbf{u} \end{pmatrix} = \begin{pmatrix} C\mathbb{I} + K & -G \\ -D & [1-C]\mathbb{I} - K' \end{pmatrix} \begin{pmatrix} \gamma_0^{\text{ext}} \mathbf{u} \\ \gamma_1^{\text{ext}} \mathbf{u} \end{pmatrix} + \begin{pmatrix} \mathbf{u}_0 \\ \mathbf{0} \end{pmatrix} , \quad (3.51)$$

where \mathbf{u}_0 may be some type of background flow. If the boundary is sufficiently smooth at the point \mathbf{x} we find that in all cases,

$$C(\mathbf{x}) = \frac{1}{2}\mathbb{I} . \quad (3.52)$$

In Eqns. 3.51 and 3.48, it is understood that the problem is to find the complete Cauchy datums $\{\gamma_0^{\text{ext/int}} \mathbf{u}(\mathbf{x}), \gamma_0^{\text{ext/int}} \mathbf{u}(\mathbf{x})\}$, $\mathbf{x} \in \Gamma$, which are given only partially by prescribed boundary conditions.

3.5 The Completed Double Layer Representation

In colloidal hydrodynamics problems, the exterior or interior problems, given respectively by 3.51 and 3.48, do not readily contain relevant quantities of interest. If one explicitly needs the distribution of velocities or surface tractions over a particle's

surface, Γ_p , then the Calderón construction is one adequate representation providing this solution. However for mobility problems, one may, for example, want to solve for particle rigid body motions subject to total external force and torque constraints. This total force does not appear explicitly in the boundary integral construction. Instead a larger system of equations is solved where force constraints are specified as,

$$F^{p,ext} = -F^{p,H} = - \int_{\Gamma_p} \mathbf{f}(\mathbf{x}) dS(\mathbf{x}) \quad (3.53)$$

$$T^{p,ext} = -T^{p,H} = - \int_{\Gamma_p} (\mathbf{x} - \mathbf{x}_c^p) \times \mathbf{f}(\mathbf{x}) dS(\mathbf{x}) \quad . \quad (3.54)$$

Rigid body motions typically are coupled to the fluid mechanics using stick or no-slip boundary conditions,

$$\mathbf{u}(\mathbf{x}) = \mathbf{U}^p + \boldsymbol{\Omega}^p \times (\mathbf{x} - \mathbf{x}_c^p), \quad \mathbf{x} \in \Gamma_p \quad , \quad (3.55)$$

where for instance,

$$\mathbf{U}^p = \frac{1}{S_{\Gamma_p}} \int_{\Gamma_p} \mathbf{u}(\mathbf{y}) dS(\mathbf{y}) \quad (3.56)$$

$$\boldsymbol{\Omega}^p = I_{\Gamma_p}^{-1} \int_{\Gamma_p} (\mathbf{x} - \mathbf{x}_c^p) \times (\mathbf{u}(\mathbf{y}) - \mathbf{U}^p) dS(\mathbf{y}) \quad , \quad (3.57)$$

where $I_{\Gamma_p}^{-1}$ may be found using the formulas of Sec. 2.11. The system of equations 3.38, 3.55 subject to constraints 3.53 and 3.54 is then fully specified up to discretization for a finite number of points \mathbf{x} but a significant amount of extra work is needed to re-arrange this system to find the solution. Roughly speaking the traction is given by $\mathbf{t} = (\mathcal{G}^{-1}K)\mathbf{u}$ which then must be substituted into Eqn. 3.53 and 3.54 and solved in conjunction with the help of Eqn. 3.55. There is no guarantee that \mathcal{G}^{-1} exists since it is derived from an integral equation of the first kind. Thus this description is relatively inadequate for our needs.

Instead, we would like to do as few computations as possible and only work with second kind integral equations (for numerical stability reasons). The completed double layer boundary integral equation method (CDL-BIEM) is one such construction that satisfies these requirements. CDL-BIEM is an indirect integral equation method for solving either the mobility or resistance problem. The method uses an indirect integral equation to represent a Stokes velocity field,

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \boldsymbol{\phi}(\mathbf{y}) dS(\mathbf{y}) \\ &= (K_{\Gamma_p} \boldsymbol{\phi})(\mathbf{x}) \end{aligned} \quad . \quad (3.58)$$

We remark here on a special property of the double layer potential. Specifically that the exterior or interior traces show jumps (discontinuities) in the potential ϕ . For smooth surfaces, it can be rigorously proven that [64],

$$\gamma_0^{\text{ext/int}}(K_{\Gamma_p}\phi)(\mathbf{x}) = \pm\frac{1}{2}\phi + (K_{\Gamma_p}\phi)(\mathbf{x}) \quad , \quad (3.59)$$

which indicates that there is a jump of $\pm\frac{1}{2}\phi$ when the limiting process is taken respectively from the direction $\pm\mathbf{n}$ in which the surface normal vector points. In this thesis we always multiply the double layer potential, in the indirect representation, by an additional factor of 2. With this factor of 2 the double layer has instead a discontinuous jump given by $\pm\phi$.

3.5.1 Completion Schemes

The velocity field associated with the indirect double layer representation is not an arbitrary Stokes flow field and ϕ technically has no relevant physical meaning, other than representing the double layers density distribution (though it still has units of velocity). Quite easily, it can be proven that the double layer operator K can only describe flows that are force and torque free and that decay as $1/r^2$. Power and Miranda [51] suggested completing the range of this operator with point force and point torque solutions suitably located inside of the particles. Their range completion scheme reads,

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \phi(\mathbf{x}) + \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) + \mathbf{F}^{p,\text{ext}} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \\ &\quad - \frac{1}{2} \left(\mathbf{T}^{p,\text{ext}} \times \nabla_{\mathbf{x}} \right) \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \\ &= \phi(\mathbf{x}) + \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) + \mathbf{F}^{p,\text{ext}} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \\ &\quad + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,\text{ext}} \quad , \quad (3.60) \end{aligned}$$

for $\mathbf{x} \in \Gamma_p$ and where $\mathbf{F}^{p,\text{ext}}$ and $\mathbf{T}^{p,\text{ext}}$ are the external forces and torques on the particle, $G(\mathbf{x}_c^p, \mathbf{x})$ is the Stokeslet centered at the particle's centroid \mathbf{x}_c^p , and $R(\mathbf{x}_c^p, \mathbf{x}) \equiv \epsilon_{ijk} \frac{(\mathbf{y}-\mathbf{x})}{r^3}$ is the Rotlet or Couplet singularity again centered at the particle's geometric centroid. The Stokeslet and Rotlet, respectively, can be easily proven to exert unit forces and torques on the particle when placed internally at the particle's center of mass. In this way, the range of the operator K can now represent arbitrary flows. If the no-slip condition is used to couple the fluid velocity to the particle RBMs,

six additional constraint equations are needed to fix the system. Power and Miranda first formulated these constraints in terms of the hydrodynamic force and torque. The CDL-BIEM method of Kim and Karilla [31], [30], [33] specifically targets the mobility problems by applying different constraints that are directly related to the particle RBMs. CDL-BIEM has been successfully applied to many different problems [49], [50], [48], [46]. A clear and detailed monograph of application to elasticity is given by Phan-Thien and Kim [47].

3.5.1.1 The Resistance Formulation

Since Power and Miranda [51] were mainly concerned with solving Stokes flow resistance problems, in order to fix the system, they chose an additional set of constraints, given by,

$$\mathbf{F}^{p,H} = - \sum_{i=1}^3 \mathbf{e}_i \langle \boldsymbol{\varphi}_i^{RBM}, \boldsymbol{\phi} \rangle \quad (3.61)$$

$$\mathbf{F}^{p,H} = - \sum_{i=1}^3 \mathbf{e}_i \langle \boldsymbol{\varphi}_{i+3}^{RBM}, \boldsymbol{\phi} \rangle \quad , \quad (3.62)$$

where $\boldsymbol{\varphi}_i^{RBM}$ for $i \in 1, \dots, 6$ are basis functions for the rigid body motions (RBMs) and are the null functions of $(\mathbb{I} + K)$. These basis functions collectively described the six eigenfunctions of the rank six eigenvalue K at -1. Here $\langle \mathbf{a}, \mathbf{b} \rangle \equiv \int_{\Gamma_p} \mathbf{a}(\mathbf{y}) \cdot \mathbf{b}(\mathbf{y}) dS(\mathbf{y})$ is an inner product and the set of RBMs must be orthonormal and satisfy,

$$\langle \boldsymbol{\varphi}^{i,RBM}, \boldsymbol{\varphi}^{j,RBM} \rangle = \delta_{ij} \quad . \quad (3.63)$$

We note here that construction of the set of RBMs satisfying Eqn. 3.63 is a relatively non-trivial computation unless the geometry is spherical. For spheres the RBMs become,

$$\boldsymbol{\varphi}^i = \frac{1}{\sqrt{S_{\Gamma_p}}} \mathbf{e}_i \quad (3.64)$$

$$\boldsymbol{\varphi}^{i+3}(\mathbf{x}) = \frac{1}{\sqrt{I_{i,\Gamma_p}}} \mathbf{e}_i \times (\mathbf{x} - \mathbf{x}_c^p) \quad . \quad (3.65)$$

For arbitrarily shaped particles, these RBMs may be constructed using the modified Gram-Schmidt orthonormalization process. This process is not explicitly given in the literature. The GPUGBEM framework uses the modified Gram-Schmidt

orthonormalization process to construct these RBMs and can therefore apply CDL-BIEM to arbitrary shaped particles.

We note here that this completion process is arbitrary in the sense that one needs only to impose six extra constraints that are linearly independent. However, we have chosen a set of constraints that is both convenient to implement and is known to provide a well-conditioned linear system. These constraints can be combined with the range completion to yield a smaller linear system. Rewriting the force and torque in terms of the constraints and using a symmetry property of G we find that,

$$\begin{aligned} \mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} &= \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \sum_{i=1}^3 \mathbf{e}_i \langle \varphi_i^{RBM}, \phi \rangle \\ &+ \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \sum_{i=1}^3 \mathbf{e}_i \langle \varphi_{i+3}^{RBM}, \phi \rangle \end{aligned} \quad (3.66)$$

The resistance problem for a rigid particle comprises the simultaneous solution of Eqns. 3.60, 3.61, and 3.62 subject to specification of \mathbf{U} and $\mathbf{\Omega}$ in Eqn. 3.55. These equations may all be combined together to yield,

$$\begin{aligned} \mathbf{U}^p + \mathbf{\Omega}^p \times (\mathbf{x} - \mathbf{x}_c^p) - \bar{\mathbf{u}}^p &= \phi(\mathbf{x}) + \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) \\ &+ \mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} \end{aligned} \quad (3.67)$$

Given $(\mathbf{U}, \mathbf{\Omega})$ one may solve Eqn. 3.67 for $\phi(\mathbf{x})$ for all points $\mathbf{x} \in \Gamma_p$. The resistance constraints Eqns. 3.61 and 3.62 are then used with the solution $\phi(\mathbf{x})$, in a post-processing step, to obtain the hydrodynamic forces and torques.

3.5.1.2 The Mobility Formulation

Convenient constraints for the mobility problem were first given by Kim and Karilla [31] and take the form,

$$\mathbf{U}^p = - \sum_{i=1}^3 \varphi_i^{RBM} \langle \varphi_i^{RBM}, \phi \rangle \quad (3.68)$$

$$\mathbf{\Omega}^p \times (\mathbf{x} - \mathbf{x}_c^p) = - \sum_{i=1}^3 \varphi_{i+3}^{RBM} \langle \varphi_{i+3}^{RBM}, \phi \rangle \quad (3.69)$$

These constraints may be combined with the boundary condition on the particle to yield,

$$\mathbf{U}^p + \mathbf{\Omega}^p \times (\mathbf{x} - \mathbf{x}_c^p) = - \sum_{i=1}^6 \varphi_i^{RBM} \langle \varphi_i^{RBM}, \phi \rangle \quad (3.70)$$

Kim and Karrila designed this completion scheme with the primary intention of seeking a Picard iterative based solution to Eqn. 3.60. It is important to point out that this choice of these specific completion constraints is the only valid choice if one seeks to obtain an iterative solution method cast as a fixed point problem where the linear operator, K , is made to be a contraction mapping with spectral radius less than one. With these constraints, the mobility problem may be solved using a Neumann series expansion, which in practice, is replaced by standard Picard iteration. This means that matrix inversions are not necessary and that by construction the method is $O(N^2)$ for N points $\mathbf{x} \in \Gamma$. We do not choose to pursue this iterative construction here; however, throughout this thesis we extensively use the CDL-BIEM construction with Power and Miranda's force and torque range completer, and with either the resistance or mobility completion schemes.

Combining Eqns. 3.60 and 3.70 yields a version of the CDL-BIEM mobility formulation,

$$-\mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} - \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} = \phi(\mathbf{x}) + \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) + \sum_{i=1}^6 \varphi_i^{RBM} \langle \varphi_i^{RBM}, \phi \rangle \quad (3.71)$$

Given the external force and torque, $(\mathbf{F}^{p,ext}, \mathbf{T}^{p,ext})$, one may solve Eqn. 3.71 for $\phi(\mathbf{x})$ for all points $\mathbf{x} \in \Gamma_p$. The mobility constraints Eqns. 3.68 and 3.69 are then used with the solution $\phi(\mathbf{x})$, in a post-processing step, to obtain the translational velocity and angular velocity.

3.5.2 The CDL Representation of an Active Particle

Active particles are defined by so-called *force-free* and *torque-free* motion. These terms are often very loosely used in describing active matter. What this precisely means is that the net hydrodynamic force and torque on the particle is zero,

$$\mathbf{F}^h = \mathbf{0} \quad (3.72)$$

$$\mathbf{T}^h = \mathbf{0} \quad (3.73)$$

If the particle were a rigid force and torque free body then it would not move subject to conditions 3.72 and 3.73. In other words, the hydrodynamic drag on the body would be zero. Noticing this fact allows us to identify the rigid body motions (RBMs) of the particle with the hydrodynamic drag force. In this case these two quantities

are related through the use of a resistance force-velocity operator. In the case of an actively propelled particle, the surface velocity is described by,

$$\mathbf{u}(\mathbf{x}) = \mathbf{U} + \boldsymbol{\Omega} \times (\mathbf{x} - \mathbf{x}_c) + \mathbf{u}^s(\mathbf{x}) \quad , \quad (3.74)$$

where $\mathbf{u}^s(\mathbf{x})$ is a point-wise additional *slip* velocity piece. If this body is now made force and torque free, the RBMs are still identified as giving rise to the drag force, but the slip velocity piece must give rise to an additional forcing term that is equal and opposite to the drag force. In this thesis, we call this the bodies *propulsive force*. If an active particle is held fixed, swimming in place, the force required to keep it from moving is precisely the propulsive force that arises from surface velocity piece, $\mathbf{u}^s(\mathbf{x})$.

Since the active particle is net force and torque free, the range completion term is not needed in the CDL description. The active particle may be perfectly modeled by the double layer alone and the mobility problem reads,

$$\mathbf{u}^s(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}) + \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \boldsymbol{\phi}(\mathbf{y}) dS(\mathbf{y}) + \sum_{i=1}^6 \boldsymbol{\varphi}_i^{RBM} \langle \boldsymbol{\varphi}_i^{RBM}, \boldsymbol{\phi} \rangle \quad . \quad (3.75)$$

This representation is incredibly simple and is capable of accurately describing the motion of any force and torque free active particle whose motion is driven by a slip velocity $\mathbf{u}^s(\mathbf{x})$. The range completion scheme only re-enters the problem in the event that one wants to include external forces and torques in the description of active particles (e.g. bottom heavy swimmers).

3.6 Conclusions

In this chapter we have shown that the Stokes BEM problem may be cast into another form, given by Eqn. 3.71, which may be solved with much greater efficiency and ease. This alternative form utilizes an indirect second kind integral equation representation and is known as CDL-BIEM. Using techniques from linear operator theory, one may prove that this representation of the mobility solution for multiple particles, in either a bounded container or an extended domain, may be cast in terms of fixed point problem where the linear operator (in this case the double layer potential) is a contraction mapping with spectral radius less than one. Moreover, it was shown that the CDL representation simplifies when it is used to model an active force and torque free particle. Because of this simplification, and contraction mapping property, it is likely that this BIE representation (Eqn. 3.75) is the most efficient representation

for describing active particles with arbitrary geometries in Stokes flow. In subsequent chapters we use this description to study the dynamics of active particles.

Chapter 4

The GPUGBEM Framework and the Galerkin Discretization

There currently exist no adequate open-source software packages or implementations for the discretization of boundary integral operators in Stokes flow. Moreover, there exist no BEM libraries that specifically implement Galerkin Boundary Element computations across multiple bodies or several coupled geometries in 3D Stokes Flow. A major portion of the work in this thesis has been in creation of a new, novel, fast, and efficient framework, dubbed GPUGBEM (GPU Galerkin Boundary Element Method), that fills this void. GPUGBEM is a BEM library, written by the author, that uses both static and dynamic polymorphism to generate Galerkin discretizations of BEM matrix operators. GPUGBEM represents surface meshes using the VCG library [10] and couples this high quality geometry library to h-adaptive Galerkin boundary element calculations. The h-adaptive mesh class has been written in house, and is used to store and perform conformal h-adaptive mesh refinement on an arbitrary BEM mesh. Mesh adaptivity and the Galerkin BEM calculations are defined in ways so that the extension of calculations for systems involving multiple bodies of arbitrary shape becomes trivial. In addition, fast n-body simulation techniques have been used to translate Galerkin BEM calculations onto GPUs using NVIDIA's CUDA Runtime API. GPUGBEM also implements rigid body dynamics algorithms for computing particle trajectories. Thus GPUGBEM allows for full dynamics simulations of arbitrarily shaped particles to be computed quickly on modern desktop computers that are equipped with a modern GPU.

Other boundary element packages do exist, which seek, in large part, to unify the language of BEM through object-orientated design (OOD). In particular there is BEM, [60], which realizes all BEM abstractions using object-orientated design and makes heavy use of virtual functions and inheritance when representing generic BEM

computations. However, this package has gone through multiple extensive revisions since this thesis work was started and has only recently appeared to be stable. Also, the BEM++ library is exclusively tailored towards the solution of the Laplace, Helmholtz, or Maxwell equations and it does not easily support multiple bodies or mesh geometries. There is also BETL, [21], which is built upon generic programming paradigms and C++ templates. This library generates optimized generic code at compile time for a specific boundary element problem. In contrast to BEM++'s dynamic polymorphism, BETL implements static compile time polymorphism. However, BETL seems to suffer from requiring its users to sign a restrictive license agreement. Static compile time polymorphism is also relatively restrictive. For example this type of coding prevents any hopes of implementing adaptive h or p meshing routines or any algorithms that require element types to change. Although, BETL claims to be fully decoupled from a specific mesh implementation (since it provides STL iterators to mesh objects) it does require that element types be specified at compile time and be uniform across the entire surface. BETL is also not readily amenable to performing BEM calculations across multiple surfaces. In this thesis we have extensively used programming ideas and principles found in both of these packages, but GPUGBEM has been written from scratch and efficiently combines both static and dynamic types of polymorphism with adaptive meshing and multi-body BEM capabilities.

In this chapter we first formalize the Galerkin Boundary Element discretization using the Calderón projection. This construction yields the general symmetric form of GBEM for a single surface. We take this procedure and apply it to discretize the completed double layer boundary integral representation of Chapter 3. This construction is generalized to multi-body systems. We show how the resistance and mobility problems may be posed using GBEM. We describe a general algorithm and procedure for automatically computing GBEM matrix representations for an arbitrary set of particles that may each be individually described by different boundary integral equations. The GPUGBEM framework is briefly explained in the context of the CDL representation for an active particle. Finally we discuss the h -adaptive meshing algorithm and its coupling to GBEM.

4.1 Boundary Element Methods

The Boundary Element Method (BEM) is a relatively mature numerical technique that solves elliptic partial differential equations with high accuracy at reasonable computational expense. The method is quite general and can be applied to many different topics and problems that may come from a variety of different fields including

elasticity, geomechanics, structural mechanics, electromagnetism, acoustics, hydraulics, low Reynolds number hydrodynamics, and biomechanics. The strength of the method lies in its ability to provide efficient solutions to problems that have complex and even time evolving or flexible geometries.

The BEM is formulated from an associated set of boundary integral equations (BIEs) that are often derived from a partial differential equation, or reciprocal identity. Integral equation formulations of PDE's have been known for well over a century, however it was not until 1978 that Brebbia [9] formalized the boundary integral equation method (BIEM) and introduced BEM as a discretization method used to compute the numerical solution of the associated BIEs. Brebbia approximated the continuous surface boundary integrals as a sum of integrals over discrete surface elements. This description naturally requires the discrete mesh representation, Γ^h , of a given object's (particle's) surface.

Boundary Element Methods enjoy significant advantages over volume based methods in certain situations. They are extremely efficient in treating exterior domain problems and offer the benefit of working with discretized surface meshes rather than a volume discretization. One downside to working with BEM is that the representative underlying linear BIEs discretize as dense matrices. This is due to the decay properties of the BEM singularities; the boundary datums emanate in all directions with a fairly persistent and *long range* decay that is usually at least $O(r^{-1})$ or $O(r^{-2})$. Conversely, volume or spatial methods such as Finite Element Methods (FEM) or Finite Difference Methods (FDM), yield sparse matrix equations and matrix representations. Solution of these sparse systems can then be efficiently obtained by the use of iterative solvers or by a straightforward mapping to large parallel architectures. However, FEM and FDM also lead to much larger, although sparse system or so-called *stiffness* matrices.

In multi-body colloidal hydrodynamics problems, BEM enjoys exceptional advantages over FEM and FDM. This is mainly due to the fact that as particles move in space, volume based methods must re-mesh a significant portion of the entire solution domain. Quite conversely, the BEM meshes are attached to the rigid body motions of the particle and may be evolved in space subject to a simple rigid body affine transform (rotation plus a translation). Re-meshing in BEM only needs to occur when geometries become distorted or come into close contact. It is also well known that volume base FEM and FDM are much less accurate in calculating Neumann datums, (i.e. derivatives) due to the fact that they must be numerically post-processed and do not come out naturally from the problem. As was seen in Chapter 3, surface tractions arise naturally in the formulation of the Calerón projection and may be computed with high accuracy with absolutely no post-processing.

In BEM we solve for the boundary distribution of an unknown function (or for one of its derivatives). Once the boundary distribution of the function is known, its value at any point in the domain can be produced by direct evaluation. BEM reduces the dimensionality a given elliptic partial differential equation by one order (e.g. in 3D the solution space is reduced to 2D surface distributions rather than the original physical volume). This dimensionality reduction results from expressing the solution in terms of the boundary distribution of fundamental singularity solutions of the underlying partial differential equation. Distribution densities of these singularities are then computed subject to boundary conditions.

In Stokes flows, direct formulations of BEM, where the primary variables are the tractions and velocities on the particle surfaces, have been used to solve a vast array of problems (see [24]). For example, a series of three papers by Ingber and Mondy [15],[42],[23] solve the mobility problem in Stokes flow. The papers by Ingber and Mondy use a direct formulation of BEM that leads to integral equations of the first kind (IE1). They use Gaussian elimination-type linear solvers to find solutions to the mobility since the more efficient iterative solvers can not deal with the high linear system condition numbers that are characteristic of IE1s. It has been well known since Fredholm's seminal papers on integral equations (circa 1900) that IE1s lead to ill-conditioned problems. In fact it is quite easy to see that for an inhomogeneous IE1 and a continuous degenerate kernel, there can only be a solution if the inhomogeneous term lies in the range of the kernel (or in this case is a multiple of the part that gets integrated out).

As discussed and shown in Chapter 3, the completed double-layer boundary integral equation method (CDL-BIEM) provided a solution to the ill-conditioning problem. Formulated by Power and Miranda [51], CDL-BIEM is a range completion method that formulates the Stokes flow BIEs in terms of integral equations of the second kind (IE2). This method has in turn gained popularity, especially in solving large systems where the IE1 condition number becomes particularly large. CDL-BIEM was formalized by the work of Karrila et al. [31], Phan-Thien and Kim [47], Phan-Thien et al. [49], Phan-Thien and Tullock [48], and Power and Wrobel [52]. Also much on BIM, BEM, and CDL-BIEM are formalized in the great books and works of Pozrikidis [53] and [54]. More technical books exist. For Galerkin BEM the excellent books by Sauter and Schwab [58] and Steinbach [64] read more like rigorous mathematical treatises and are highly recommended.

The BEM results in a coupled set of linear algebraic equations. For small systems of equations techniques such as LU decomposition may be readily used to obtain the solution. For larger systems one must use iterative solvers such as the CG method by

Hestenes and Stiefel [20], the CGS method by Sonneveld [63], the BICGstab method by van der Vorst [66], or the GMRES method by Saad and Schultz [56]. GPUGBEM makes extensive use of these solvers, depending on the symmetries and properties of the discretized matrix operators.

BEM formulations are naively $O(N^3)$ for N degrees of freedom (e.g. mesh nodes) owing solution using a primitive dense matrix inversion technique. This is the primary reason for why BEM is not readily used for in the simulation of dense macromolecular suspensions. If however, one combines a Krylov subspace iterative solver with either the Barnes-Hut multipole methods [5] or fast multipole methods (FMM) [17] the calculation time complexity changes to respectively $O(N \log(N))$ and $O(N)$ [16]. If each particle has N degrees of freedom, this means that the simulation of potentially thousands of particles is possible on desktop computers. A relatively different approach that is highly competitive with FMM is in representing BEM matrices with so-called *hierarchical matrices*. These matrices are termed H -matrices [18] and are data sparse representations of the dense BEM matrices. Nearly all matrix operations may be performed on H -matrix representations in almost optimal time complexities. These fast algorithms have not been implemented in the GPUGBEM framework. This is left as future work.

Both BEM and CDL-BIEM have trouble dealing with lubrication forces when suspensions go above 30 volume percent. Lubrication corrections require one to either adaptively re-mesh a particles surface in order to resolve “numerically” singular integrals between close particles¹ or utilize nonlinear transformation techniques to smooth out the singularity. In this thesis, near contact interactions are resolved using spatial adaptivity on the mesh.

4.1.1 Symmetric Galerkin Formulation

The equations from the interior Calderón system are evaluated on a boundary partitioning Γ_D and Γ_N such that $\Gamma = \Gamma_D \cup \Gamma_N$,

$$C\mathbf{g}_D(\mathbf{x}) = (G\gamma_1^{\text{int}}\mathbf{u})_\Gamma(\mathbf{x}) - (K\gamma_0^{\text{int}}\mathbf{u})_\Gamma(\mathbf{x}) + (N_0f)_\Omega(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_D \quad (4.1)$$

$$[I - C]\mathbf{g}_N(\mathbf{x}) = (K'\gamma_1^{\text{int}}\mathbf{u})_\Gamma(\mathbf{x}) + (D\gamma_0^{\text{int}}\mathbf{u})_\Gamma(\mathbf{x}) + (N_1f)_\Omega(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_N \quad (4.2)$$

¹The double layer can become singular as particle elements approach one another since in close proximity a small separation between points could lose numerical significance or result in a sharply peaked integrand.

where we have made use of the known Cauchy data \mathbf{g}_N and \mathbf{g}_D . The exterior case follows similarly as,

$$[I - C]\mathbf{g}_D(\mathbf{x}) = -(G\gamma_1^{\text{ext}}\mathbf{u})_{\Gamma}(\mathbf{x}) + (K\gamma_0^{\text{ext}}\mathbf{u})_{\Gamma}(\mathbf{x}) + u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_D \quad (4.3)$$

$$C\mathbf{g}_N(\mathbf{x}) = -(K'\gamma_1^{\text{ext}}\mathbf{u})_{\Gamma}(\mathbf{x}) - (D\gamma_0^{\text{ext}}\mathbf{u})_{\Gamma}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_N \quad . \quad (4.4)$$

The Cauchy data is then decomposed into known and unknown parts,

$$\begin{aligned} \gamma_0^{\text{int/ext}}\mathbf{u}(\mathbf{x}) &= \mathbf{g}_D(\mathbf{x}) + \tilde{\mathbf{u}}_D(\mathbf{x}) \\ \gamma_1^{\text{int/ext}}\mathbf{u}(\mathbf{x}) &= \mathbf{g}_N(\mathbf{x}) + \tilde{\mathbf{u}}_N(\mathbf{x}) \end{aligned} \quad , \quad (4.5)$$

in which the unknown Cauchy data is denoted as $\tilde{\mathbf{u}}_D(\mathbf{x})$ and $\tilde{\mathbf{u}}_N(\mathbf{x})$. Inserting the decompositions shown in Eqn. 4.5 into Eqn. 4.1-4.4 and transferring the resulting system to residual form with appropriate test functions $w(\mathbf{x})$ and $v(\mathbf{x})$ yields the symmetric Galerkin formulation of the interior problem,

$$\langle G\tilde{\mathbf{u}}_N, w \rangle_{\Gamma_D} - \langle K\tilde{\mathbf{u}}_D, w \rangle_{\Gamma_D} = \langle (C + K)\mathbf{g}_D - G\mathbf{g}_N - N_0f, w \rangle_{\Gamma_D} \quad (4.6)$$

$$\langle K'\tilde{\mathbf{u}}_N, v \rangle_{\Gamma_N} + \langle D\tilde{\mathbf{u}}_D, v \rangle_{\Gamma_N} = \langle (I - C - K')\mathbf{g}_N - D\mathbf{g}_D - N_1f, v \rangle_{\Gamma_N} \quad , \quad (4.7)$$

and the symmetric Galerkin formulation of the exterior problem,

$$\langle -G\tilde{\mathbf{u}}_N, w \rangle_{\Gamma_D} + \langle K\tilde{\mathbf{u}}_D, w \rangle_{\Gamma_D} = \langle (I - C - K)\mathbf{g}_D + G\mathbf{g}_N - \mathbf{u}_0, w \rangle_{\Gamma_D} \quad (4.8)$$

$$\langle -K'\tilde{\mathbf{u}}_N, v \rangle_{\Gamma_N} - \langle D\tilde{\mathbf{u}}_D, v \rangle_{\Gamma_N} = \langle (C + K')\mathbf{g}_N + D\mathbf{g}_D, v \rangle_{\Gamma_N} \quad , \quad (4.9)$$

for $\langle f, g \rangle_{\Gamma} \equiv \int_{\Gamma} f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$. To proceed with numerical solution, both the inner product and all boundary element operators must be discretized. Thus, the surface Γ must be discretized. In addition, appropriate spaces of test functions $w(\mathbf{x})$ and $v(\mathbf{x})$ must be defined. These test functions will be constructed explicitly as nodal element functions from the surface-mesh discretization.

4.1.2 Boundary Approximation and Discretization

First we introduce a decomposition of the k 'th surface, Γ_k , into a union of M disjoint boundary elements (also known as a mesh),

$$\begin{aligned} \Gamma_k &\approx \Gamma_{k,M}^h = \bigcup_{m=1}^M \bar{\tau}_m \quad , \\ |\Gamma_k^h| &= M \end{aligned} \quad (4.10)$$

with boundary elements τ_m , and a notion of mesh size, h . From here we make the following approximation,

$$\int_{\Gamma} (\cdots) dS(\bar{\mathbf{y}}) \approx \sum_{m=1}^M \int_{\tau_m} (\cdots) dS(\mathbf{y}), \quad \forall \bar{\mathbf{y}} \in \Gamma \text{ and } \mathbf{y} \in \Gamma_{k,M}^h, \quad (4.11)$$

For the integration of the boundary element, τ_m , a parameterization is introduced such that every boundary element τ is the image of a reference element, $\hat{\tau}$ under a mapping,

$$\chi_{\tau} : \hat{\tau} \rightarrow \tau \quad . \quad (4.12)$$

Boundary elements τ are defined as $d - 1$ dimensional surface patches embedded in a d dimensional space. For three-dimensional Euclidean space, τ corresponds to a two dimensional surface patch embedded in \mathbb{R}^3 . Therefore the mapping $\chi_{\tau} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ maps from the plane into three dimensional Euclidean space. The boundary elements usually consist of either triangles or quadrilaterals and so the corresponding reference elements are chosen to be either the unit triangle, $\hat{\tau}_{\Delta}$ or the unit square, $\hat{\tau}_{\square}$. These reference elements may be defined as,

$$\hat{\tau}_{\Delta} \equiv \{(\xi_1, \xi_2) \in \mathbb{R}^2 : 0 \leq \xi_2 < \xi_1 \leq 1\} \quad (4.13)$$

$$\hat{\tau}_{\square} \equiv \{(\xi_1, \xi_2) \in \mathbb{R}^2 : 0 \leq (\xi_2, \xi_1) \leq 1\} \quad . \quad (4.14)$$

If these reference domains are re-defined, then an appropriate modification of shape functions will be necessary. Each reference element $\hat{\tau}$ has a set of local points $\hat{\mathbf{p}}_i$ where $i = 1, \dots, N$. The correspondence between the number of points belonging to boundary element uniquely defines the geometry of that element. The geometry of the element τ is defined by the set,

$$E_N \equiv \bigcup_{i=1}^N \mathbf{p}_i \quad . \quad (4.15)$$

Commonly these points are referred to as *nodes* and or *boundary nodes*. For example the sets $E_3 = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ and $E_4 = \{\mathbf{p}_1, \dots, \mathbf{p}_4\}$ define planar triangles and quadrilaterals. These two element types are commonly called *flat elements*. Also available are so called *curved elements* which include larger sets of nodes in their definitions. An interesting research direction would be to discover or invent new parameterizations that facilitate more accurate numerical quadrature techniques. For example, a charts-atlas parameterization can generate much smoother surfaces and higher quality geometric representations. This in turn improves interpolation errors and overall solution quality.

The mapping χ_τ is commonly (but need not be) defined as a linear combination of nodal shape functions, $\varphi_i^\alpha \in \mathbb{R}$ of order α such that reference coordinates $\boldsymbol{\xi} \in \hat{\tau}$ are mapped to points $\boldsymbol{x} \in \tau$. With a polynomial ansatz of dimension N , a χ_τ mapping can be defined by,

$$\chi_\tau(\boldsymbol{\xi}) = \boldsymbol{x}(\boldsymbol{\xi}) = \sum_{i=1}^N \varphi_i^\alpha(\boldsymbol{\xi}) \boldsymbol{p}_i, \quad \alpha \in \mathbb{N} \quad . \quad (4.16)$$

Every node \boldsymbol{p}_j is related to a distinct local or reference node $\hat{\boldsymbol{p}}_j$ such that

$$\boldsymbol{p}_j = \chi_\tau(\hat{\boldsymbol{p}}_j)$$

holds. If this constraint is inserted into Eqn. 4.15, the description is that of Lagrange nodal shape functions which satisfy the delta property,

$$\varphi_i^\alpha(\hat{\boldsymbol{p}}_j) = \delta_{ij} \quad . \quad (4.17)$$

Construction of the nodal basis shape functions involves inversion of a generalized Vandermonde matrix. The construction proceeds as follows. For every nodal point $\hat{\boldsymbol{p}}_j$ one can define a linear form that maps elements, g , from a polynomial space $P = P^\alpha(\hat{\tau})$ to \mathbb{R} ,

$$L_j : g \in P \rightarrow g(\hat{\boldsymbol{p}}_j) \in \mathbb{R} \quad . \quad (4.18)$$

Expanding each shape function using a basis $\{g_1, \dots, g_N\}$, it follows that,

$$\varphi_j = \sum_{k=1}^N a_{kj} g_k \quad . \quad (4.19)$$

From here, the delta property implies that,

$$\delta_{ij} = L_i(\varphi_j) = L_i \left(\sum_{k=1}^N a_{kj} g_k \right) = \sum_{k=1}^N L_i(g_k) a_{kj}, \quad 1 \leq i, j \leq N \quad , \quad (4.20)$$

which yields a system of N linear equations for each j . From here one obtains a matrix equation,

$$\begin{aligned} \mathbf{L}\mathbf{A} &= \mathbf{I} \\ \mathbf{L} &= \{L_i(g_k)\}_{i,k=1}^N \quad . \\ \mathbf{A} &= \{a_{kj}\}_{k,j=1}^N \end{aligned} \quad (4.21)$$

Matrix \mathbf{L} is a generalized Vandermonde matrix, \mathbf{I} is the identity, and \mathbf{A} is the coefficient matrix where columns contain coefficients of the corresponding shape functions. Shape functions in 2D FEM are almost always identical to the 3D BEM shapefunction constructions. Good resources that derive shape functions for various elements include the excellent books by Solín, [62], [61] and the libMesh library, [34].

4.1.2.1 Linear Quadrilateral Shape Functions

To find the shape functions for a linear quadrilateral element, $\hat{\tau}_{\square}$, the polynomial space is taken to be $P^1(\hat{\tau}_{\square}) = \text{span}\{1, \xi_1, \xi_2, \xi_1\xi_2\}$. The set of linear forms then becomes,

$$\begin{aligned} g_1(\hat{\mathbf{x}}) &= 1, & g_2(\hat{\mathbf{x}}) &= \xi_1 \\ g_3(\hat{\mathbf{x}}) &= \xi_2, & g_4(\hat{\mathbf{x}}) &= \xi_1\xi_2 \\ L_1(g) &= g(\hat{\mathbf{p}}_1), & L_2(g) &= g(\hat{\mathbf{p}}_2) \\ L_3(g) &= g(\hat{\mathbf{p}}_3), & L_4(g) &= g(\hat{\mathbf{p}}_4) \end{aligned} \quad . \quad (4.22)$$

From which the generalized Vandermonde matrix is determined to be,

$$\mathbf{L}_{\square} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad , \quad (4.23)$$

and the coefficient matrix \mathbf{A} is,

$$\mathbf{A}_{\square} = \mathbf{L}^{-1}\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \quad , \quad (4.24)$$

and the shape functions are,

$$\varphi_1^1(\boldsymbol{\xi}) = 1 - \xi_1 - \xi_2 + \xi_1\xi_2 \quad (4.25)$$

$$\varphi_2^1(\boldsymbol{\xi}) = \xi_1 - \xi_1\xi_2 \quad (4.26)$$

$$\varphi_3^1(\boldsymbol{\xi}) = \xi_1\xi_2 \quad (4.27)$$

$$\varphi_4^1(\boldsymbol{\xi}) = \xi_2 - \xi_1\xi_2 \quad . \quad (4.28)$$

4.1.2.2 Linear Triangle Shape Functions

To find the shape functions for a linear quadrilateral element, $\hat{\tau}_\square$, the polynomial space is taken to be $P^1(\hat{\tau}_\square) = \text{span}\{1, \xi_1, \xi_2\}$. The set of linear forms then becomes,

$$\begin{aligned} g_1(\hat{\mathbf{x}}) &= 1, & g_2(\hat{\mathbf{x}}) &= \xi_1, & g_3(\hat{\mathbf{x}}) &= \xi_2 \\ L_1(g) &= g(\hat{\mathbf{p}}_1), & L_2(g) &= g(\hat{\mathbf{p}}_2), & L_3(g) &= g(\hat{\mathbf{p}}_3) \end{aligned} \quad . \quad (4.29)$$

From which the generalized Vandermonde matrix is determined to be,

$$\mathbf{L}_\square = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad . \quad (4.30)$$

and the coefficient matrix \mathbf{A} is,

$$\mathbf{A}_\square = \mathbf{L}^{-1}\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \quad . \quad (4.31)$$

and the shape functions are,

$$\varphi_1^1(\boldsymbol{\xi}) = 1 - \xi_1 \quad (4.32)$$

$$\varphi_2^1(\boldsymbol{\xi}) = \xi_1 - \xi_2 \quad (4.33)$$

$$\varphi_3^1(\boldsymbol{\xi}) = \xi_2 \quad . \quad (4.34)$$

4.1.3 Galerkin Discretization

We define spaces of continuous boundary elements of order p on our discrete mesh Γ_k^h as,

$$S^{p,0}(\Gamma_k^h, \boldsymbol{\chi}) \equiv \{\psi \in C^0(\Gamma_k) | \forall \tau \in \Gamma_k^h : \psi|_\tau \circ \chi_\tau \in \mathbb{P}_p^\tau\} \quad . \quad (4.35)$$

Conversely we denote the space of discontinuous boundary elements of order p by $S^{p,-1}(\Gamma_k^h)$. In this thesis, all calculations exclusively use piecewise linear and continuous boundary elements from the space, $S^{1,0}$. We may define boundary element spaces over the spatial components of the partition $\Gamma_k^h = \Gamma_{k,D}^h \cup \Gamma_{k,N}^h$. We may choose to define different spaces for the Dirichlet and Neumann datums, $S^\gamma(\Gamma_{k,D}^h)$ and $S^\beta(\Gamma_{k,N}^h)$ in accordance with Eqn. 4.35. In the construction shown here we select $\gamma = \beta \equiv \gamma \geq 1$ so as to avoid discontinuous representations of the Dirichlet data. The basis functions used in expanding the unknown datums are referred to as *trial* functions. These

expansions may be written as,

$$\tilde{\mathbf{u}}_N \approx \mathbf{u}_{N,h}^\gamma(\mathbf{x}) = \sum_{i=1}^N \mathbf{u}_{N,i} \varphi_i^\gamma(\mathbf{x}), \quad \in S^{\gamma,0}(\Gamma_{k,N}^h) \quad (4.36)$$

$$\tilde{\mathbf{u}}_D \approx \mathbf{u}_{D,h}^\gamma(\mathbf{x}) = \sum_{i=1}^M \mathbf{u}_{D,i} \psi_i^\gamma(\mathbf{x}), \quad \in S^{\gamma,0}(\Gamma_{k,D}^h) \quad (4.37)$$

These approximations are inserted into the variational formulation to yield two equations with N unknown parameters $u_{N,k}$ and M unknown parameters $u_{D,l}$. To yield a system of equations, the variational formulation is then passed through the respective subspace's basis such the the problem reads:

Find $(\mathbf{u}_{N,h}^\gamma, \mathbf{u}_{D,h}^\gamma)$ such that the interior problem satisfies

$$\langle G\tilde{\mathbf{u}}_{N,h}^\gamma, w_h^\gamma \rangle_{\Gamma_{D,h}} - \langle K\tilde{\mathbf{u}}_{D,h}^\gamma, w_h^\gamma \rangle_{\Gamma_D^h} = \langle (C + K) \mathbf{g}_{D,h}^\gamma - G\mathbf{g}_{N,h}^\gamma - N_0 f_h, w_h^\gamma \rangle_{\Gamma_D^h} \quad (4.38)$$

$$\langle K' \tilde{\mathbf{u}}_{N,h}^\gamma, v_h^\gamma \rangle_{\Gamma_N^h} + \langle D\tilde{\mathbf{u}}_{D,h}^\gamma, v_h^\gamma \rangle_{\Gamma_N^h} = \langle (I - C - K') \mathbf{g}_{N,h}^\gamma - D\mathbf{g}_{D,h}^\gamma - N_1 f_h, v_h^\gamma \rangle_{\Gamma_N^h} \quad (4.39)$$

for all test functions (w_h^γ, v_h^γ) and or such that the exterior problem satisfies,

$$\langle -G\tilde{\mathbf{u}}_{N,h}^\gamma, w_h^\gamma \rangle_{\Gamma_{D,h}} + \langle K\tilde{\mathbf{u}}_{D,h}^\gamma, w_h^\gamma \rangle_{\Gamma_D^h} = \langle (I - C - K) \mathbf{g}_{D,h}^\gamma + G\mathbf{g}_{N,h}^\gamma - \mathbf{u}_{0,h}, w_h^\gamma \rangle_{\Gamma_D^h} \quad (4.40)$$

$$\langle -K' \tilde{\mathbf{u}}_{N,h}^\gamma, v_h^\gamma \rangle_{\Gamma_N^h} - \langle D\tilde{\mathbf{u}}_{D,h}^\gamma, v_h^\gamma \rangle_{\Gamma_N^h} = \langle (C + K') \mathbf{g}_{N,h}^\gamma + D\mathbf{g}_{D,h}^\gamma, v_h^\gamma \rangle_{\Gamma_N^h} \quad (4.41)$$

is satisfied for all $w_h^\gamma \in S^{\gamma,0}(\Gamma_{k,D}^h)$ and $v_h^\gamma \in S^{\gamma,0}(\Gamma_{k,N}^h)$. Equations 4.40 and 4.41 are understood to be discretized under application of Eqn. 4.11 to the inner product, and a second time to individual BEM operators over the subscripted surface. Numerical solution proceeds by transforming these integrals to the reference element domain taking them to quadrature.

4.1.4 Transformation of Integrals to Reference Elements

To transform Eqn. 4.11 to local coordinates, it remains to expressed the differential surface element $dS(\mathbf{y})$ in local coordinates. The *Jacobian matrix* represents this transformation as,

$$\mathbf{J}_\tau(\boldsymbol{\xi}) = \begin{bmatrix} \frac{\partial \chi_\tau(\boldsymbol{\xi})}{\partial \xi_1} & \frac{\partial \chi_\tau(\boldsymbol{\xi})}{\partial \xi_2} \end{bmatrix}, \quad \mathbf{J}_\tau \in \mathbb{R}^{3 \times 2} \quad (4.42)$$

and gives a linear approximation of the vector field \mathbf{x} in a neighborhood of $\boldsymbol{\xi}$. The deformation of an infinitesimal surface element $dS(\mathbf{x})$ is then given by,

$$\begin{aligned} dS(\mathbf{x}) &= \sqrt{\det(\mathbf{J}_\tau^T(\boldsymbol{\xi}) \cdot \mathbf{J}_\tau(\boldsymbol{\xi}))} d\boldsymbol{\xi} \\ &= \sqrt{g_\tau(\boldsymbol{\xi})} d\boldsymbol{\xi} \end{aligned} \quad , \quad (4.43)$$

where $g_\tau(\boldsymbol{\xi})$ is the *Gram determinant*.

4.1.5 Discrete Forms of Galerkin Boundary Element Integrals

We simplify the discussion by examining the explicit forms of integral operators shown in Eqn. 4.40 and 4.41. The basis functions described so far are globally defined on the discrete mesh Γ_k^h . They are chosen to be related to the the local element basis functions through the notion of local element interpolations. More specifically on the reference element, the global test and trial functions identified as the same set of local element basis functions. This construction is given formally as,

$$\psi_i(\mathbf{x}) \Big|_{\tau_k} = \begin{cases} (\psi_{j,\tau_k} \circ \chi_{\tau_k}^{-1})(\mathbf{x}), & \tau_k \in \text{supp}(\psi_i), \quad \psi_{j,\tau_k}(\chi_{\tau_k}^{-1}(\mathbf{p}_i)) = 1 \\ 0 \end{cases} \quad (4.44)$$

where $\text{supp}(\psi_i)$ is the support of the i 'th globally labeled node. For flat triangle meshes, this support is simply the i 'th node's vertex-face 1-ring $v_{i,k}^* \equiv \{\tau : \tau \in \Gamma_k^h, \mathbf{p}_i \in \bar{\tau}\}$. This domain may be written as,

$$\text{supp}(\psi_i) = \{\tau \in \Gamma_k^h : \tau \in v_i^*\} \quad . \quad (4.45)$$

Since we are ultimately solving for unknown datums at global boundary nodes (i.e. vertices) of the flat triangles, the construction Eqn. 4.11 implies that the i, j matrix element for the general BEM operator K is given by

$$\mathbf{K}_h[i, j] = \langle K\psi_j^\gamma, \psi_i^\gamma \rangle_{\Gamma_{D,h}} = \int_{\text{supp}(\psi_i^\gamma)} \psi_i^\gamma(\mathbf{x}) \int_{\text{supp}(\psi_j^\gamma)} K(\mathbf{y} - \mathbf{x}) \psi_j^\gamma(\mathbf{y}) dS(\mathbf{y}) dS(\mathbf{x}) \quad . \quad (4.46)$$

It is, in general, unclear as to whether Eqn. 4.46 can be performed analytically. The integrals are normally computed using numerical quadratures. In general, this computation is far from trivial. In particular, the integral becomes singular when

$\text{supp}(\psi_i^\gamma) \cap \text{supp}(\psi_i^\gamma) \neq 0$. The supports may either share elements that are vertex adjacent (point singularity), edge adjacent (line singularity), or that are coincident (triangular singularity). Each case requires special care. In this thesis we utilize the Sauter Schwab quadratures [13], [58], to handle these singularities.

The kernel of the integral (shown in Eqn. 4.46) may be mapped to the reference element and be written as,

$$\hat{K}(\boldsymbol{\xi}^y - \boldsymbol{\xi}^x) = \psi_i^\gamma(\boldsymbol{\xi}^x) K(\chi_{\tau_y}(\boldsymbol{\xi}^y) - \chi_{\tau_x}(\boldsymbol{\xi}^x)) \psi_j^\gamma(\boldsymbol{\xi}^y) \sqrt{g_{\tau_x}(\boldsymbol{\xi}^x)} \sqrt{g_{\tau_y}(\boldsymbol{\xi}^y)} \quad . \quad (4.47)$$

Taken to quadrature over elements $\hat{\tau}_x$ and $\hat{\tau}_y$ that are in $\text{supp}(\psi_i^\gamma) \cup \text{supp}(\psi_j^\gamma)$, Eqn. 4.47 becomes,

$$I[\mathbf{K}] = \int_{\hat{\tau}_x} \int_{\hat{\tau}_y} \hat{K}(\boldsymbol{\xi}^y - \boldsymbol{\xi}^x) d\boldsymbol{\xi}^y d\boldsymbol{\xi}^x \approx \sum_{i=1}^n \sum_{j=1}^m \omega_i^x \omega_j^y \hat{K}(\boldsymbol{\xi}_i^y - \boldsymbol{\xi}_j^x) \quad , \quad (4.48)$$

where ω_i^x, ω_j^y are appropriately chosen quadrature weights and $\boldsymbol{\xi}_i^x, \boldsymbol{\xi}_j^y$ are appropriately chosen Gauss points on $\hat{\tau}_x$ and $\hat{\tau}_y$.

Throughout the rest of this thesis we use a simplified and highly convenient notation $K_{\Gamma_i \Gamma_j}$ to mean the full matrix form resulting from performing the computations shown in Eqn. 4.46 over all globally defined sets of test functions defined on the discrete mesh Γ_i and trial functions defined on the discrete mesh Γ_j for any general kernel K (not just the Stokes double layer potential).

4.1.6 Mass Matrix

The Mass Matrix appears when one calculates the inner product of a given function defined in the trial space, with a test function. In general we may expand some function $\phi(\mathbf{x}) = \sum_j^N \psi_j(\mathbf{x}) \phi^j$ where ϕ^j are nodal values of ϕ and the summation is taken over N nodes, $i \in \{1, \dots, N\}$. The inner product with a test function ψ_k may be written as $\langle \sum_j^N \psi_j(\mathbf{x}) \phi^j, \psi \rangle$ where the property of linearity yields the matrix expression $M \boldsymbol{\phi}_{i=1}^N$. The element of M at position $[k, l]$ is given as,

$$\mathbb{M}[k, l] = \int_{\text{supp}(\psi_k^\beta) \cap \text{supp}(\psi_l^\beta)} \psi_l^\beta(\mathbf{y}) \psi_k^\beta(\mathbf{y}) dS(\mathbf{y}) \quad . \quad (4.49)$$

The integration is performed over the intersection of supports for the global basis functions ψ_l and ψ_k . This matrix has a particularly convenient representation in the case of linear triangular elements. Since \mathbf{x} admits an expansion in the same basis functions, i.e. $\mathbf{x}(\boldsymbol{\xi}) = \sum_{j=1}^3 \psi_j(\boldsymbol{\xi}) \mathbf{x}^j$, one may exploit the linearity of the functions

$\psi_k(\mathbf{x})$ such that,

$$\begin{aligned}
 \psi_k(\mathbf{x}) &= \psi_k \left[\sum_{j=1}^3 \psi_j(\boldsymbol{\xi}) \mathbf{x}^j \right] \\
 &= \sum_{j=1}^3 \psi_j(\boldsymbol{\xi}) \psi_k(\mathbf{x}^j) \\
 &= \sum_{j=1}^3 \psi_j(\boldsymbol{\xi}) \delta_{jk} \\
 &= \psi_k(\boldsymbol{\xi})
 \end{aligned} \tag{4.50}$$

Using the fact that the gram matrix and hence the Jacobian determinant is constant for linear triangles, one may use Eqn. 4.50 in Eqn. 4.49 to find that,

$$\mathbb{M}[k, l] = \sqrt{g_{\tau_{\mathbf{y}}}(\boldsymbol{\xi}^{\mathbf{y}})} \int_{\text{supp}(\psi_k^\beta) \cap \text{supp}(\psi_l^\beta)} \psi_l^\beta(\boldsymbol{\xi}^{\mathbf{y}}) \psi_k^\beta(\boldsymbol{\xi}^{\mathbf{y}}) d\boldsymbol{\xi}^{\mathbf{y}} \tag{4.51}$$

In typical applications where one uses element by element assembly, Eqn. 4.51 reduces to a product of a constant local mass matrix times the gram determinant. We remark that the mass matrix shown here is a square matrix in the number of degrees of freedom on a boundary element mesh $|\Gamma_k^h|$. If this notion of the mass matrix, as an interpolation operator, is to be used for vector BIEs, its elements must be augmented by identity matrices.

4.2 The Galerkin Discretization Applied to Linearized Viscous Flows

Ultimately we are interested in studying a class of problems where a particle, which may be actively self-propelled, is placed inside of a closed container. The container is given its own set of properties and could, for example, be chosen to be incompressible, yet flexible, in which case it could serve as a representative model of a membrane bound vesicle. The CDL-BIEM formulation discussed in Chapter 3 is used to describe the particle's motion. We start by providing the discretization of this problem with application to the sedimentation problem in Stokes flow.

4.2.1 Discretization of the Completed Double Layer Representation

The starting equation is the boundary integral representation describing the motion of the particle,

$$\sum_{j=1}^6 \varphi_j^{RBM}(\mathbf{x}) \langle \varphi_j^{RBM}, \phi \rangle + \phi(\mathbf{x}) + \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) = -\mathbf{b}(\mathbf{x}) \quad (4.52)$$

$$\mathbf{b}(\mathbf{x}) = \mathbf{u}_0(\mathbf{x}) + \mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} \quad .$$

The derivation of this equation has been given previously in Chapter 3 Eqn. 3.71 and is not repeated here.

We proceed to discretize the system by constructing the variational form of Eqn. 4.52 with appropriate test and trial spaces. Let,

$$\phi(\mathbf{x}) \approx \phi_h^\beta(\mathbf{x}) = \sum_{l=1}^M \phi_l \psi_l^\beta(\mathbf{x}) \quad . \quad (4.53)$$

Inner product terms can be developed as follows,

$$\begin{aligned} \langle \varphi_j^{RBM}, \phi \rangle &= \int_{\Gamma} \phi(\mathbf{y}) \cdot \varphi_j^{RBM}(\mathbf{y}) dS(\mathbf{y}) \\ &= \int_{\Gamma} \left[\sum_l \phi_l \psi_l^\beta(\mathbf{y}) \right] \cdot \varphi_j^{RBM}(\mathbf{y}) dS(\mathbf{y}) \\ &= \int_{\Gamma} \left[\sum_l \phi_l \psi_l^\beta(\mathbf{y}) \right] \cdot \left[\sum_k \varphi_{jk}^{RBM} \psi_k^\beta(\mathbf{y}) \right] dS(\mathbf{y}) \\ &= \sum_l \sum_k \phi_l \cdot \varphi_{jk}^{RBM} \int_{\text{supp}(\psi_k^\beta)} \psi_l^\beta(\mathbf{y}) \psi_k^\beta(\mathbf{y}) dS(\mathbf{y}) \end{aligned} \quad . \quad (4.54)$$

Upon multiplication of Eqn. 4.52 by a test function, thus forming the variational form of the equations, we find that the RBM inner product in discretized form can be expressed as,

$$\begin{aligned} \langle \varphi_j^{RBM}, \psi_k^\beta \rangle &= \int_{\Gamma} \varphi_j^{RBM}(\mathbf{y}) \psi_k^\beta(\mathbf{y}) dS(\mathbf{y}) \\ &= \int_{\Gamma} \sum_l \varphi_{jl}^{RBM} \psi_l^\beta(\mathbf{y}) \psi_k^\beta(\mathbf{y}) dS(\mathbf{y}) \\ &= \sum_l \varphi_{jl}^{RBM} \int_{\text{supp}(\psi_k^\beta)} \psi_l^\beta(\mathbf{y}) \psi_k^\beta(\mathbf{y}) dS(\mathbf{y}) \end{aligned} \quad . \quad (4.55)$$

This implies that the terms $\langle \varphi_j^{RBM} \langle \varphi_j^{RBM}, \phi \rangle, \psi_k^\beta \rangle$ for $k \in \{1, \dots, M\}$ generates the matrix equation,

$$\begin{aligned} \langle \varphi_j^{RBM} \langle \varphi_j^{RBM}, \phi \rangle, \psi_k^\beta \rangle &= \langle \varphi_j^{RBM}, \psi_k^\beta \rangle \langle \varphi_j^{RBM}, \phi \rangle, \quad \forall k \\ &= \sum_{j=1}^6 (\mathbb{M} \varphi_j^{RBM}) (\varphi_j^{RBM} \cdot \mathbb{M} \phi) \end{aligned} \quad (4.56)$$

where φ_j^{RBM} and ϕ are stacked vectors holding respectively the nodal RBMs and the nodal potentials and \mathbb{M} is the mass matrix given by the construction shown in Eqn. 4.49.

Another term is needed in the iterative version of CDL-BIEM to map the eigenvalue +1 of the double layer operator, K , to zero. Consequently, this bounds the spectral radius of $|K| \in (-1, 1)$ making Picard iteration possible. Note that the mobility completion scheme has actually mapped the rank six eigenvalue at -1 to zero. We include the discretization of this term for completeness. The variational form of this extra term $\varphi^*(\mathbf{x}) \langle \varphi^*, \phi \rangle$, where φ^* is proportional to the surface normal vector, may be written as,

$$\begin{aligned} \langle \varphi^* \langle \varphi^*, \phi \rangle, \psi_k^\beta \rangle &= \langle \varphi^*, \psi_k^\beta \rangle \langle \varphi^*, \phi \rangle, \quad \forall k \\ &= (\mathbb{M} \varphi^*) (\varphi^* \cdot \mathbb{M} \phi) \end{aligned} \quad , \quad (4.57)$$

and the variational form of $\frac{1}{2} \varphi^*(\mathbf{x}) \langle \varphi^*, \mathbf{b} \rangle$ is,

$$\begin{aligned} \langle \frac{1}{2} \varphi^* \langle \varphi^*, \mathbf{b} \rangle, \psi_k^\beta \rangle &= \frac{1}{2} \langle \varphi^*, \psi_k^\beta \rangle \langle \varphi^*, \mathbf{b} \rangle, \quad \forall k \\ &= \frac{1}{2} (\mathbb{M} \varphi^*) (\varphi^* \cdot \mathbb{M} \mathbf{b}) \end{aligned} \quad . \quad (4.58)$$

Finally the single terms $\phi(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ can be written as,

$$\langle \phi, \psi_k^\beta \rangle = \mathbb{M} \phi \quad (4.59)$$

$$\langle \mathbf{b}, \psi_k^\beta \rangle = \mathbb{M} \mathbf{b} \quad . \quad (4.60)$$

Therefore in order to solve this iterative system one must also be able to evaluate nodal values of φ_j^{RBM} for each j , and the nodal values of φ^* and \mathbf{b} . The procedures used to derive Eqns. 4.56, 4.58, 4.59, 4.60 in combination with the general procedure for computing Galerkin matrix representations of BEM operators, Eqn. 4.46, are all that is needed to represent and solve a wide class of problems in multi-body hydrodynamics.

4.2.1.1 Single Particle Solution

If Picard iteration is not desired, one may choose instead to solve the system using explicit matrix inversion or an iterative solver. In reduced notation, the system to solve may be written as,

$$\left[\mathbb{M}_{\Gamma_p} + K_{\Gamma_p \Gamma_p} + \sum_{j=1}^6 \mathbb{M}_{\Gamma_p} \boldsymbol{\varphi}_j^{RBM} \left(\boldsymbol{\varphi}_j^{RBM} \cdot \mathbb{M}_{\Gamma_p} \right) \right] \times [\boldsymbol{\phi}_p] = -\mathbb{M}_p \mathbf{b}_p \quad , \quad (4.61)$$

where the system's stiffness matrix is given by the number of degrees of freedom on Γ_p given by $|\Gamma_p|$ multiplied by the dimension of the BEM kernel. Since the Stokes equations are vector equations, and the fundamental solutions are 2nd order tensors, the total system matrix size is $3|\Gamma_p| \times 3|\Gamma_p|$.

4.2.2 Multiple Particles and General Matrix Representations

The mobility problem for N particles of arbitrary shape follows as a pair-wise sum of interactions due to the linearity of the representative exterior BIE. In the variational formulation, each particle needs to know about its own density field, $\boldsymbol{\phi}_{\Gamma_p}$, its own set of pairwise $K_{\Gamma_p \Gamma_{p'}}$, and its own set of pairwise \mathcal{M}_{Γ_p} . If we observe the eigenfunction requirements that

$$\boldsymbol{\varphi}_{jp}^{RBM}(\mathbf{x}) = 0, \quad \forall \mathbf{x} \notin \Gamma_p \quad (4.62)$$

$$\boldsymbol{\varphi}_{jp}^*(\mathbf{x}) = 0, \quad \forall \mathbf{x} \notin \Gamma_p \quad . \quad (4.63)$$

It then follows that Eqn. 4.52 generalizes as,

$$\begin{aligned} \sum_{j=1}^6 \boldsymbol{\varphi}_{jp}^{RBM}(\mathbf{x}) \langle \boldsymbol{\varphi}_{jp}^{RBM}, \boldsymbol{\phi}_p \rangle + \boldsymbol{\phi}_p(\mathbf{x}) + \sum_{p'} (K_{pp'} \boldsymbol{\phi}_{p'}) (\mathbf{x}) - \boldsymbol{\varphi}_p^*(\mathbf{x}) \langle \boldsymbol{\varphi}_p^*, \boldsymbol{\phi}_p \rangle &= \mathbf{b}'(\mathbf{x}) \\ \mathbf{b}'(\mathbf{x}) &= -\mathbf{b}(\mathbf{x}) - \frac{1}{2} \boldsymbol{\varphi}_p^*(\mathbf{x}) \langle \boldsymbol{\varphi}_p^*, \mathbf{b} \rangle \\ \mathbf{b}(\mathbf{x}) &= \mathbf{u}_0(\mathbf{x}) + \sum_p \left[\mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} \right] \\ &\forall p \in \{1, \dots, N\} \end{aligned} \quad , \quad (4.64)$$

where we have given the full multi-body iterative version of CDL-BIEM construction.

Examining Eqn. 4.64 closely reveals that the multi-body formulation has a highly symmetric structure. For every particle p , there is a self interaction matrix given by $K_{\Gamma_p \Gamma_p}$. There are also $p-1$ pairwise interaction matrices given by $K_{\Gamma_i \Gamma_p}$, $\forall i \neq p$. The

key to the multi-body Galerkin discretization is realizing that the i 'th block matrix row is formed by *testing* of Eqn. 4.64 with global test functions taken from surface Γ_i . This means that the i 'th block matrix row is given by,

$$\begin{bmatrix} K_{\Gamma_1\Gamma_1}^s & \dots & K_{\Gamma_1\Gamma_i} & \dots & K_{\Gamma_1\Gamma_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{\Gamma_i\Gamma_1} & \dots & K_{\Gamma_i\Gamma_i}^s & \dots & K_{\Gamma_i\Gamma_N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K_{\Gamma_N\Gamma_1} & \dots & K_{\Gamma_N\Gamma_i} & \dots & K_{\Gamma_N\Gamma_N}^s \end{bmatrix} \times \begin{bmatrix} \phi_{\Gamma_1} \\ \vdots \\ \phi_{\Gamma_i} \\ \vdots \\ \phi_{\Gamma_N} \end{bmatrix} = \begin{bmatrix} (\mathbb{M}\mathbf{b}')_{\Gamma_1} \\ \vdots \\ (\mathbb{M}\mathbf{b}')_{\Gamma_i} \\ \vdots \\ (\mathbb{M}\mathbf{b}')_{\Gamma_N} \end{bmatrix}, \quad (4.65)$$

where $K_{\Gamma_i\Gamma_i}^s$ is the self GBEM interaction, given by,

$$K_{\Gamma_i\Gamma_i}^s = \mathbb{M}_{\Gamma_i} + K_{\Gamma_i\Gamma_i} + \sum_{j=1}^6 \mathbb{M}_{\Gamma_i} \varphi_j^{RBM} \left(\varphi_j^{RBM} \cdot \mathbb{M}_{\Gamma_i} \right), \quad (4.66)$$

and the off-diagonal elements are computed by straightforward applications of Eqn. 4.46 using the Stokes double layer potential. The matrix construction shown in Eqn. 4.65 can be generalized to include problems where one may choose to use different BIE's to model various portions of a system. In the next section we present this construction in the context of a particle in cell geometry. The key lies in exploiting the linearity of the stokes equations and writing the fluid velocity $\mathbf{u}(\mathbf{x})$ as,

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^c(\mathbf{x}) + \sum_p \mathbf{u}^p(\mathbf{x}) \quad , \quad (4.67)$$

where $\mathbf{u}^c(\mathbf{x})$ is the velocity contribution from the container and $\mathbf{u}^p(\mathbf{x})$ is the velocity contribution of particle p . The construction shown in 4.67 is highly general and may be used for an arbitrary number of bodies. Each velocity contribution may also be arbitrarily described, for example, by using a variety of different BEM representations.

4.3 Mobility and Resistance Formulation for Particle-in-Cell Models

We may decompose the fluid velocity as a sum of contributing velocities due to a single immersed particle and a surrounding container. The particle and container surfaces may be discretized using various types of 2-d finite elements, mapped to corresponding 2-manifolds in \mathbb{R}^3 . We label these surfaces, Γ_p and Γ_c respectively. The fluid velocity at some point, $\mathbf{x} \in \mathbb{R}^3$, is identified as belonging to a restriction

on \mathbb{R}^3 , that is generated by the union of the particle and container surfaces and is representative of the fluid domain, $\tilde{\Omega}$. We may write the fluid velocity as,

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^p(\mathbf{x}) + \mathbf{u}^c(\mathbf{x}) \quad . \quad (4.68)$$

It is then useful to define complementary velocity fields $\bar{\mathbf{u}}^p(\mathbf{x})$ and $\bar{\mathbf{u}}^c(\mathbf{x})$ such that,

$$\bar{\mathbf{u}}^p(\mathbf{x}) = \mathbf{u}(\mathbf{x}) - \mathbf{u}^p(\mathbf{x}) \quad (4.69)$$

$$\bar{\mathbf{u}}^c(\mathbf{x}) = \mathbf{u}(\mathbf{x}) - \mathbf{u}^c(\mathbf{x}) \quad , \quad (4.70)$$

from which one can more conveniently express velocity contributions relative to a complementary flow field. These complementary flow fields may be conveniently interpreted as disturbance flows.

4.3.1 Particle Contribution

In accordance with Eqn. 3.60, the velocity contribution of the particle may be written as,

$$\begin{aligned} \mathbf{u}^p(\mathbf{x}) &= \bar{\mathbf{u}}^c(\mathbf{x}) \\ &= \int_{\Gamma_p} K(\mathbf{x}, \mathbf{y}) \cdot \boldsymbol{\phi}(\mathbf{y}) dS(\mathbf{y}) + \mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} \quad , \end{aligned} \quad (4.71)$$

where $K(\mathbf{x}, \mathbf{y})$ is the Stokes double layer potential, $\boldsymbol{\phi}$ defines the pointwise distribution of K , $\mathbf{F}^{p,ext}$ and $\mathbf{T}^{p,ext}$ are the external forces and torques on the particle, $G(\mathbf{x}_c^p, \mathbf{x})$ is the Stokes single layer potential centered at the particle's centroid \mathbf{x}_c^p , and $R(\mathbf{x}_c^p, \mathbf{x}) \equiv \epsilon_{ijk} \frac{(y-x)_j}{r^3}$ is the Rotlet or Couplet singularity again centered at the particle's geometric centroid. Although we have completed the range of $(K\boldsymbol{\phi})(\mathbf{x})$ with a point force and point torque, we could have chosen other flows that would impart a finite force and torque to the body. Therefore the specific form of this range completer should not be considered a limitation of the method. In the limit that $\mathbf{x} \rightarrow \Gamma_p$ we find that,

$$\begin{aligned} \mathbf{U}^p + \boldsymbol{\Omega}^p \times (\mathbf{x} - \mathbf{x}_c) - \bar{\mathbf{u}}^p &= \boldsymbol{\phi}(\mathbf{x}) + \int_{\Gamma_{p:i}} K(\mathbf{x}, \mathbf{y}) \cdot \boldsymbol{\phi}(\mathbf{y}) dS(\mathbf{y}) \\ &+ \mathbf{F}^{p,ext} \cdot \frac{G(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^p, \mathbf{x})}{8\pi\mu} \cdot \mathbf{T}^{p,ext} \quad , \end{aligned} \quad (4.72)$$

where we have used that the particle moves as a rigid body. Note that the jump is usually $1/2$ but we have chosen to multiply K by a factor of 2 in order to move the eigenvalues of $(K\phi)(\mathbf{x})$ to $[-1, 1]$ for convenience.

4.3.2 Container Contribution

In formulating the container contribution we note that the container normal is taken to point out of the fluid domain $\tilde{\Omega}$, meaning that it is exterior to Γ_c . The velocity contribution of the container may be written as,

$$\begin{aligned} \mathbf{u}^c(\mathbf{x}) &= \bar{\mathbf{u}}^p(\mathbf{x}) \\ &= \int_{\Gamma_c} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) \end{aligned} \quad . \quad (4.73)$$

In the limit that $\mathbf{x} \rightarrow \Gamma_c$ we find that,

$$-\bar{\mathbf{u}}^c(\mathbf{x}) = -\phi(\mathbf{x}) + \int_{\Gamma_c} K(\mathbf{x}, \mathbf{y}) \cdot \phi(\mathbf{y}) dS(\mathbf{y}) - \varphi^*(\mathbf{x}) \langle \varphi^*, \phi \rangle, \quad \forall \mathbf{x} \in \Gamma_c \quad , \quad (4.74)$$

where the additional term involving ϕ^* acts to map the eigenvalue of the double layer at -1 to 0 and where ϕ^* is an eigenfunction of $(\mathbb{I} + K')$ proportional to the normal vector, \mathbf{n} .

4.3.3 Formulation and Solution of the System

The system of integral equations may be discretized by first meshing the surfaces Γ_p and Γ_c in accordance with some desired mesh metric, h . From here, ϕ may be expanded in a finite basis that may be taken to span the solution space. We choose to enforce a zero weighted residual condition on the system and arrive at the familiar Galerkin discretization whose solution is taken to belong to a subset of the half integer Sobolev space $\mathcal{H}^{1/2}(\Gamma)$. With the added mobility completion schemes, Eqns. 3.68, 3.69, and resistance completion schemes Eqns. 3.61, 3.62, the Galerkin discretizations and associated linear systems may be derived using straightforward application of the constructions shown in Section 4.2.1.

4.3.4 Mobility Problem

The linear system for the mobility problem may be written in the form,

$$\begin{bmatrix} (\mathbb{M}_{\Gamma_p} + K_{\Gamma_p\Gamma_p} + \sum_j (\mathbb{M}_{\Gamma_p} \boldsymbol{\varphi}_j) \boldsymbol{\varphi}_j \cdot \mathbb{M}_{\Gamma_p}) & K_{\Gamma_p\Gamma_c} \\ K_{\Gamma_c\Gamma_p} & (-\mathbb{M}_{\Gamma_c} + K_{\Gamma_c\Gamma_c} - (\mathbb{M}_{\Gamma_c} \boldsymbol{\varphi}^*) \boldsymbol{\varphi}^* \cdot \mathbb{M}_{\Gamma_c}) \end{bmatrix} \times \begin{bmatrix} \boldsymbol{\phi}_{\Gamma_p} \\ \boldsymbol{\phi}_{\Gamma_c} \end{bmatrix} = \begin{bmatrix} -B_{\Gamma_p} \\ -B_{\Gamma_c} \end{bmatrix}, \quad (4.75)$$

where B_{Γ_k} is defined as,

$$B_{\Gamma_k} \equiv \left[\mathbb{M} \left(\mathcal{G} \mathbf{F}^{p,ext} + \mathcal{R} \mathbf{T}^{p,ext} \right) \right]_{\Gamma_k}. \quad (4.76)$$

In post processing, the linear and angular velocities, $(\mathbf{U}^p, \boldsymbol{\Omega}^p)$, are solved from the mobility completion relationships given in Eqns. 3.68 and 3.69. These completion relationships are linear in both \mathbf{U}^p and $\boldsymbol{\Omega}^p$, and these quantities may be extracted by forming the inner product of 3.68 and 3.69 with each $\boldsymbol{\varphi}_i^{RBM}$. Assuming the RBM eigenfunctions are orthonormalized and letting $\mathbf{r} \equiv \mathbf{x} - \mathbf{x}_c^p$, one finds that,

$$\mathbf{U}^p = - \sum_{i=j}^3 (\boldsymbol{\varphi}_j^{RBM})_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} (\boldsymbol{\phi})_{\Gamma_p} \quad (4.77)$$

$$\boldsymbol{\Omega}^p = \begin{bmatrix} \left[(\epsilon_{ijk} r_k)_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} (\boldsymbol{\varphi}_4^{RBM})_{\Gamma_p} \right]^T \\ \left[(\epsilon_{ijk} r_k)_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} (\boldsymbol{\varphi}_5^{RBM})_{\Gamma_p} \right]^T \\ \left[(\epsilon_{ijk} r_k)_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} (\boldsymbol{\varphi}_6^{RBM})_{\Gamma_p} \right]^T \end{bmatrix}^{-1} \begin{bmatrix} -(\boldsymbol{\varphi}_4^{RBM})_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} \boldsymbol{\phi}_{\Gamma_p} \\ -(\boldsymbol{\varphi}_5^{RBM})_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} \boldsymbol{\phi}_{\Gamma_p} \\ -(\boldsymbol{\varphi}_6^{RBM})_{\Gamma_p} \cdot \mathbb{M}_{\Gamma_p} \boldsymbol{\phi}_{\Gamma_p} \end{bmatrix}. \quad (4.78)$$

4.3.5 Resistance Problem

The linear system for the resistance problem may be written in the form,

$$\begin{bmatrix} (\mathbb{M}_{\Gamma_p} + K_{\Gamma_p\Gamma_p} + B_{\Gamma_p}) & K_{\Gamma_p\Gamma_c} \\ K_{\Gamma_c\Gamma_p} + B_{\Gamma_c} & (-\mathbb{M}_{\Gamma_c} + K_{\Gamma_c\Gamma_c} - (\mathbb{M}_{\Gamma_c} \boldsymbol{\varphi}^*) \boldsymbol{\varphi}^* \cdot \mathbb{M}_{\Gamma_c}) \end{bmatrix} \times \begin{bmatrix} \boldsymbol{\phi}_{\Gamma_p} \\ \boldsymbol{\phi}_{\Gamma_c} \end{bmatrix} = \begin{bmatrix} [\mathbf{U} + \boldsymbol{\Omega} \times (\mathbf{x} - \mathbf{x}_c)]_{\Gamma_p} \\ \mathbf{0}_{\Gamma_c} \end{bmatrix}, \quad (4.79)$$

where B_{Γ_k} is defined as,

$$B_{\Gamma_k} \equiv \sum_{i=1}^3 \left[\mathbb{M}_{\Gamma_k} (\mathcal{G} \mathbf{e}_i)_{\Gamma_k} \boldsymbol{\varphi}_i^{RBM} \cdot \mathbb{M}_{\Gamma_p} + \mathbb{M}_{\Gamma_k} (\mathcal{R} \mathbf{e}_i)_{\Gamma_k} \boldsymbol{\varphi}_{i+3}^{RBM} \cdot \mathbb{M}_{\Gamma_p} \right]. \quad (4.80)$$

4.4 The GPUGBEM Framework

The GPUGBEM framework centers around shielding the user from having to create individual matrix blocks such as those seen in Eqns. 4.65, 4.75, or 4.79. To accomplish this, one must first identify the commonalities that all GBEM calculations rely on. All boundary element calculations derive from a discrete representation of an object's surface as a mesh. In addition, all objects require their own mass matrix and self-interaction matrix, regardless of the underlying BIE representation. If the CDL mobility description is used to represent an object, then that object also requires nodal orthonormalized rigid body motion vectors, φ_j^{RBM} , $j \in \{1, \dots, 6\}$, and a specification for the rigid body motions, namely $(\mathbf{U}, \mathbf{\Omega})$. The location of this matrix block must also be specified, but this may be related to a unique particle index label. All boundary element operators must also implement a set of functions describing how to compute self interactions, pairwise interactions, and the right hand side loading vector. This CDL mobility object may now serve as a base class from which other descriptions of particles may be built using inheritance. An example of this minimal type of base class is given below (note that the following code fragments are not meant to be complete compilable C++ codes),

```

class ObjectBase : public ObjectCounter< ObjectBase >{
public:    //public member functions
ObjectBase( )
: mesh_( ), id_( this -> live( ) - 1) { }
std::size_t GiveID( ) { return id_; }
std::size_t Rows( ) { return rows_; }
std::size_t Cols( ) { return cols_; }
//virtual functions...
virtual void Init( BEM* bem_computer_ptr ) = 0;
virtual void ComputeSelfInteraction( BEM* bem_computer_ptr,
                                     RigidBodyMotions* rbms_ ) = 0;
virtual void ComputePairwiseInteraction( BEM* bem_computer_ptr,
                                         DenseMatrix& K ) = 0;

virtual void ComputeRHS( ) = 0;
virtual NodalVector& GiveRHS( ) = 0;
protected: //protected data members
//mesh identifier used to place in matrix blocks (important)
std::size_t id_;
//the object's mesh
BEMMesh mesh_;
//interpolation matrix or "mass matrix"
SparseMatrix M_;
//dense "self-interaction" matrix

```

```

DenseMatrix Kself_;
//nodal orthonormalized rigid body motion vectors and
// normalization factors
NodalVector phi1_, phi2_, phi3_, phi4_, phi5_, phi6_;
NumericType nf1_, nf2_, nf3_, nf4_, nf5_, nf6_;
//rigid body motion information
Vector3d U_, Omega_;
//sizing for matrix items
std::size_t rows_, cols_;
}; // end of class ObjectBase

```

The point of the ObjectBase class is to illustrate how a base object may be used to hide explicit BEM computations from the user. All derived classes must implement the pure virtual functions. For example, in order to describe a squirmer, the self interaction must be described by Eqn. 4.66 and this computation in turn requires that one compute $K_{\Gamma_{sq}\Gamma_{sq}}$, all nodal rigid body motion vectors, and the mass matrix. Pairwise interactions are simply computed by calling a function which computes $K_{\Gamma_k\Gamma_{sq}}$ over the test space Γ_k . These pairwise matrix blocks are computed while fixing the base object's mesh as the *trial* space. This means that all pairwise interactions fill the rows of the column indexed by *id_* and are formed by calling ComputePairwiseInteraction over all test meshes in the system. The right hand side or load vector is computed as an interpolation over the squirming slip velocity \mathbf{u}^s . The class shown below shows how a swimmer object inherits from the base class and defines these functions generically.

```

class ObjectSwimmer
: public ObjectBase {
    ObjectSwimmer( const NumericType B1, const NumericType B2,
                  const Vector3d e, const std::size_t subdiv = 1 )
: squirmer_motion_( B1, B2, e ), rbd_( )
{
    //first build sphere
    this->mesh_.BuildSphere( subdiv );
    //allocate space for all of the structures
    rows_ = this->mesh_.NDoFs( ) * 3;
    cols_ = rows_;
    this->M_.resize( rows_, cols_ );
    this->Kself_.resize( rows_, cols_ );
    this->phi1_.resize( rows_ );
    this->phi2_.resize( rows_ );
    this->phi3_.resize( rows_ );
    this->phi4_.resize( rows_ );
    this->phi5_.resize( rows_ );
    this->phi6_.resize( rows_ );
    us_.resize( rows_ );
}

```



```

virtual NodalVectorRef GiveRHS( ) { return us_; }

private:    //private data members
//nodal squirming "slip-velocity"
NodalVector us_;
//generator for squirming slip velocity
SquirmMotion squirmer_motion_;
//generator for trajectory
RigidBodyDynamics rbd_;
}; // end of class ObjectSwimmer

```

This object base inheritance scheme is incredibly useful and modular as it wraps complicated and error prone boundary element calculations into clean particle objects that are attached to an underlying boundary integral equation representation in a derived class. In the GPUBEM framework, this inheritance scheme is used in representing all interacting bodies.

4.4.1 GPU GBEM Computations Using CUDA

We remark very briefly on ways to organize boundary element data so that calculations may be performed on GPUs. A boundary element calculation only requires knowledge of individual boundary elements in terms of local nodal orderings, global node labels, and the set of global nodal points. After copying this set of information over to the GPU the calculation may be carried out, at least in principle, by straight forward n-body simulation techniques, i.e. by mapping the outer loop in the well-known element-element assembly procedure to the individual CUDA cores. However, this technique only works well if all calculations are numerically equivalent in their number of operations. For singular quadrature in GBEM, this condition is grossly violated. Each singularity case requires different orders of quadrature. However, this situation may be resolved by first parsing the singularity types based on vertex-face 1-rings and then launching individual kernels for each group of elements having the same singularity type. In GPUBEM, this is precisely how self-interaction matrices are computed. We note that element-element assembly procedures naturally give rise to race conditions. These have to also be dealt with in an appropriate manner. One may solve this problem by using atomic operations when assembling at global matrix locations. This does not result in much of a slowdown since the number of CUDA cores accessing a particular global matrix location is only ever proportional to the number of faces in a vertices 1-ring.

4.4.2 h -Adaptive Meshing in GPUGBEM

In adaptive mesh refinement, one typically marks groups of elements for subdivision based on a heuristic (error measure/indicator). In this thesis this heuristic is loosely chosen to be a scaled distance ratio, which we refer to as the *nearly singular distance ratio*. This scaled distance ratio allows one to distinguish between cases where the straightforward application of regular Gauss quadrature may produce erroneous results. This situation tends to occur when two boundary elements, τ_x and τ_y , are in near contact. The formal statement of this situation goes as follows: let h_x denote the diameter of τ_x such that,

$$h_x = \sup_{\mathbf{x}, \mathbf{y} \in \tau_x} |\mathbf{x} - \mathbf{y}| \quad , \quad (4.81)$$

then the distance between elements τ_x and τ_y may be formally written as,

$$\text{dist}(\tau_x, \tau_y) = \min\{|\mathbf{y} - \mathbf{x}| : \mathbf{x} \in \tau_x, \mathbf{y} \in \tau_y\} \quad . \quad (4.82)$$

Defining the element diameter of τ_y in an analogous way, the nearly singular distance ratio is defined as,

$$d_{xy} = \frac{\text{dist}(\tau_x, \tau_y)}{\max\{h_x, h_y\}} \quad (4.83)$$

In situations where $d_{xy} \leq C$ for some distance cutoff C , experience has shown that application of regular Gauss quadrature will fail to accurately resolve these element contributions to the GBEM stiffness matrix. The idea is to then refine all element pairs (τ_x, τ_y) such that $d_{xy} \leq C$. In this thesis the distance cutoff is always set to 1. We find that this distance cutoff yields high quality results.

Element pairs (τ_x, τ_y) such that $d_{xy} \leq 1$ are refined using an extension of the Newest Vertex Bisection (NVB) method, [40], [41] to triangle 2-manifolds. The algorithm is coupled to various mesh smoothing routines which accurately project newly formed nodes onto the surface. The surface need not have an explicit parameterization. The computation of d_{xy} for all triangle pairs $(\tau_x, \tau_y) \in \Gamma_i \times \Gamma_j$ is naturally an $O(|\Gamma_i||\Gamma_j|)$ calculation (approximately $O(N^2)$ for N elements). GPUGBEM builds fast spatial querying structures on all meshes so that this calculation can be reduced to $O(N \log(N))$. We keep track of the element pairs $\{(\tau_x, \tau_y) \in \Gamma_i \times \Gamma_j : d_{xy} \leq 1\}$. These pairs of elements are subdivided using NVB. We note that the NVB algorithm does not introduce hanging nodes so long as a suitable compatibility labeling is constructed. In GPUGBEM, this compatibility labeling is constructed by finding longest edges of triangle pairs. This means that meshes which are initially conformal stay conformal. Keeping the mesh conformal is vital in maintaining the C^0 continuity requirements of the imposed continuous boundary element space, $S^{p,0}(\Gamma_k^h, \mathcal{X})$. The overall adaptive

mesh refinement algorithm for the GBEM stiffness matrix is given in Algorithm 1. An example output of the algorithm for two spheres in close contact is shown in Fig. 5.9. The algorithm in no way depends on a spherical geometry and works for arbitrarily shaped meshes.

Algorithm 1 Refinement List for Γ_s^h Given Γ_q^h

```

1: search_octree  $\leftarrow$  BuildSearchOctree( $\Gamma_s^h$ )
2: query_points  $\leftarrow$  BuildQueryPoints( $\Gamma_q^h$ )
3: for all qp  $\in$  query_points do
4:   sp  $\leftarrow$  SearchOctreeClosest(qp)
5:   sp_support  $\leftarrow$  DOFSupport(sp)
6:    $\{\tau_q, h_q\} \leftarrow$  GetElementAndSize(qp)
7:   for all  $\tau_s \in$  sp_support do
8:      $h_s \leftarrow$  GetElementSize( $\tau_s$ )
9:      $nsr \leftarrow$  dist{ $\tau_s, \tau_q$ }/max( $h_s, h_q$ )
10:    if  $nsr <$  dist_cutoff ||  $h_s/h_q >$  diam_cutoff then
11:      refine_list  $\leftarrow$   $\tau_s$ 
12:    end if
13:  end for
14: end for

```

Chapter 5

The Problem of an Active Particle Inside of a Porous Container

The motion of an active particle inside of a spherical container is an interesting and important problem for a multitude of reasons. Solution of this problem allows one to understand the unique fluid mechanics associated with active particles in confinement. Solving this problem also serves as a necessary step in understanding how viscous fluid mechanics may affect cellular biomechanics. For example, this problem could be modified in a straightforward way to study the dynamics and hydrodynamic interactions of motor proteins moving on microtubules inside of a container, [59]. As a first approximation, the motor proteins could be thought of as active particles undergoing constrained motion, tangent to the microtubule's axis of symmetry. The same fluid mechanical equations and boundary integral equations would govern the dynamics of these motor proteins, microtubules, and surrounding container. Solution of this system could be found using the same discretizations and algorithms. The active particle and porous container problem could also be used to help understand more complex problems involving viscous fluid structure interactions (FSI) between many arbitrarily shaped active particles and a surrounding container. The equations that model the particle and porous container system generalize under the construction shown in Chapter 4 Sec. 4.2.2 to systems with arbitrary particle geometries and heterogeneous boundary integral representations. In addition, the solution procedures, theory, and numerical routines presented in this chapter generalize directly to particles of arbitrary shape, with arbitrary physical properties. Attempting to resolve the detailed fluid mechanics inside of a biological cell is an exponentially more complex problem and task. By understanding the simpler case of how both viscous fluid mechanics and confinement influence the dynamics of the active particle, we work towards obtaining a more complete understanding of cellular particle dynamics.

The solution for the dynamics of the active particle and container motions not only reveals the effects of confinement on the motion of the active particle, but also the unique physical relationships between net particle and container translation velocities subject to container confinement effects and the imposed porous container boundary conditions. Since the particle-container geometry is axisymmetric, the creeping flow problem admits analytical solutions, so long as the geometry remains perfectly concentric. These analytical solutions may be used in validating numerical results and are particularly useful in designing numerical schemes that accurately approximate solutions when separate geometries are in close contact.

In this chapter, we first study the problem where an active particle, specifically a spherical squirmer [7], swims inside of a concentric, rigid, non-porous container. The quasi steady-state dynamics of this system are seen to decay over a much shorter timescale when compared to the case where the active particle is replaced with a forced particle. Next, analytical solutions are developed for a spherical squirmer inside of a concentric, rigid, porous container. A new porous container model is formulated allowing fluid to leak from the container in proportion to a discontinuous jump in stress across the container's surface. This stress jump is induced by the squirmer's active motion and is parameterized by a set of permeability resistance coefficients. These resistance coefficients may be different in both the tangential and normal directions relative to the container surface. We show that, when preference is given to permeability in the normal direction or the tangential direction, the container can only move, respectively, in an anti-parallel or parallel direction relative to squirmer's motion.

The analytical results are supplemented and validated with numerical results obtained using the Galerkin Boundary Element Method and the GPUGBEM framework. Details for this numerical method and its general construction are given in Chapter 4, but the constructions shown in this chapter are self-contained. Novelty is found in how the boundary integral equations are formulated and solved. Specifically, we verify that the forms of these Galerkin discretizations are correct and show that they produce highly accurate and stable solutions for the dynamics of the active particle and porous container system.

5.1 Contributions

1. A detailed fluid mechanical formulation is developed for describing the motion of an active particle, immersed in a Stokesian fluid, and confined by a surrounding

container. Full multi-body hydrodynamic interactions are accurately accounted for and included in all theoretical results and simulations.

2. Exact analytical mobility solutions are obtained for this problem when the particle and container geometry is perfectly concentric and when the container is non-porous.
3. The problem formulation is generalized to include a porous container by introducing a new model for describing fluid flow across the container's surface. This porous container model is a generalization of Darcy's law. Fluid flow across the container is driven by the motion of the active particle and is made proportional to the jump in stress across the container. New exact analytical solutions, in the context of flow fields and particle mobility solutions, are obtained for this particle and porous container geometry. All relevant fluid mechanical variables and datums are fully resolved (stress, pressure, and velocity).
4. A novel and new boundary integral representation of the problem is developed. A double layer representation is used to describe the active particle. A single layer potential is used together with the porous model to describe the motion of the container. The proposed porous model naturally gives rise to an integral equation of the second kind. These boundary integral equations describe both the fluid velocity at the container surface and the particle rigid body motions.
5. A novel interpretation of the Galerkin method is proposed and used to discretize, and produce highly accurate numerical solutions that show excellent agreement with the exact analytical models.
6. Numerical solutions are preformed which fully characterize the dynamics of this particle and porous container problem. Numerical results are obtained for a wide variety of container to particle size ratios b/a , permeability resistance parameters, squirmer types, and squirmer positions.
7. Thousands of squirmer trajectories are simulated and obtained using a novel, fast, and highly efficient algorithm that only relies on pre-computed fluid mechanical calculations. Squirmer and container dynamics are fully characterized by these trajectories. Single trajectories are obtained at a rate much faster than real-time. In addition we carry out full 3D trajectory analyses and answer the question, "Which type of squirmer traverses the length of the container the fastest?"

flow in Ω_i and Ω_e is governed by the steady Stokes equations,

$$\mu \nabla^2 \mathbf{u}^{i,e} = \nabla p^{i,e} \quad (5.1)$$

$$\nabla \cdot \mathbf{u}^{i,e} = 0 \quad . \quad (5.2)$$

Flow in the porous fluid region Ω_p may be described in various ways, the most useful of which is either macroscopic resolution using Darcy's law or Brinkman's equations, or be abstracted away completely in the form of a pressure gradient taken to be proportional to a membrane resistance parameter R_m and the relative motion of the fluid at the membrane.

The problem is then fully specified by switching to either a resistance, mobility, or a mixed formalism and asking questions, for instance, about the translational velocities of the squirmer and container under the action of known external forces. Since we are interested in the motion of the squirmer and container under force and torque free conditions we solve the mobility problem under the additional specification of a slip velocity, $\mathbf{u}^{sl}(\mathbf{x})$, on the squirmer's surface. The problem's geometric and spatial partitions are illustrated in Fig. 5.1.

5.2.1 Tangential Squirmers

In recent years, the squirmer model, originally formulated by Lighthill [37] and Blake [7], has become a workhorse for active matter calculations in low Reynolds number flows. In an important set of papers, Ishikawa and Pedley used the model to study the hydrodynamic interactions between two squirmers [27], and the rheology of both semi-dilute [26] and dense suspensions [25] of tangential squirmers. They made an important contribution by noticing that the velocity field of a tangential squirmer could be written in a form independent of basis. In this thesis we use this representation up to the first two squirming modes, B_1, B_2 .

In the Blake's squirmer model [7], the deformation field on the surface of the squirming particle is substituted with a slip velocity that is assumed to be both axisymmetric (spatially dependent only on θ) and time dependent. This slip velocity may be written in spherical coordinates as,

$$\mathbf{u}^{sl}(\theta, t) = u_r^{sl}(\theta, t) \mathbf{e}_r + u_\theta^{sl}(\theta, t) \mathbf{e}_\theta \quad . \quad (5.3)$$

The squirmer moves along its axis of symmetry or orientation, \mathbf{e}_s , and θ is the angle between this symmetry axis and a point $\mathbf{r} \equiv \mathbf{x} - \mathbf{x}_c^{sq}$ on the squirmer's surface. We note that Pak and Lauga [45] have recently used Lamb's general solution for the Stokes

equations to derive a more general type of spherical squirmer that does not assume axisymmetry and consequently has an additional azimuthal velocity component.

Theoretical studies of squirmers in contemporary literature usually examine the fluid mechanics and dynamics of purely tangential squirmers with time-independent slip velocities. These simplifications (or rather restrictions) of the squirmer boundary condition, Eqn. 5.3, imply that the stress close to the surface of the squirmer is purely tangential and that the related surface deformation field is time-averaged. While these simplifications may seem fairly restrictive, purely tangential and time-averaged motion at the squirmer boundary has been shown to be accurate in modeling ciliated organisms such as *Paramecia* and *Volvox*.

Tangential squirmers are also typically simplified under the further restriction of only using the first two expansion coefficients of u_θ in the representation of the squirmer's tangential surface deformation dynamics. These *squirming modes* are denoted as B_1 and B_2 from which the slip velocity may be written more compactly as,

$$\mathbf{u}^{sl}(\mathbf{r}^{sq}, \mathbf{e}_s) = B_1(1 + \beta(\mathbf{e}_s \cdot \hat{\mathbf{r}}^{sq})) [(\mathbf{e}_s \cdot \hat{\mathbf{r}}^{sq})\hat{\mathbf{r}}^{sq} - \mathbf{e}_s] \quad , \quad (5.4)$$

where $\beta = B_2/B_1$ is a ratio that describes the type of squirmer and $\hat{\mathbf{r}}^{sq} = \mathbf{r}/a$ is the radial unit vector on the surface of the squirmer. Far away from the squirmer the velocity field created by the first mode, B_1 , is that of a source dipole and decays as $O(r^{-3})$. The velocity field created by the second mode B_2 is that of a force dipole and so decays as $O(r^{-2})$. These two coefficients or squirming modes are sufficient to capture the far field dynamics of a variety of biological micro-squirmers. In the literature the sign of B_2 associates the names *pusher* and *puller* for respectively $B_2 < 0$ and $B_2 > 0$. When $B_2 = 0$ the squirmer is termed *neutral* and becomes a source dipole swimmer. The free space translational velocity of the squirmer may be calculated in various ways, one of which is by using the formalism presented in [65], in which case one finds that $\mathbf{U}_{fs}^{sq} = 2/3B_1\mathbf{e}_s$ and that the first tangential squirming mode provides the only nonzero contribution to this velocity.

In the following analysis, time-independent tangential squirmers in confinement are studied up to the first two squirming modes. We note that analogous solutions may be obtained for more complex squirmer boundary conditions, e.g. with the inclusion of an arbitrary number of radial and tangential squirming modes and also with the much less frequently used and less well know azimuthal squirming modes.

5.3 Analytical Solutions

The fluid mechanics problem of a squirming particle immersed in a viscous fluid and subject to the confinement effects of a surrounding container admits analytical solutions only when the container-particle geometry is perfectly concentric. The full flow field and motions of the particle and container may be obtained by several different approaches. For instance, one may use Lamb's general solution [36] in the form,

$$\mathbf{u} = \sum_{n=-\infty}^{\infty} \left[\nabla \times (\mathbf{r}\chi_n) + \nabla\Phi_n + \frac{(n+3)}{2\mu(n+1)(2n+3)} r^e \nabla p_n - \frac{n}{\mu(n+1)(2n+3)} \mathbf{r}p_n \right], \quad (5.5)$$

where χ_n , Φ_n and p_n are each solid spherical harmonics. The solution is then obtained when these three harmonic functions are determined, for each n , from the prescribed boundary conditions. For a Stokes fluid between concentric spheres, there are no conditions to be satisfied at the origin or at infinity and so the solution involves both positive and negative orders of the three harmonic functions. Thus there is, in general, six equations to be solved simultaneously for each n from which the harmonics in Eqn. 5.5 may be determined. Reigh and Lauga [55] have recently taken this approach and have studied the case in which a squirmer is moving inside of a viscous drop. In this thesis we prefer the use of a different solution method.

If it is assumed that the flow is axisymmetric, i.e. $\mathbf{u} = \mathbf{u}(r, \theta)$, the solution may also be obtained by following the standard stream function approach [19]. We fix the problem using a concentric geometry with a Cartesian basis having \mathbf{e}_Z pointing upwards and \mathbf{e}_Y pointing into the page. The swimming orientation \mathbf{e}_s is taken to point in the direction of \mathbf{e}_Z . The squirmer translational velocity is taken to be $\mathbf{U}^{sq} = U_z^{sq} \mathbf{e}_Z$ and the membrane velocity is $\mathbf{U}^m = U_z^m \mathbf{e}_Z$. Under the tangential squirmer simplifications, the stream function solution can be identified with the general form of the stream function for a translating sphere, although subject now to tangential slip induced by \mathbf{u}^{sl} . Solution of the problem is similar to that of the standard or traditional concentric hard spheres or viscous drop problems [19]. The stream functions in regions Ω_i and Ω_e may be written as,

$$\psi^i(r, \theta) = \left(A_1^i r^4 + A_2^i r^2 + A_3^i r + \frac{A_4^i}{r} \right) \sin^2(\theta) \quad (5.6)$$

$$\psi^e(r, \theta) = \left(A_1^e r^4 + A_2^e r^2 + A_3^e r + \frac{A_4^e}{r} \right) \sin^2(\theta) \quad (5.7)$$

from which the set of constants $\{A_j^i, A_j^e\}$ for $j \in \{1, \dots, 4\}$ is unknown. The fluid velocity in region Ω_e is assumed to decay to quiescence as $r \rightarrow \infty$ and so $A_1^e = A_2^e = 0$. The remaining unknown constants are determined by imposing six additional boundary conditions on the squirmer and container surfaces.

The boundary conditions on the squirmer surface are given by matching the squirmer's surface velocity to the fluid velocity at $r = a$. These conditions may be written as,

$$\left[\mathbf{u}^i(r, \theta) - \left(\mathbf{U}^{sq} + \boldsymbol{\Omega}^{sq} \times \mathbf{r} + \mathbf{u}^{sl}(r, \theta) \right) \right] \cdot \mathbf{e}_r \Big|_{r=a} = 0 \quad (5.8)$$

$$\left[\mathbf{u}^i(r, \theta) - \left(\mathbf{U}^{sq} + \boldsymbol{\Omega}^{sq} \times \mathbf{r} + \mathbf{u}^{sl}(r, \theta) \right) \right] \cdot \mathbf{e}_\theta \Big|_{r=a} = 0 \quad , \quad (5.9)$$

where \mathbf{U}^{sq} and $\boldsymbol{\Omega}^{sq}$ denote translational and angular velocities of the squirmer and $\mathbf{u}^{sl}(r, \theta)$ is the surface slip velocity (Eqn. 5.4) but written in spherical coordinates. Since we have assumed that the squirmer's surface has no azimuthal component, no net angular motion can be generated along \mathbf{e}_Z and so one can also write that $\boldsymbol{\Omega}^{sq} = \mathbf{0}$.

The squirmer experiences net translation along its orientation (in \mathbf{e}_Z). In the absence of fluid inertia, the force balance on a neutrally buoyant and net force and torque free particle reveals that the total hydrodynamic force is zero, $\mathbf{F}^H = \mathbf{0}$. In the context of the imposed boundary conditions 5.8 and 5.9, the net hydrodynamic force factors into two terms, namely the drag force, \mathbf{F}^D , due to resistive motion of the fluid acting against the instantaneous particle geometry and the propulsive force, \mathbf{F}^P , arising from the imposed slip velocity or surface deformation field. From a different point of view, the problem of squirming may be solved by superimposing the velocity fields due to the propulsive force (often called the *pumping problem*) and due to the rigid body motion (often called the *resistance problem*). However, in terms of the Boundary Element Method it is more efficient and often more useful to enforce the full nil constraint on the hydrodynamic force. The translational velocity of the squirmer is found by solving,

$$\mathbf{F}^H = \mathbf{0} \quad , \quad (5.10)$$

for \mathbf{U}^{sq} . Once $\psi^i(r, \theta)$ is determined, Eqn. 5.10 may be used to solve for \mathbf{U}^{sq} subject to the general and well-known formula,

$$F_z^H = \mu\pi \int_{\Gamma_{sq}} \left[(r \sin(\theta))^3 \frac{\partial}{\partial r} \left(\frac{E^2 \psi(r, \theta)}{(r \sin(\theta))^2} \right) r \right] \Big|_{r=a} d\theta \quad , \quad (5.11)$$

after which general expressions for velocities, pressures, and stresses may be recovered using standard formulas. When the container is made porous, this same analysis may be used to find the container velocity, U^m .

5.3.1 The Rigid Non-Porous Container

In the case of the non-porous rigid container, the fluid mechanics problems in regions Ω_i and Ω_e are fully decoupled. Moreover, a momentum balance on the fluid in region Ω_i reveals that the net force and torque on the container are equal and opposite to that experienced by the squirmer. Since the squirmer is assumed to be neutrally buoyant and thus net force and torque free, the container must also be net force and torque free. Therefore, the container does not move.

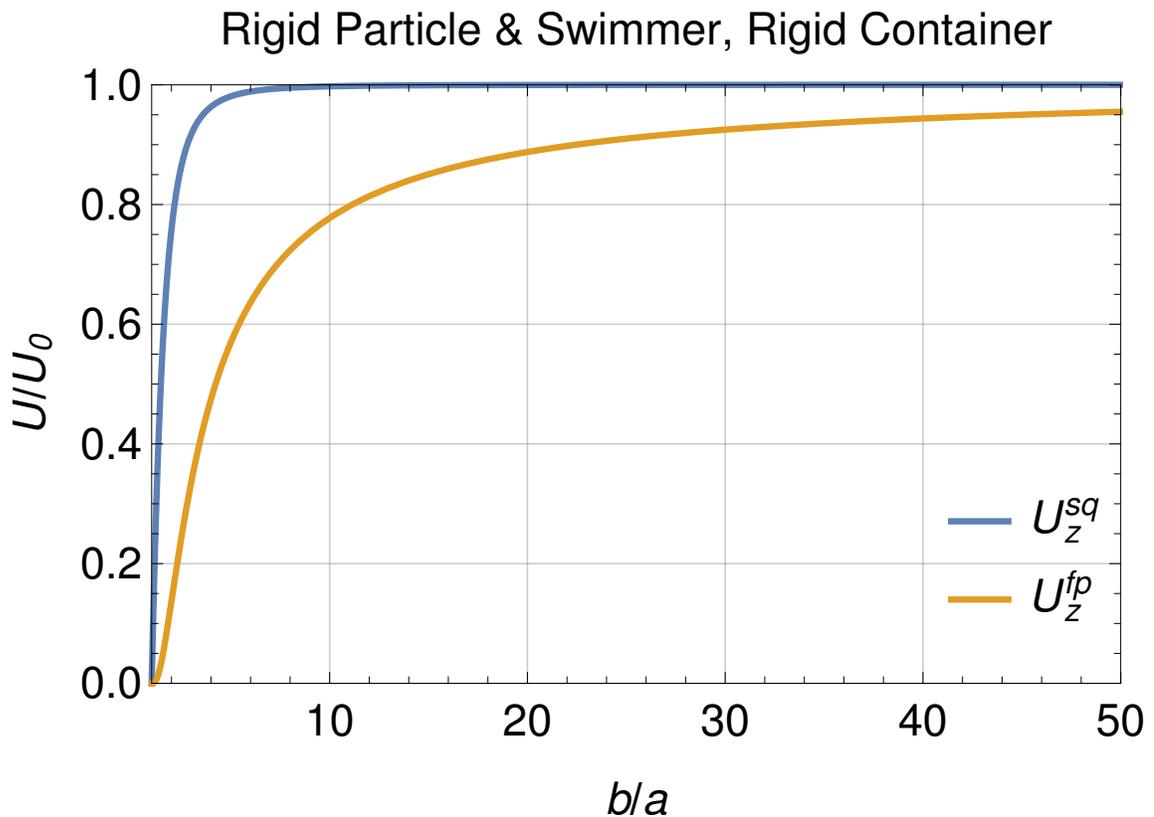


Fig. 5.2 The center of mass translation velocities of the squirmer, U_z^{sq} and a forced particle U_z^{fp} are plotted as a function of the container particle size ratio, b/a . The squirmer translational velocity approaches its free space squirming velocity fairly rapidly in comparison to the forced particle as $b/a \rightarrow \infty$.

The problem then reduces to specifying the standard kinematic and dynamic boundary conditions on the surface of the container, which are given by the vanishing

of the normal and tangential velocity components,

$$\psi^i(r, \theta) \Big|_{r=b} = 0 \quad (5.12)$$

$$\frac{\partial \psi^i(r, \theta)}{\partial r} \Big|_{r=b} = 0 \quad . \quad (5.13)$$

The system of equations for $\psi^i(r, \theta)$ is then fully specified subject to the boundary conditions defined in Eqns. 5.8, 5.9, 5.12, 5.13. This system of equations may be solved for the coefficients, set of constants $\{A_j^i, A_j^e\}$. Subject to the net force and torque free constraint, given in Eqn. 5.10, the squirmer translational velocity is found to be,

$$\mathbf{U}^{sq} = -\frac{B_1(a-b)(3a^3 + 6a^2b + 4ab^2 + 2b^3)}{3(a^4 + a^3b + a^2b^2 + ab^3 + b^4)} \mathbf{e}_z \quad (5.14)$$

Equation 5.14 together with the solution to the unit forced concentric sphere problem found in Happel and Brenner [19]. This is shown in Fig. 5.2

5.3.2 The Rigid Porous Container

The rigid porous container (rigid membrane) is a different problem in the sense that interface conditions between the fluid in interior and exterior regions now become coupled across the membrane. To proceed one must develop or accept an appropriate model which governs the fluid mechanics in the porous shell layer. There are only two possible viewpoints to take when developing a porous media flow model.

In the microscopic approach, one may apply the Stokes equations at the level of a characteristic porous matrix size $O(a^p)$, and resolve the fluid mechanics within the porous media structure explicitly using, for example, no-slip conditions on the membrane structure. However, this explicit resolution of the fluid mechanics in the interstitial porous regions is normally abandoned since it requires unrealistic amounts of computational time and prohibitive complexity in describing the geometry. Even if the microscopic approach is successfully carried out, it is also likely to produce locally oscillatory velocity and pressure data that would need smoothing for interpretation.

Alternatively in the macroscopic approach, one may try to express membrane structure in terms of effective parameters. Even for membranes with heterogeneity's in their porous structures, parameters such as the permeability can be described successfully with a variety of models of varying degrees of complexity. Techniques to obtain macroscopic models from microscopic flow solutions usually involve volume averaging in the porous media. Therefore the permeability is understood as an effective

parameter translating volume averaged fluid-solid quantities from a smaller scale to a scale often much larger than a^p .

Historically, Darcy's law was first empirically obtained based on macroscopic porous flow ideas. It may be written as,

$$\mathbf{u}^p = -\frac{K}{\mu}\nabla p^p \quad , \quad (5.15)$$

where it should be understood that the pressure, viscosity, and velocity are now volume averaged quantities and that K is a second order permeability tensor. Since the order of derivative in \mathbf{u}^p is one less than in the Stokes equations, one is forced to make an additional ad-hoc assumption on a tangential boundary condition. No-slip in the tangential direction is the easiest condition to enforce, however, extensive research on these boundary conditions has led to alternative tangential stress discontinuity conditions [6], [57], [28].

Today, it is also recognized that volume averaging of the Stokes equations leads one to conclude that the stationary form of Brinkman's equations governs the flow in the porous region. Brinkman's equations have been verified by the microscopic solution of the Stokes equations near the porous fluid interfaces [38]. Brinkman's equations may be written as,

$$\nabla p^p = \mu\nabla^2\mathbf{u}^p - \mu K^{-1} \cdot \mathbf{u}^p \quad (5.16)$$

$$\nabla \cdot \mathbf{u}^p = 0 \quad (5.17)$$

As long as there is no deformation, the permeability is a constant second order tensor, intrinsic to the porous medium. Since the Reynolds number in the porous medium now depends on a characteristic length a^p , which could for example be indicative of pore size or void space dimensions, there are situations where the viscous term $\mu\nabla^2\mathbf{u}^p$ is small relative to $\mu K^{-1} \cdot \mathbf{u}^p$. In these cases, Brinkman's equations reduce to Darcy's law. An extension of this work could be to study the dynamics with Brinkman's equations

To couple creeping flow with that in porous media one also needs to specify boundary condition at the fluid-porous interface. In general there are three conditions that must be prescribed on the interface.

1. A permeable interface must have continuity of the normal velocity. This is a simple consequence of assumed fluid incompressibility and thus simple mass conservation in the free fluid and porous fluid regions. In the coupling of the

fluid in region Ω_i to that in the porous region this condition takes the form,

$$\mathbf{u}^i \cdot \mathbf{n} = \mathbf{u}^p \cdot \mathbf{n} \quad (5.18)$$

2. A condition relating the pressures in the two fluids across the membrane interface has to be prescribed. From a fundamental perspective, this condition implies that flow is driven across the porous/permeable interface by a pressure gradient.
3. Finally, an additional condition on the tangential component of the fluid velocity at the interface must be specified. If one enforces the classic no-slip condition, then the tangential velocity of the fluid must be continuous across the fluid/-porous interface. The dynamic condition of no slip may be a good assumption for small permeabilities in which case there would be larger portions of the material interface between pores. However, for membranes with large pores, tangential transport of fluid across the membrane is more likely to occur.

We first consider a model where we 1) assume Darcy's law governs the fluid flow in Ω_p and 2) where ∇p^p may be approximated well by its first order finite difference. For a finite second order derivative in the pressure, this approximation is accurate to $O(l_c)$ where l_c is a length describing the thickness of our membrane. For thin membranes, this approximation is useful under the assumption that $l_c \ll \{a, b, b - a\}$. We then group the permeability parameters into an effective membrane resistance parameter $R_m \equiv l_c/K$. For pressure driven flow normal to the membrane, we remark here that $R_m \equiv R_\perp$. A similar driving force, although now related to a tangential viscous stress jump, may be used to account for tangential slip relative to the membrane. Although use of this condition is still debated, it provides the most simple model by which fluid can pass through the membrane with a finite tangential velocity. This construction is similar to that used by Jones [28]. Accordingly we use R_\parallel as a tangential effective permeability parameter to model tangential slip in our system. The kinematic boundary condition, Eqn. 5.18 is then supplemented with continuity in the tangential fluid velocity and normal and viscous traction jumps given by,

$$\mathbf{u}^i \cdot \mathbf{t} = \mathbf{u}^p \cdot \mathbf{t} \quad (5.19)$$

$$-p^e + p^i + \mu(\mathbf{n} \cdot \tau^e \mathbf{n} - \mathbf{n} \cdot \tau^i \mathbf{n}) = R_\perp \mu(\mathbf{u}^p - \mathbf{U}^m) \cdot \mathbf{n} \quad (5.20)$$

$$\mu(\mathbf{t} \cdot \tau^e \mathbf{n} - \mathbf{t} \cdot \tau^i \mathbf{n}) = R_\parallel \mu(\mathbf{u}^p - \mathbf{U}^m) \cdot \mathbf{t} \quad (5.21)$$

for unit normal and tangential vectors \mathbf{n}, \mathbf{t} . We supplement these conditions with $\mathbf{u}^i = \mathbf{u}^p = \mathbf{u}^e$ at $r = b$ which must hold for a thin membrane. Condition 5.21 deserves some additional comments. In particular, it results in discontinuous partial derivatives,

$$\left. \frac{\partial u_\theta^i}{\partial r} \right|_{r=b} \neq \left. \frac{\partial u_\theta^e}{\partial r} \right|_{r=b} . \quad (5.22)$$

This condition may be understood as one possible mechanism by which the squirmers tangential slip boundary condition is transmitted to the container.

5.3.3 Resolving Flow in the Porous Region

The system of algebraic equations, Eqns. 5.6, 5.7 subject to the boundary conditions, Eqns. 5.8, 5.9, 5.18, 5.19, 5.20, and 5.21 are now fully specified subject to the implied decay conditions in the exterior fluid. The system is solved simultaneously to obtain constants $\{A_j^i, A_j^e\}$ for $j \in \{1, \dots, 4\}$. Subject to the force and torque free constraint, Eqn. 5.10, the squirmer and membrane translational velocities may be obtained as,

$$\mathbf{U}^{sq} = \frac{B_1 \left(3a^5 R_\perp R_\parallel - 5a^3 b^2 R_\perp R_\parallel + 2b^5 R_\perp R_\parallel + 10b^4 (R_\perp + 2R_\parallel) \right)}{3 \left(a^5 (-R_\perp) R_\parallel + b^5 R_\perp R_\parallel + 5b^4 (R_\perp + 2R_\parallel) \right)} \mathbf{e}_z \quad (5.23)$$

$$\mathbf{U}^m = \frac{10a^3 b B_1 (R_\perp - R_\parallel)}{3 \left(a^5 (-R_\perp) R_\parallel + b^5 R_\perp R_\parallel + 5b^4 (R_\perp + 2R_\parallel) \right)} \mathbf{e}_z . \quad (5.24)$$

Both Eqns. 5.23 and 5.24 are independent of the second squirming mode B_2 and β . This means that for the concentric geometry, all types of squirming particles (pushers $\beta < 0$, neutral $\beta = 0$, and pullers $\beta > 0$) move with the same translational speed. It should be noted that the free space squirming velocity is also independent of B_2 and β and so there is no a priori reason that the fully axisymmetric particle-container should behave differently.

The resistance parameters $R = \{R_\parallel, R_\perp\}$ have units $1/L$ and so all resistances in all subsequent plots are nondimensionalized by multiplying with the particle size a . The translational velocity of the squirmer and container, \mathbf{U}^{sq} and \mathbf{U}^m , respectively given by Eqns. 5.23 and 5.24, are shown in Fig. 5.3 for various combinations of $R = \{R_\parallel, R_\perp\}$. When there is relatively weak resistance to normal flow across the membrane, $R_\parallel > R_\perp$, the squirmer moves opposite to the container. However, when $R_\parallel < R_\perp$ and tangential flow is favored, the container and squirmer both move in the same direction. In the situation where $R_\parallel < R_\perp$, flow across the membrane is primarily in the tangential direction and almost exactly mirrors the squirmer's tangential slip boundary condition. The squirmer is able to transmit its slip boundary condition onto the container walls by forming recirculating vortical flows which extend out beyond the container walls. This effectively turns the container into a larger version of a

tangential squirmer. On the other hand, when there is weak resistance to normal flow, the squirmer always pushes a net volume of fluid behind its body. Consequently, the passive porous container, which now allows this fluid to flow through its surface in the normal direction, is convected backwards with this net fluid motion. These effects are seen more clearly in the velocity field plots shown in the upcoming sections.

The models limiting behavior of $\{R_{\parallel}, R_{\perp}\} \rightarrow \{\infty, \infty\}$ reproduces the non-porous container solution, Eqn. 5.14 (note that these two curves overlap). Additionally, $U^{sq}(b/a)$ shows a global minimum implying that there is a special size ratio where the squirmer moves the slowest. This is understood as a confinement effect where the squirmer becomes inefficient at moving fluid tangentially around its body. Both U^{sq} and U^m decay away, respectively, to the free space squirmering velocity, $2/3B_1$, and 0 as $b/a \rightarrow \infty$.

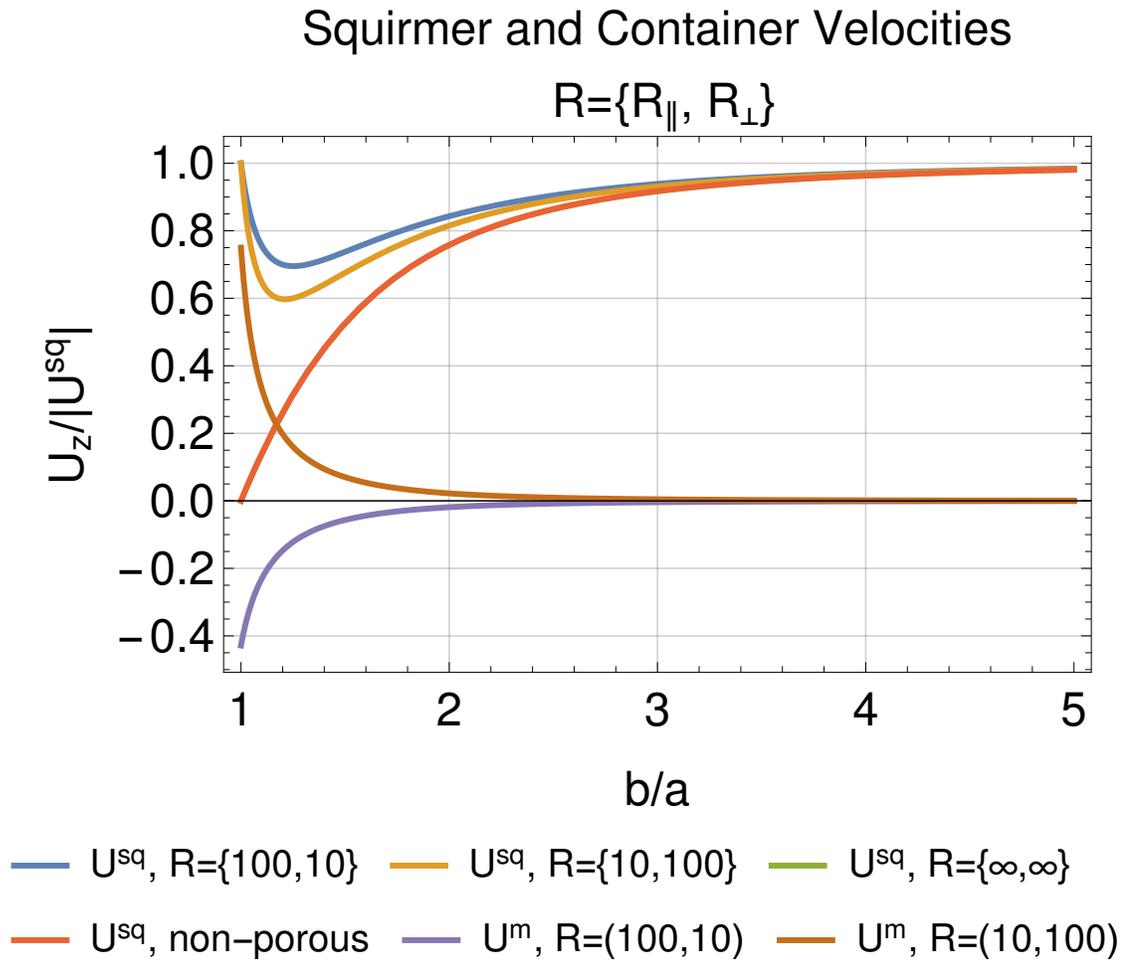


Fig. 5.3 The normalized z-component of the squirmer's and container's translational velocity, U_z^{sq} and U_z^m , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$

Additional plots, shown in Figs. 5.10, 5.11, 5.12, and 5.13, have been constructed for a wider variety of resistances. In general, a higher resistance leads to slower squirmer and container translational speeds.

5.3.3.1 Limiting Behaviors

Several important limiting behaviors may be recovered from Eqn. 5.23 and 5.24. Fixing the squirmer's size as a , if the size ratio $b/a \rightarrow \infty$, the squirming particle's speed is affected less and less by confinement and the free space squirming velocity, $2/3B_1$, is recovered with a container velocity that approaches zero. If $b/a \rightarrow 1$ or $a \rightarrow b$ then the squirmer perfectly transmits its boundary conditions to the container and the squirmer moves with the free space velocity $2/3B_1$ with the container translating at a speed reduced by $(R_\perp - R_\parallel)/(R_\perp + 2R_\parallel)$. Summarized more formally, for these two simple cases,

$$\lim_{b \rightarrow \infty} \mathbf{U}^{sq} = \frac{2}{3}B_1 \mathbf{e}_Z, \quad \lim_{b \rightarrow \infty} \mathbf{U}^m = \mathbf{0} \quad (5.25)$$

$$\lim_{b \rightarrow a} \mathbf{U}^{sq} = \frac{2}{3}B_1 \mathbf{e}_Z, \quad \lim_{b \rightarrow a} \mathbf{U}^m = \frac{2}{3}B_1 \left(\frac{R_\perp - R_\parallel}{R_\perp + 2R_\parallel} \right) \mathbf{e}_Z \quad . \quad (5.26)$$

In the case where fluid leaks out of the membrane isotropically, meaning that $R_\perp = R_\parallel = R$, the container does not translate, and the squirmer translates with a speed given by,

$$\lim_{R_\perp \rightarrow R_\parallel} \mathbf{U}^{sq} = \frac{B_1 (3a^5 R^2 - 5a^3 b^2 R^2 + 2b^5 R^2 + 30b^4 R)}{3(a^5 (-R^2) + b^5 R^2 + 15b^4 R)} \mathbf{e}_Z \quad . \quad (5.27)$$

The two cases in which the fluid is only permitted to leak normally or tangentially to the container results in two different container and squirmer velocities. This implies that mechanisms for normal and tangential fluid leakage are not equivalent in the sense that only discontinuities in the normal stress may be driven by a finite pressure jump. There is no corresponding equivalent mechanism to drive fluid tangentially across the membrane. These two limiting cases correspond to where either R_\perp or R_\parallel diverges while keeping the other permeability resistance finite. The container and

squirmer translational velocities become,

$$\lim_{R_{\perp} \rightarrow \infty} \mathbf{U}^{sq} = \frac{B_1 (3a^5 R_{\parallel} - 5a^3 b^2 R_{\parallel} + 2b^5 R_{\parallel} + 10b^4)}{3 (a^5 (-R_{\parallel}) + b^5 R_{\parallel} + 5b^4)} \mathbf{e}_Z \quad (5.28)$$

$$\lim_{R_{\perp} \rightarrow \infty} \mathbf{U}^m = \frac{10a^3 b B_1}{3 (a^5 (-R_{\parallel}) + b^5 R_{\parallel} + 5b^4)} \mathbf{e}_Z$$

$$\lim_{R_{\parallel} \rightarrow \infty} \mathbf{U}^{sq} = \frac{B_1 (3a^5 R_{\perp} - 5a^3 b^2 R_{\perp} + 2b^5 R_{\perp} + 20b^4)}{-3a^5 R_{\perp} + 3b^5 R_{\perp} + 30b^4} \mathbf{e}_Z \quad (5.29)$$

$$\lim_{R_{\parallel} \rightarrow \infty} \mathbf{U}^m = -\frac{10a^3 b B_1}{-3a^5 R_{\perp} + 3b^5 R_{\perp} + 30b^4} \mathbf{e}_Z$$

The limit in which both $R_{\perp} \rightarrow \infty$ and $R_{\parallel} \rightarrow \infty$ recovers the solution for the rigid non-porous container, Eqn. 5.14. The case in which $\{R_{\perp}, R_{\parallel}\} \rightarrow \{\infty, 0\}$ recovers a special case of a viscous drop, where there is no viscosity discontinuity, yet a zero shear stress jump at the membrane fluid interface. This is corroborated with comparison to Reigh's solution, [55]. In the limits that $R_{\perp} \rightarrow 0$ or $R_{\parallel} \rightarrow 0$ the squirmer velocity goes to the free space solution, $\mathbf{U}^{sq} \rightarrow 2/3 B_1 \mathbf{e}_Z$, yet the container moves at a reduced speed given by,

$$\lim_{R_{\perp} \rightarrow 0} \mathbf{U}^{sq} = \frac{2}{3} B_1 \mathbf{e}_Z, \quad \lim_{R_{\perp} \rightarrow 0} \mathbf{U}^m = -\frac{a^3 B_1}{3b^3} \mathbf{e}_Z \quad (5.30)$$

$$\lim_{R_{\parallel} \rightarrow 0} \mathbf{U}^{sq} = \frac{2}{3} B_1 \mathbf{e}_Z, \quad \lim_{R_{\parallel} \rightarrow 0} \mathbf{U}^m = \frac{2a^3 B_1}{3b^3} \mathbf{e}_Z \quad (5.31)$$

owing again to the difference in the whether a normal or shear stress discontinuity drives fluid across the membrane.

5.3.3.2 Velocity Fields

The full fluid velocity field may be constructed inside and outside of the container by using the well-known relationships between the Stokes stream function and the pointwise fluid velocity. In spherical coordinates these are given by,

$$u_r^{i,e}(r, \theta) = -\frac{1}{r^2 \sin(\theta)} \frac{\partial \psi^{i,e}(r, \theta)}{\partial \theta} \quad (5.32)$$

$$u_{\theta}^{i,e}(r, \theta) = \frac{1}{r \sin(\theta)} \frac{\partial \psi^{i,e}(r, \theta)}{\partial r} \quad (5.33)$$

In terms of numerics, the velocity fields are most efficiently constructed in Cartesian coordinates. Since our geometry is defined in the xz plane, standard mappings may be

used to convert Eqns. 5.32 and 5.33 to vector functions of (x, z, ϕ) . It is then a matter of evaluating these Cartesian vector functions pointwise and piecewise over the sets $\{\psi^i(x, z, \phi) : b^2 \geq x^2 + z^2 \geq a^2, \phi \in \{0, \pi\}\}$, and $\{\psi^e(x, z, \phi) : x^2 + z^2 > b^2, \phi \in \{0, \pi\}\}$. Streamlines may also be constructed using standard formalisms.

The velocity fields with streamlines have been constructed by sampling several parameters, namely the ratio b/a , the set of permeability resistances $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$, and squirmer types $\beta \in \{-5, 0, +5\}$. Figures 5.4, 5.14, 5.15, show the velocity fields for a $\beta = +5$ puller, a $\beta = 0$ neutral squirmer, and a $\beta = -5$ pusher respectively and for a size ratio of $b/a = 2$ and permeability parameterization $R_{\perp} = 100, R_{\parallel} = 10$. Subsequent Figures 5.22, 5.23, and 5.24, show a sweep over the same set of squirmer types but now with $R_{\perp} = 10$ and $R_{\parallel} = 100$.

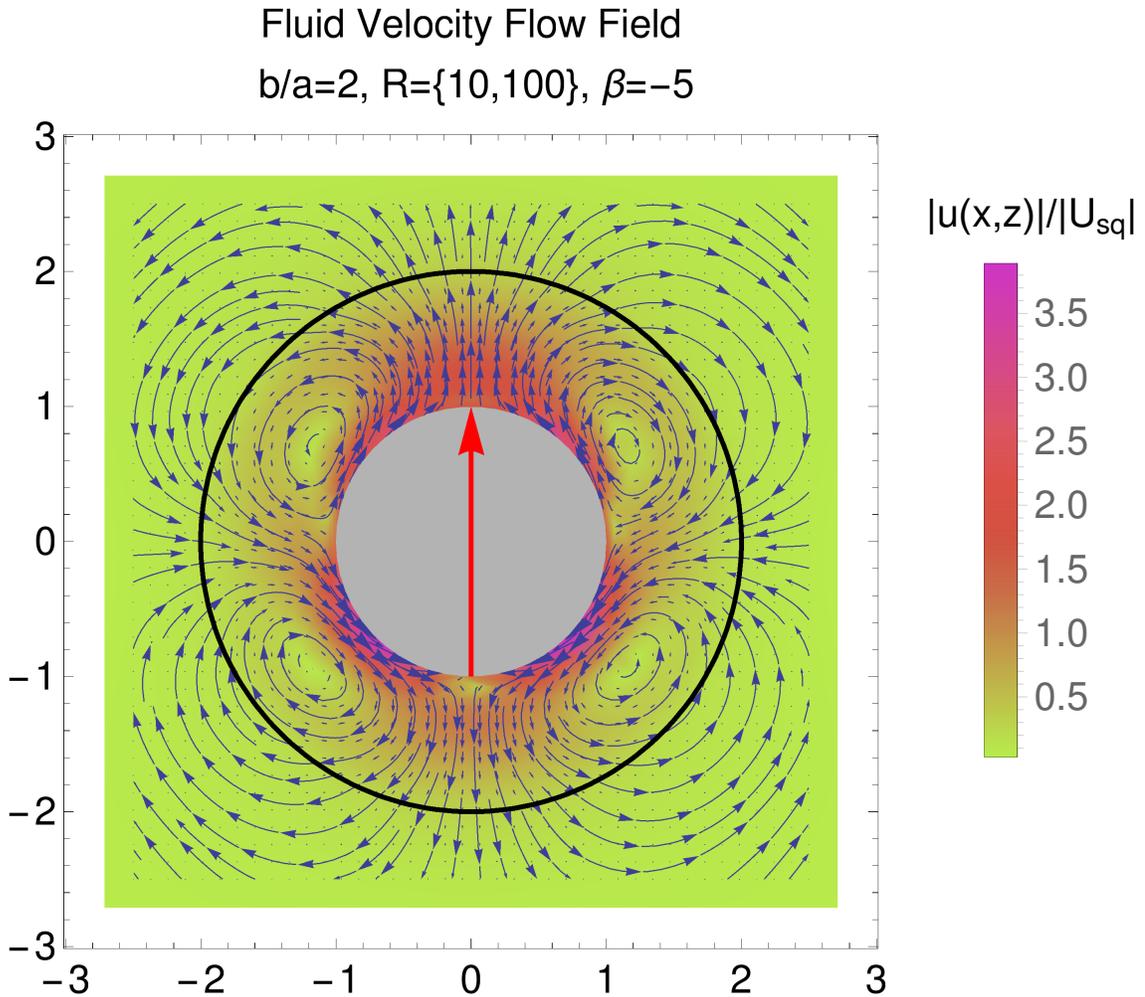


Fig. 5.4 The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

Figures {5.16, 5.17, 5.18, 5.25, 5.26, 5.27} and {5.19, 5.20, 5.21, 5.28, 5.29, 5.30} show sweeps over the same set of squirmers, i.e. $\beta \in \{-5, 0, +5\}$ and permeability resistances, though now with different size ratios of, respectively, $b/a = 5$ and $b/a = 10$.

Velocity fields parameterized by permeable resistances R_{\perp}, R_{\parallel} such that $R_{\perp} > R_{\parallel}$ show that the fluid exits preferentially in tangential directions to the membrane. Conversely the condition, $R_{\perp} < R_{\parallel}$ forces fluid flow to enter and exit the membrane with a larger normal component. It should be emphasized that the translational velocities of the squirmer and container, in the concentric geometry, are independent of the squirmer's type. However, the flow fields are very much dependent on β .

In particular, neutral squirmers are seen to produce flow fields that are very similar to the source dipole flow field. Pushers tend to draw fluid in radially and expel it out axially. On the other hand pullers draw fluid in axially and expel it out radially. Both pushers and pullers expel fluid asymmetrically relative to the xy plane, thus generating net motion along their orientation \mathbf{e}_s (in this case in $+\mathbf{e}_z$). Pushers and pullers should functionally behave like the point force dipole solution of Stokes flow. However, the container induces vortical flows in both the z -anterior and z -posterior portions of the container. This unique phenomena is purely of hydrodynamic origin owing to the confinement effects of the container.

5.4 Boundary Integral Formulation

Since the Stokes equations are linear, we may decompose the fluid velocity inside of the container as a sum of velocities arising from integral contributions from the squirmer with surface Γ_{sq} , and the container surface Γ_c . The fluid velocity at some point \mathbf{x} in the interior fluid region, Ω_i , is then given as,

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^{sq}(\mathbf{x}) + \mathbf{u}^c(\mathbf{x}), \quad \mathbf{x} \in \Omega_i \quad (5.34)$$

It is then useful to define complementary velocity fields $\bar{\mathbf{u}}^{sq}(\mathbf{x})$ and $\bar{\mathbf{u}}^c(\mathbf{x})$ such that,

$$\bar{\mathbf{u}}^{sq}(\mathbf{x}) = \mathbf{u}(\mathbf{x}) - \mathbf{u}^{sq}(\mathbf{x}) \quad (5.35)$$

$$\bar{\mathbf{u}}^c(\mathbf{x}) = \mathbf{u}(\mathbf{x}) - \mathbf{u}^c(\mathbf{x}) \quad , \quad (5.36)$$

from which one can more conveniently express velocity contributions relative to a complementary flow field or a disturbance flow.

5.4.1 Squirmer Contribution

Conditional on whether or not we want to solve for squirmer surface tractions, we may not necessarily choose the double layer representation for the squirmer. There is no straightforward way to develop a second-kind integral formulation for surface tractions by directly using the mobility Stresslet density. In fact, this procedure can only be carried out if the surface tractions are direct mappings from rigid body motion surface velocity fields [33], [32], [11]. Squirmers do not have boundary conditions solely dependent on no-slip rigid body motions, rather there is usually a modeled slip velocity that gives rise to some form of propulsive force. However, in the mobility problem, there is no need to solve for the surface tractions, unless, for example, we want to model a constant power squirmer. The solution here uses the complete double layer formulation to describe the active squirmer's motion under integral force free and torque free constraints. The velocity contribution due to the squirmer may be written as,

$$\mathbf{u}^{sq}(\mathbf{x}) = \int_{\Gamma_{sq}} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) dS(\mathbf{y}) + \mathbf{F}^{sq,ext} \cdot \frac{G(\mathbf{x}_c^{sq}, \mathbf{x})}{8\pi\mu} \quad (5.37)$$

$$\begin{aligned} & - \frac{1}{2} \left(\mathbf{T}^{sq,ext} \times \nabla_{\mathbf{x}} \right) \cdot \frac{G(\mathbf{x}_c^{sq}, \mathbf{x})}{8\pi\mu} \\ & = \int_{\Gamma_{sq}} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) dS(\mathbf{y}) + \mathbf{F}^{sq,ext} \cdot \frac{G(\mathbf{x}_c^{sq}, \mathbf{x})}{8\pi\mu} \quad (5.38) \\ & \quad + \frac{R(\mathbf{x}_c^{sq}, \mathbf{x})}{8\pi\mu} \mathbf{T}^{sq,ext} \quad , \end{aligned}$$

where $K(\mathbf{x}, \mathbf{y})$ is the Stokes double layer potential multiplied by a factor of 2, $\phi(\mathbf{y})$ is the double layer density, $\mathbf{F}^{sq,ext}$ and $\mathbf{T}^{sq,ext}$ are the external force and torque on the squirmer (usually set to zero), $G(\mathbf{x}_c^{sq}, \mathbf{x})$ is the Stokes single layer potential centered at the squirmer's centroid \mathbf{x}_c^{sq} , and $R(\mathbf{x}_c^{sq}, \mathbf{x}) \equiv \epsilon_{ijk} \frac{(\mathbf{x} - \mathbf{x}_c^{sq})_j}{r^3}$ is the Rotlet or Couplet singularity again centered at the squirmers geometric centroid. Although we have completed the range of $(\mathcal{K}\phi)(\mathbf{x})$ with a point force and point torque, we could have chosen other flows that would impart a finite force and torque.

In the limit that $\mathbf{x} \rightarrow \Gamma_{sq}$ we find the squirmer disturbance flow to be,

$$\begin{aligned} & \mathbf{U}^{sq} + \boldsymbol{\Omega}^{sq} \times (\mathbf{x} - \mathbf{x}_c) \\ & + \mathbf{u}^{sl}(\mathbf{x}) - \bar{\mathbf{u}}^{sq}(\mathbf{x}) = \phi(\mathbf{x}) + \int_{\Gamma_{sq}} K(\mathbf{x}, \mathbf{y}) \varphi(\mathbf{y}) dS(\mathbf{y}) \quad , \quad (5.39) \\ & \quad + \mathbf{F}^{sq,ext} \cdot \frac{G(\mathbf{x}_c^{sq}, \mathbf{x})}{8\pi\mu} + \frac{R(\mathbf{x}_c^{sq}, \mathbf{x})}{8\pi\mu} \mathbf{T}^{sq,ext} \end{aligned}$$

where we have used the no-slip boundary condition coupling the fluid velocity on the surface of the particle to the particle's rigid body motion. An additional known slip velocity, $\mathbf{u}^{sl}(\mathbf{x})$, is prescribed and is solely responsible for generating a propulsive force that drives the body. For a force and torque free swimmer the range completion terms drop out of the CDL formulation and the swimmer may be fully described by the double layer potential,

$$\mathbf{U}^{sq} + \mathbf{\Omega}^{sq} \times (\mathbf{x} - \mathbf{x}_c) + \mathbf{u}^{sl}(\mathbf{x}) - \bar{\mathbf{u}}^{sq}(\mathbf{x}) = \phi(\mathbf{x}) + \int_{\Gamma^{sq}} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) dS(\mathbf{y}) \quad . \quad (5.40)$$

Equation 5.40 is likely the most efficient way to represent active matter problems since force and torque free constraints are automatically satisfied when using the double layer representation. Although the singularity present in the kernel K is proportional to $1/r^2$, Eqn. 5.40 is a second kind integral equation and admits rigorous analysis under the Fredholm-Riesz-Schauder theory, [22], [35].

With additional modifications, Eqn. 5.40 may be written solely in terms of the potential ϕ . Using the well known completion relationships [31],

$$\mathbf{U}^{sq} = - \sum_{j=1}^3 \varphi^{j,RBM}(\mathbf{x}) \langle \varphi^{j,RBM}, \phi \rangle \quad (5.41)$$

$$\mathbf{\Omega}^{sq} \times (\mathbf{x} - \mathbf{x}_c^{sq}) = - \sum_{j=4}^6 \varphi^{j,RBM}(\mathbf{x}) \langle \varphi^{j,RBM}, \phi \rangle \quad (5.42)$$

the eigenvalue -1 of rank six, corresponding to the operator $(\mathbb{I} + \mathcal{K})$, is mapped to zero. The associated eigenfunctions correspond to the six different and linearly independent rigid body motions, $\varphi^{i,RBM}, i \in \{1, \dots, 6\}$, made orthonormal using the modified Gram-Schmidt process. The completion relationships, 5.41, 5.42 may be combined with Eqn. 5.40 viz.,

$$\begin{aligned} \mathbf{u}^{sl}(\mathbf{x}) - \bar{\mathbf{u}}^{sq}(\mathbf{x}) = \phi(\mathbf{x}) + \int_{\Gamma^{sq}} K(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) dS(\mathbf{y}) \\ + \sum_{j=1}^6 \varphi^{j,RBM}(\mathbf{x}) \langle \varphi^{j,RBM}, \phi \rangle \quad . \quad (5.43) \end{aligned}$$

Upon solution of ϕ , post processing with the completion relations, Eqns. 3.68, 3.69, allows recovery of \mathbf{U}^{sq} and $\mathbf{\Omega}^{sq}$.

5.4.2 Container Contribution

In formulating the container contribution we make explicit indications of where \mathbf{x} and \mathbf{y} are located relative to the defined container normal vector \mathbf{n} . Since we naturally take the container normal to point exterior to the container (and in general exterior to a closed geometry), we denote quantities that depend on points \mathbf{x} and \mathbf{y} , in the volume Ω_e and in the direction of \mathbf{n} , with superscript $+$. We analogously denote quantities that depend on \mathbf{x} and \mathbf{y} as points in the volume, Ω_i bounded by $\Gamma_{c-} \cup \Gamma_{p+}$ with superscript $-$. This allows for more efficient notation to be used for the construction of stress jumps across the container surface.

Under zero viscosity contrast inside and outside of the container, we find that,

$$\begin{aligned} \mathbf{u}^c(\mathbf{x}) - \bar{\mathbf{u}}^c(\mathbf{x}) &= \frac{1}{2} [\mathbf{u}^+(\mathbf{x}) - \mathbf{u}^-(\mathbf{x})] + \int_{\Gamma_c} K(\mathbf{x}, \mathbf{y}) [\mathbf{u}^+(\mathbf{y}) - \mathbf{u}^-(\mathbf{y})] dS(\mathbf{y}) \\ &\quad - \int_{\Gamma_c} G(\mathbf{x}, \mathbf{y}) [\mathbf{f}^+(\mathbf{y}) - \mathbf{f}^-(\mathbf{y})] dS(\mathbf{y}) \end{aligned} \quad (5.44)$$

If we assume that the fluid velocity is continuous across the membrane, as would be the case for pressure driven flow through a cylindrical pore (i.e. fully developed Hagen-Poiseuille flow), $\mathbf{u}^+(\mathbf{x}) = \mathbf{u}^-(\mathbf{x})$, the equations then reduce to,

$$\mathbf{u}^c(\mathbf{x}) - \bar{\mathbf{u}}^c(\mathbf{x}) = - \int_{\Gamma_c} G(\mathbf{x}, \mathbf{y}) [\mathbf{f}^+(\mathbf{y}) - \mathbf{f}^-(\mathbf{y})] dS(\mathbf{y}) \quad (5.45)$$

We emphasize that $\mathbf{u}^c(\mathbf{x})$ is the fluid velocity evaluated on the container / membrane surface and is different from the membrane point-wise velocity (to be denoted as $\mathbf{u}^m(\mathbf{x})$) unless no-slip is used in conjunction with the standard kinematic boundary condition. To proceed, we construct a constitutive law for the stress jump across the membrane. For the jump in stress a general form holds,

$$\begin{aligned} \Delta \mathbf{f}(\mathbf{y}) &\equiv [\mathbf{f}^+(\mathbf{y}) - \mathbf{f}^-(\mathbf{y})] \\ &= - [\mathbf{f}^b(\mathbf{y}) + \mathbf{f}^\gamma(\mathbf{y}) + \mathbf{f}^p(\mathbf{x})] \end{aligned} \quad (5.46)$$

which includes forces related to membrane bending \mathbf{f}^b , tension \mathbf{f}^γ , and resistance to permeability \mathbf{f}^p . More generally, the container surface Γ_c may be modeled as an incompressible 2D fluid. For instance, if one chooses to model a lipid bilayer, the Canham-Helfrich energy may be used to model the bending energy. The surface incompressibility constraint is often imposed by using a Lagrange multiplier associated with in-plane membrane tension. First variations in the Canham-Helfrich energy yield

bending and tension force densities [8],

$$\mathbf{f}^b(\mathbf{y}) = E_b(2\Delta_s H + 2H(H^2 - K))\mathbf{n}(\mathbf{y}) \quad (5.47)$$

$$\mathbf{f}^\gamma(\mathbf{y}) = -2\gamma H\mathbf{n}(\mathbf{y}) + \nabla_s \gamma \quad . \quad (5.48)$$

However, restricting our attention to a rigid membrane, we allow the membrane to only support a permeable resistance force density and define,

$$\begin{aligned} \mathbf{f}^p(\mathbf{y}) &= -[\mathbf{u}^c(\mathbf{y}) - \mathbf{u}^m(\mathbf{y})] \cdot [R_n \mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y}) + R_b \mathbf{b}(\mathbf{y})\mathbf{b}(\mathbf{y}) + R_t \mathbf{t}(\mathbf{y})\mathbf{t}(\mathbf{y})] \\ &= -\{\mathbf{u}^c(\mathbf{y}) - [\mathbf{U}^m + \boldsymbol{\Omega}^m \times (\mathbf{y} - \mathbf{x}_c^m)]\} \\ &\quad \cdot \{R_n \mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y}) + R_b \mathbf{b}(\mathbf{y})\mathbf{b}(\mathbf{y}) + R_t \mathbf{t}(\mathbf{y})\mathbf{t}(\mathbf{y})\} \end{aligned} \quad , \quad (5.49)$$

in analogy with Darcy's law. This model is equivalent to the analytical formulation under the identifications $R_n \equiv R_\perp$, $R_\parallel \equiv R_b \equiv R_t$. The vectors \mathbf{t} , \mathbf{b} , \mathbf{n} are respectively the tangent, bitangent, and normal vectors and provide a localized point-wise tangent space on the container surface.

In addition, integral constraints on the total hydrodynamic force and torque on the membrane container must be made to fix the system,

$$\int_{\Gamma_c} \Delta \mathbf{f}(\mathbf{y}) dS(\mathbf{y}) = \mathbf{F}^h = \mathbf{0} \quad (5.50)$$

$$\int_{\Gamma_c} (\mathbf{y} - \mathbf{x}_c^c) \times \Delta \mathbf{f}(\mathbf{y}) dS(\mathbf{y}) = \mathbf{T}^h = \mathbf{0} \quad . \quad (5.51)$$

We remark here on a subtlety in connecting the analytical and numerical models. By definition, first-order derivatives (i.e. \mathbf{t} , \mathbf{b}) only make sense when there is a global parameterization assigned to the mesh. Applying normal and tangential stress jump boundary conditions therefore requires consistent definitions of tangent spaces that must originate from making differential geometric estimates on a globally parametric mesh. If the mesh is nonparametric, after determining a normal vector, the tangent vector space can be approximated using a local parameterization, e.g. a mapping $(s, t) \rightarrow x_0 + s\mathbf{t} + t\mathbf{b}$. In what follows, we parameterize with spherical coordinates, making the identifications $\mathbf{t} \equiv \mathbf{e}_\theta$ and $\mathbf{b} \equiv \mathbf{e}_\phi$ in a body coordinate system attached to the container center.

5.5 Galerkin Discretization

The system of equations may be discretized by constructing the variational forms of Eqns. 5.43 and 5.45 subject to the integral constraints, Eqns. 5.50, 5.51. In what

follows we assume that our surfaces Γ_{sq}, Γ_c are piecewise smooth and admit conformal mesh decompositions $\Gamma_{sq}^h, \Gamma_c^h$. It is assumed that any given 2-manifold Γ_k may be decomposed into a mesh Γ_k^h ,

$$\Gamma_k \approx \Gamma_{k,M}^h = \bigcup_{m=1}^M \bar{\tau}_m \quad (5.52)$$

with M boundary elements τ_m and where h is a globally defined mesh element width. In what follows, for convenience, the M subscript is dropped under the assumption that an appropriate global indexing set has been constructed with cardinality M .

5.5.1 Continuous Boundary Element Spaces

In general we may construct a space of continuous boundary elements of order p on our discrete mesh Γ_k^h as,

$$S^{p,0}(\Gamma_k^h, \boldsymbol{\chi}) \equiv \{\psi \in C^0(\Gamma_k) \mid \forall \tau \in \Gamma_k^h : \psi|_{\tau} \circ \chi_{\tau} \in \mathbb{P}_p^{\tau}\} \quad , \quad (5.53)$$

where $\boldsymbol{\chi} = \{\chi_{\tau} : \tau \in \Gamma_k^h\}$ is an element mapping vector and \mathbb{P}_p^{τ} is an order p element polynomial space. The element mapping $\chi_{\tau}(\boldsymbol{\xi})$ given by $\chi_{\tau} : \hat{\tau} \subset \mathbb{R}^2 \rightarrow \tau \subset \mathbb{R}^3$ is defined using the typical parametric element mapping, which in the case of flat planar triangles is also an affine mapping,

$$\begin{aligned} \chi_{\tau}(\boldsymbol{\xi}) &= \sum_{i=0}^2 \mathbf{p}_i \psi_i(\boldsymbol{\xi}) \\ \chi_{\tau}(\boldsymbol{\xi}) &= \mathbf{p}_0 + \xi_1(\mathbf{p}_1 - \mathbf{p}_0) + \xi_2(\mathbf{p}_2 - \mathbf{p}_1) \end{aligned} \quad (5.54)$$

The triangular reference element is defined as,

$$\hat{\tau} \equiv \{\boldsymbol{\xi} \in \mathbb{R}^2 : 0 \leq \xi_2 < \xi_1 \leq 1\} \quad , \quad (5.55)$$

where $\{\hat{\mathbf{p}}_0, \hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2\}$ are the corners of the domain $\hat{\tau}$ and identify with their lifts into \mathbb{R}^3 as $\chi_{\tau}(\hat{\mathbf{p}}_j) = \mathbf{p}_j$. In general, Lagrangian type shape functions for this domain, may be built by inverting a generalized Vandermonde matrix. In doing so it is useful to define a local nodal indexing set,

$$I_p^{\hat{\tau}} \equiv \{(i, j) \in \mathbb{N}_0^2 \times \mathbb{N}_0^2 : 0 \leq i \leq j \leq p\} \quad , \quad (5.56)$$

where for 3 node triangles $I_1^{\hat{\tau}} = \{(0, 0), (0, 1), (1, 1)\}$. Points are given by $\hat{\mathbf{p}}_{(i,j)}^p = (i/p, j/p)$, $\forall (i, j) \in I_p^{\hat{\tau}}$. The set of shape or basis functions for flat 3-node triangles is

simple, and may be written as,

$$\psi_0(\boldsymbol{\xi}) = 1 - \xi_1 \quad (5.57)$$

$$\psi_1(\boldsymbol{\xi}) = \xi_1 - \xi_2 \quad (5.58)$$

$$\psi_2(\boldsymbol{\xi}) = \xi_2 \quad (5.59)$$

with the property,

$$\psi_i(\hat{\mathbf{p}}_j) = \delta_{ij} \quad , \quad (5.60)$$

where we have explicitly identified the nodes with the triangle vertices. In particular, and for simplicity, we expand our unknowns, $\{\boldsymbol{\phi}, \mathbf{u}^c, \mathbf{U}^m, \boldsymbol{\Omega}^m\}$, in the space of piecewise continuous linear boundary element basis functions $S^{1,0}(\Gamma_k^h, \boldsymbol{\chi})$. The unknown nodal dependent field quantities are then understood to be interpolated isoparametrically over the number of global nodes. In constructing this interpolation, it is also useful to introduce a global nodal indexing set for Γ_k^h defined as,

$$\mathcal{I}_k \equiv \{\chi_\tau(\hat{\mathbf{p}}_{(i,j)}) : \forall \tau \in \Gamma_k^h, \forall (i,j) \in I_p^\tau\} \quad , \quad (5.61)$$

under which the unknown double layer potential and membrane fluid velocity read,

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x}) &= \sum_{j \in \mathcal{I}_{sq}} \boldsymbol{\phi}_j \psi_j(\mathbf{x}) \\ \mathbf{u}^c(\mathbf{x}) &= \sum_{j \in \mathcal{I}_c} \mathbf{u}_j^c \psi_j(\mathbf{x}) \end{aligned} \quad . \quad (5.62)$$

It is common place to refer to an unknown field quantity's associated basis functions as *trial* functions.

In boundary element methods, there is little restriction on the choice of element type so long as the boundary can be discretized with well shape-conditioned elements. Though, one may choose a sub-parametric interpolation scheme for unknown field quantities (meaning the geometry is less smoothly resolved), the isoparametric concept is more efficient since, the unknown nodal dependent variables lie at the same locations as the nodal coordinates. We remark that the use of flat triangles may be cast as a high-level implementation issue under modern object-oriented design principles. There are no computer code restrictions since the assembly of BEM linear systems is always formulated as an element by element calculation and may be generically implemented as a function of the number of element nodes.

5.5.2 Galerkin BEM

In constructing the Galerkin formulation of a BIE, one multiplies the entire equation by a global weighting or test function and integrates over the BIE's domain. For a continuous C^0 interpolation, such test functions are defined on triangle patch domains under a globally labeled node's vertex-face 1-ring. For the i 'th globally labeled node, and its vertex-face 1-ring $v_{i,k}^* \equiv \{\tau : \tau \in \Gamma_k^h, \mathbf{p}_i \in \bar{\tau}\}$, this domain may be written as,

$$\text{supp}(\psi_{i,k}) = \{\tau \in \Gamma_k^h : \tau \in v_i^*\} \quad . \quad (5.63)$$

The test functions are chosen from the same set of affine 3-node triangular shape functions, though defined globally though the notion of local element interpolants,

$$\psi_i(\mathbf{x}) \Big|_{\tau_k} = (\psi_{j,\tau_k} \circ \chi_{\tau_k}^{-1})(\mathbf{x}), \quad \tau_k \in \text{supp}(\psi_i), \quad \psi_{j,\tau_k}(\chi_{\tau_k}^{-1}(\mathbf{p}_i)) = 1 \quad (5.64)$$

i.e. under the identification that τ_k is in the support of the global basis function and that the locally defined shape function and its node identifies precisely with the globally labeled node. This construction yields the so-called *Bubnov-Galerkin* or *Galerkin* discretization.

Multiplying Eqn. 5.43 and 5.45 by a test function $\psi_i(\mathbf{x})$, integrating over the BIE's relevant domain, and using linearity yields,

$$\begin{aligned} \langle \psi_i, \mathbf{u}^{sl} \rangle_{\Gamma_{sq}} - \langle \psi_i, \bar{\mathbf{u}}^{sq} \rangle_{\Gamma_{sq}} &= \langle \psi_i, \boldsymbol{\phi} \rangle_{\Gamma_{sq}} + \langle \psi_i, (\mathcal{K}\boldsymbol{\phi})_{\Gamma_{sq}} \rangle_{\Gamma_{sq}} \\ &+ \sum_{j=1}^6 \langle \psi_i, \boldsymbol{\varphi}^{j,RBM} \rangle_{\Gamma_{sq}} \langle \boldsymbol{\varphi}^{j,RBM}, \boldsymbol{\varphi} \rangle_{\Gamma_{sq}} \quad . \end{aligned} \quad (5.65)$$

$$\langle \psi_i, \mathbf{u}^c \rangle_{\Gamma_c} - \langle \psi_i, \bar{\mathbf{u}}^c \rangle_{\Gamma_c} = -\langle \psi_i, (\mathcal{G}\Delta\mathbf{f})_{\Gamma_c} \rangle_{\Gamma_c} \quad (5.66)$$

Several simplifying notations have been used here in order to obtain cleaner expressions. The form $\langle f, g \rangle_{\Gamma} \equiv \int_{\Gamma} f(\mathbf{x})g(\mathbf{x})dS(\mathbf{x})$ defines the inner product of two functions over the surface Γ . Operator notation $(\mathcal{K}\boldsymbol{\phi})_{\Gamma} \equiv \int_{\Gamma} K(\mathbf{x}, \mathbf{y})\boldsymbol{\phi}(\mathbf{y})dS(\mathbf{y})$ is used for its compact notational representation. Operator forms $\langle \cdot, \cdot \rangle_{\Gamma_{\text{test}}\Gamma_{\text{trial}}}$ are to be read as an inner product and imply a nested double integral first over the *test* function space defined on Γ_{test} and the *trial* function space defined on Γ_{trial} .

The appropriate number of equations is obtained by passing each $\psi_i \forall i \in \mathcal{I}_k$ through Eqn. 5.65 and 5.66. This procedure yields two dependent rectangular linear systems, which when combined yield a square linear system of dimension $3(|\mathcal{I}_{sq}| + |\mathcal{I}_c|) \times 3(|\mathcal{I}_{sq}| + |\mathcal{I}_c|)$.

5.5.3 Matrix Representation of Galerkin BEM Integrals

Subject to the mesh decomposition and the chosen finite dimensional boundary element subspace, a third approximation is made, adding to discretization error, by breaking the continuous integral up into a sum of integrals over individual elements,

$$\int_{\Gamma_k} (\cdots) dS(\bar{\mathbf{y}}) \approx \sum_{m=1}^M \int_{\tau_m} (\cdots) dS(\mathbf{y}), \quad \forall \bar{\mathbf{y}} \in \Gamma_k \text{ and } \mathbf{y} \in \Gamma_{k,M}^h \quad . \quad (5.67)$$

Inspection of Eqns. 5.65 and 5.66 reveals that there are several different general classes of computations that must be computed. Terms involving the inner product with a known function are only single integrals. If one applies isoparametric interpolation to this known function, the evaluation of this integral is extremely efficient and fast. Since each basis function has finite support given by Eqn. 5.63, under this interpolation, this inner product is simply the mass matrix times the nodal values of the known function. For example,

$$\begin{aligned} \langle \psi_i, \mathbf{u}^{sl} \rangle_{\Gamma_{sq}} &= \int_{\text{supp}(\psi_i)} \psi_i(\mathbf{x}) \sum_{j \in \mathcal{I}_{sq}} \mathbf{u}_j^{sl} \psi_j(\mathbf{x}) dS(\mathbf{x}) \\ &= \sum_{j \in \mathcal{I}_{sq}} \mathbf{u}_j^{sl} \int_{\text{supp}(\psi_i)} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) dS(\mathbf{x}), \end{aligned} \quad . \quad (5.68)$$

Passing each $\psi_i \forall i \in \mathcal{I}_{sq}$ through Eqn. 5.68 implies that the overall matrix representation for $\langle \psi_i, \mathbf{u}^{sl} \rangle$ may be written as $\mathbb{M}_{\Gamma_{sq}}(\mathbf{u}^{sl})_{\Gamma_{sq}} \equiv (\mathbb{M}\mathbf{u}^{sl})_{\Gamma_{sq}}$ where the general form of the mass matrix elements, $\mathbb{M}_{\Gamma_k}[i, j]$ may be written as,

$$\mathbb{M}_{\Gamma_k}[i, j] = \int_{\text{supp}(\psi_i) \cap \text{supp}(\psi_j)} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) dS(\mathbf{x}) \quad . \quad (5.69)$$

The matrix representation of $\langle \psi_i, \phi \rangle_{\Gamma_{sq}}$ is identically $(\mathbb{M}\phi)_{\Gamma_{sq}}$. Computations of inner products using the mass matrix are extremely efficient and fast because \mathbb{M}_{Γ_k} is always a sparse matrix. On the other hand, terms involving inner products between a basis function and BEM operator imply double integrals over two possibly identical or dissimilar meshes, respectively giving rise to the self-interaction and pairwise dense interaction matrices. Extreme care must be taken when evaluating the self-interactions since BEM operators are always singular for $\mathbf{x} \rightarrow \mathbf{y}$ or $\mathbf{y} \rightarrow \mathbf{x}$ and singular integrations dominate the matrix diagonals (and thus the conditioning of the linear system and the overall accuracy of the solution). For example, the self-interaction $\langle \psi_i, (\mathcal{K}\phi)_{\Gamma_{sq}} \rangle_{\Gamma_{sq}}$,

computation is given more explicitly by,

$$\begin{aligned} \langle \psi_i, (\mathcal{K}\phi)_{\Gamma_{sq}} \rangle_{\Gamma_{sq}} &= \int_{\text{supp}(\psi_i)} \psi_i(\mathbf{x}) \int_{\text{supp}(\psi_j)} K(\mathbf{x}, \mathbf{y}) \left(\sum_{j \in \mathcal{I}_{sq}} \phi_j \psi_j(\mathbf{y}) \right) dS(\mathbf{y}) dS(\mathbf{x}) \\ &= \sum_{j \in \mathcal{I}_{sq}} \phi_j \int_{\text{supp}(\psi_i)} \int_{\text{supp}(\psi_j)} K(\mathbf{x}, \mathbf{y}) \psi_i(\mathbf{x}) \psi_j(\mathbf{y}) dS(\mathbf{y}) dS(\mathbf{x}) \end{aligned} \quad (5.70)$$

Equation 5.70 generates the dense matrix representation $\mathcal{K}_{\Gamma_{sq}\Gamma_{sq}}(\phi)_{\Gamma_{sq}}$,

$$\mathcal{K}_{\Gamma_{sq}\Gamma_{sq}}[i, j] = \int_{\text{supp}(\psi_i)} \int_{\text{supp}(\psi_j)} K(\mathbf{x}, \mathbf{y}) \psi_i(\mathbf{x}) \psi_j(\mathbf{y}) dS(\mathbf{y}) dS(\mathbf{x}) \quad , \quad (5.71)$$

where we have used the short hand $\mathcal{K}_{\Gamma_{test}\Gamma_{trial}}$ where *test* and *trial* are the geometries that respectively define the domains of test and trial basis functions.

The discretization and numerical computation of boundary element operators, in general, produces dense matrix representations since the associate singularity solutions always have long range decay, i.e. proportional to $1/r^\alpha$, $\alpha \geq 1$. Because of this singularity, the accurate numerical computation of Eqn. 5.71 is a relatively complex problem that is far from having a trivial solution. In Galerkin methods, when the supports of the basis functions intersect, three classes of singularities occur: coincident, edge adjacent, and vertex adjacent. The accurate computation of each singularity demands different numerical integration routines and special numerical code logic. Each singularity case has been studied in great detail by a variety of previous authors. In this thesis we use the Sauter-Schwab integration rules, [13], [58], since they are black-box integration procedures that produce high accuracy at relatively low computational cost.

The integration and matrix representation of $\langle \psi_i, (\mathcal{G}\Delta \mathbf{f})_{\Gamma_c} \rangle_{\Gamma_c}$ follows identically to Eqn. 5.70 and 5.71 although the singularity of \mathcal{G} is one order less than \mathcal{K} . Since the unknowns $\mathbf{U}^m, \mathbf{\Omega}^m$ are constant vectors, they interpolate as constants. One possible way to account for this is to first define a function (for notational convenience),

$$R^{TBN}(\mathbf{y}; R_t, R_b, R_n) = R_n \mathbf{n}(\mathbf{y}) \mathbf{n}(\mathbf{y}) + R_b \mathbf{b}(\mathbf{y}) \mathbf{b}(\mathbf{y}) + R_t \mathbf{t}(\mathbf{y}) \mathbf{t}(\mathbf{y}) \quad (5.72)$$

and rewrite the term $(\mathcal{G}\Delta\mathbf{f})_{\Gamma_c}$ as a sum of three terms, namely,

$$\begin{aligned}
(\mathcal{G}\Delta\mathbf{f})_{\Gamma_c} &= \int_{\Gamma_c^h} G(\mathbf{x}, \mathbf{y}) \left[\{\mathbf{u}^c(\mathbf{y}) - [\mathbf{U}^m + \boldsymbol{\Omega}^m \times (\mathbf{y} - \mathbf{x}_c^m)]\} \cdot R^{TBN}(\mathbf{y}) \right] dS(\mathbf{y}) \\
&\stackrel{\text{indx}}{=} \sum_{j \in \mathcal{I}_c} u_{j:m}^c \int_{\text{supp}(\psi_j)} G_{kl}(\mathbf{x}, \mathbf{y}) R_{lm}^{TBN}(\mathbf{y}) \psi_j(\mathbf{y}) dS(\mathbf{y}) \\
&\quad - U_m^m \int_{\Gamma_c^h} G_{kl}(\mathbf{x}, \mathbf{y}) R_{lm}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&\quad - \Omega_m^m \int_{\Gamma_c^h} G_{kl}(\mathbf{x}, \mathbf{y}) R_{ln}^{TBN}(\mathbf{y}) \epsilon_{nmp}(\mathbf{y} - \mathbf{x}_c^m)_p dS(\mathbf{y}) \\
&= (\mathcal{G}^{TBNu} \mathbf{u}^c)_{\Gamma_c} - (\mathcal{G}^{TBNU} \mathbf{U}^m)_{\Gamma_c} - (\mathcal{G}^{TBN\Omega} \boldsymbol{\Omega}^m)_{\Gamma_c}
\end{aligned} \tag{5.73}$$

The rigid body motion terms, $\langle \psi_i, \boldsymbol{\varphi}^{j,RBM} \rangle_{\Gamma_{sq}} \langle \boldsymbol{\varphi}^{j,RBM}, \boldsymbol{\varphi} \rangle_{\Gamma_{sq}}$, follow along the same lines as Eqn. 5.68 and are constructed as,

$$\langle \psi_i, \boldsymbol{\varphi}^{j,RBM} \rangle_{\Gamma_{sq}} \langle \boldsymbol{\varphi}^{j,RBM}, \boldsymbol{\varphi} \rangle_{\Gamma_{sq}} \rightarrow (\mathbb{M} \boldsymbol{\varphi}^{j,RBM})_{\Gamma_{sq}} (\boldsymbol{\varphi}^{j,RBM} \cdot \mathbb{M})_{\Gamma_{sq}} \tag{5.74}$$

Pairwise interactions must also be computed. They are present in the *disturbance* flow inner products $\langle \psi_i, \bar{\mathbf{u}}^{sq} \rangle_{\Gamma_{sq}}$ and $\langle \psi_i, \bar{\mathbf{u}}^c \rangle_{\Gamma_c}$. The structure of the integrations is similar to the self-interaction integrations, except the double integrals are now computed over patch product domains on dissimilar meshes, e.g. $(\tau_i, \tau_j) \in \Gamma_{sq}^h \times \Gamma_c^h$. If the surfaces are well-separated, the computation is straightforward because the singularity in the integral's kernel becomes relatively smooth. However, when the surfaces are close, care must also be taken in appropriately handling these regular integrations adaptively. It is commonplace to refer to these close surface-surface integrations as *nearly singular integrations* which are roughly defined by $\mathbf{x} \approx \mathbf{y}$ and imply that the integral's kernel $\propto 1/r^\alpha$ diverges.

Discretization of the force and torque constraints as single integrals over Γ_c only requires expansion in the trial space, in this case the boundary element subspace built

on Γ_c . The force constraint may be expanded as,

$$\begin{aligned}
\int_{\Gamma_c} \Delta \mathbf{f}(\mathbf{y}) dS(\mathbf{y}) &= \int_{\Gamma_c} \{ \mathbf{u}^c(\mathbf{y}) - [\mathbf{U}^m + \boldsymbol{\Omega}^m \times (\mathbf{y} - \mathbf{x}_c^m)] \} \\
&\quad \cdot R^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&\stackrel{\text{indx}}{=} \sum_{j \in \mathcal{I}_c} u_{j:k}^c \int_{\text{supp}(\psi_j)} R_{kp}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&\quad - U_k^m \int_{\Gamma_c} R_{kp}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&\quad - \Omega_l^m \int_{\Gamma_c} \epsilon_{klm}(\mathbf{y} - \mathbf{x}_c^c)_m R_{kp}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&= (R^{TBNu} \mathbf{u}^c)_{\Gamma_c} - (R^{TBNU} \mathbf{U}^m)_{\Gamma_c} - (R^{TBN\Omega} \boldsymbol{\Omega}^m)_{\Gamma_c}
\end{aligned} \tag{5.75}$$

A completely analogous procedure may be used to partition the torque constraint into a sum of three quantities. This expansion proceeds as,

$$\begin{aligned}
\int_{\Gamma_c} (\mathbf{y} - \mathbf{x}_c^c) \times \Delta \mathbf{f}(\mathbf{y}) dS(\mathbf{y}) &= \int_{\Gamma_c} (\mathbf{y} - \mathbf{x}_c^c) \\
&\quad \times [\{ \mathbf{u}^c(\mathbf{y}) - [\mathbf{U}^m + \boldsymbol{\Omega}^m \times (\mathbf{y} - \mathbf{x}_c^m)] \} \\
&\quad \cdot R^{TBN}(\mathbf{y})] dS(\mathbf{y}) \\
&\stackrel{\text{indx}}{=} \sum_{j \in \mathcal{I}_c} u_{j:k}^c \int_{\text{supp}(\psi_j)} \epsilon_{qrp}(\mathbf{y} - \mathbf{x}_c^c)_r R_{kp}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&\quad - U_k^m \int_{\Gamma_c} \epsilon_{qrp}(\mathbf{y} - \mathbf{x}_c^c)_r R_{kp}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&\quad - \Omega_l^m \int_{\Gamma_c} \epsilon_{qrp}(\mathbf{y} - \mathbf{x}_c^c)_r \epsilon_{klm}(\mathbf{y} - \mathbf{x}_c^c)_m R_{kp}^{TBN}(\mathbf{y}) dS(\mathbf{y}) \\
&= (R^{TBNr \times u} \mathbf{u}^c)_{\Gamma_c} - (R^{TBNr \times U} \mathbf{U}^m)_{\Gamma_c} - (R^{TBNr \times \Omega} \boldsymbol{\Omega}^m)_{\Gamma_c}
\end{aligned} \tag{5.76}$$

5.5.4 Global Linear System

The global linear system may be assembled by first constructing the matrix representation of each inner product, factoring out unknowns, and then inserting each inner product matrix block into the global BEM stiffness matrix. Since the Stokes equations are vector equations and the singularity solutions are second order tensors, the matrix dimensions follow by multiplying the number of nodes by three. Let the number of nodes in geometry Γ_k^h be defined by the cardinality of the index set \mathcal{I}_k . For the squirmer porous container geometry the number of nodes for each geometry is defined as $N_{sq} = 3|\mathcal{I}_{sq}|$ and $N_c = 3|\mathcal{I}_c|$. The squirmer porous container problem then

takes the form,

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \dim(A) &= (N_{sq} + N_c + 6) \times (N_{sq} + N_c + 6) \quad , \\ \dim(x) &= \dim(b) = (N_{sq} + N_c + 6) \times (1) \end{aligned} \quad (5.77)$$

where A is the BEM stiffness matrix, \mathbf{x} is the vector of unknowns and \mathbf{b} is the load vector. Expanding the notation, Eqn. 5.77 is equivalent to,

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} (\boldsymbol{\phi})_{\Gamma_{sq}} \\ (\mathbf{u}^c)_{\Gamma_c} \\ \mathbf{U}^m \\ \boldsymbol{\Omega}^m \end{bmatrix} = \begin{bmatrix} (\mathbf{u}^{sl})_{\Gamma_{sq}} \\ (\mathbf{0})_{\Gamma_c} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad , \quad (5.78)$$

where the self-interaction (diagonal) terms are given by,

$$A_{11} = \mathbb{M}_{\Gamma_{sq}} + \mathcal{K}_{\Gamma_{sq}, \Gamma_{sq}} + \sum_{j=1}^6 (\mathbb{M}\boldsymbol{\varphi}^{j,RBM})_{\Gamma_{sq}} (\boldsymbol{\varphi}^{j,RBM} \cdot \mathbb{M})_{\Gamma_{sq}} \quad (5.79)$$

$$A_{22} = -\mathcal{G}_{\Gamma_c \Gamma_c}^{TBNU} - \mathbb{M}_{\Gamma_c} \quad (5.80)$$

$$A_{33} = -R_{\Gamma_c}^{TBNU} \quad (5.81)$$

$$A_{44} = -R_{\Gamma_c}^{TBNUr \times \Omega} \quad . \quad (5.82)$$

The pairwise interactions (off-diagonal) terms are given by,

$$A_{12} = -\mathcal{G}_{\Gamma_{sq} \Gamma_c}^{TBNU}, \quad A_{13} = \mathcal{G}_{\Gamma_{sq}}^{TBNU}, \quad A_{14} = \mathcal{G}_{\Gamma_{sq}}^{TBNU\Omega} \quad (5.83)$$

$$A_{21} = \mathcal{K}_{\Gamma_c \Gamma_{sq}}, \quad A_{23} = \mathcal{G}_{\Gamma_c}^{TBNU}, \quad A_{24} = \mathcal{G}_{\Gamma_c}^{TBNU\Omega} \quad . \quad (5.84)$$

The final two rows of the stiffness matrix are given by discretizing the force and torque integral constraints, Eqns. 5.51 and 5.50.

$$A_{31} = [\mathbf{0}]_{3 \times N_{sq}}, \quad A_{32} = R_{\Gamma_c}^{TBNU}, \quad A_{34} = -R_{\Gamma_c}^{TBNU\Omega} \quad (5.85)$$

$$A_{41} = [\mathbf{0}]_{3 \times N_{sq}}, \quad A_{32} = R_{\Gamma_c}^{TBNUr \times u}, \quad A_{34} = -R_{\Gamma_c}^{TBNUr \times U} \quad . \quad (5.86)$$

5.6 Numerical Solution and Comparison with the Analytical Model

We first show agreement between the analytical model and the GBEM numerics when the container's and squirmer's center of mass are coincident. This can be accomplished for a variety of confinement ratios $b/a > 1$, permeability parameter sets, and squirmer types (given by the ratio of $\beta = B_2/B_1$). However, axisymmetry constraints implied by the coincident squirmer porous container geometry demands that $R_t = R_b \equiv R_{\parallel}$ and $R_n \equiv R_{\perp}$ implying that the proposed BEM model is slightly more general. In the analytical model, the squirmer orientation is also fixed as $\mathbf{e}_s = \mathbf{e}_z$. The numerical solution is independent of the squirmer's orientation in the sense that one may always construct a rotation matrix \mathbf{R} that takes the $\mathbf{e}_s \rightarrow \mathbf{e}_z$ without affecting the axisymmetry of the system. Thus the numerical solution in the concentric geometry is always equivalent to the analytical model up to a rotation.

5.6.1 Parameterization

To solve the global linear system given in Eqns. 5.77 or 5.78, one must choose container permeability parameters $\{R_t, R_b, R_n\}$, squirmer and container radii $\{a, b\}$, a set of squirming parameters describing orientation and squirming modes $\{\mathbf{e}_s, B_1, B_2\}$, and initial positions for the container and squirmer centers of mass $\{\mathbf{x}_c^c, \mathbf{x}_c^{sq}\}$. So long as the number of nodes is kept relatively small, the solution of the linear system may be obtained relatively efficiently using standard LU decomposition. Situations in which $b/a \approx 1$ or where $(b - a)/a \approx \delta$ for $\delta \ll 1$ require one to use either mesh adaptivity and/or efficient nearly singular quadratures to control element size and errors in numerical integrations.

In this work the solution of the linear system is computed using the Galerkin BEM discretization down to a container size of $b/a = 1.1$ with appropriate spatial mesh adaptivity subject to the nearly singular ratio distance cutoff defined in Chapter 4. The linear system is solved over the range of $1.1 \leq b/a \leq 10$ using an advanced, fast, GPU Galerkin BEM (GPUGBEM) implementation that uses the most recent version of the CUDA API. This implementation has been created as a special general framework for solving a wide variety of multi-body fluid structure interaction problems in Stokes flow (see GPUGBEM framework in Chapter 4). These calculations are computed on a NVIDIA GTX 780 using an analogous implementation of the well-known fast-nbody simulation techniques [44]. However, the Galerkin BEM for vector equations is considerably more difficult to implement on GPUs as many issues arise

in various subtle forms: race conditions, data partitioning, and thread concurrency. Also, although not obvious, each different type of operator (self, pairwise, integral constraints, etc...) must all be implemented, optimized, and consequently computed in different ways.

The numerical solution is computed down to a container size ratio of $b/a = 1.1$ without utilizing nearly singular integration/quadrature routines. During each solve, the squirmer and container mesh are optimized using a very efficient and fast $O(N \log(N))$ mesh subdivision algorithm (see GPUGBEM in Chapter 4). Roughly speaking, this new local method for h-adaptive meshing of the container and squirmer surfaces conformally refines all element pairs $\{(\tau_x, \tau_y) \in \Gamma_c^h \times \Gamma_{sq}^h : d_{xy}(\tau_x, \tau_y) \leq 1\}$. The function $d_{xy}(\tau_x, \tau_y)$ is defined by,

$$d_{xy} = \frac{\text{dist}(\tau_x, \tau_y)}{\max(h_x, h_y)} \quad , \quad (5.87)$$

where h_x and h_y are local element diameters of elements τ_x and τ_y .

5.6.2 Concentric Geometry BEM Results

The analytical model given in Eqn. 5.23 and 5.24 predicts that both the squirmer and membrane translational velocities are independent of the type of squirmer (determined by β). This is confirmed in the BEM calculations shown in Figs. 5.5, 5.31, 5.32, 5.33, 5.34, 5.35. The BEM model produces results that are in extremely good agreement with the analytic model. Specifically, global minimums in the squirmer's translational velocity are recovered. As a general trend, it is observed that smaller resistances yield greater squirmer speeds. The recovery of the non-porous container solution using high values of permeability resistances (10^{12}) highlights the fact that the squirmer's speed is always bounded from below by the non-porous container solution. Less resistance in the normal direction also produces faster squirming speeds when compared to a container having a tangential resistance of similar magnitude.

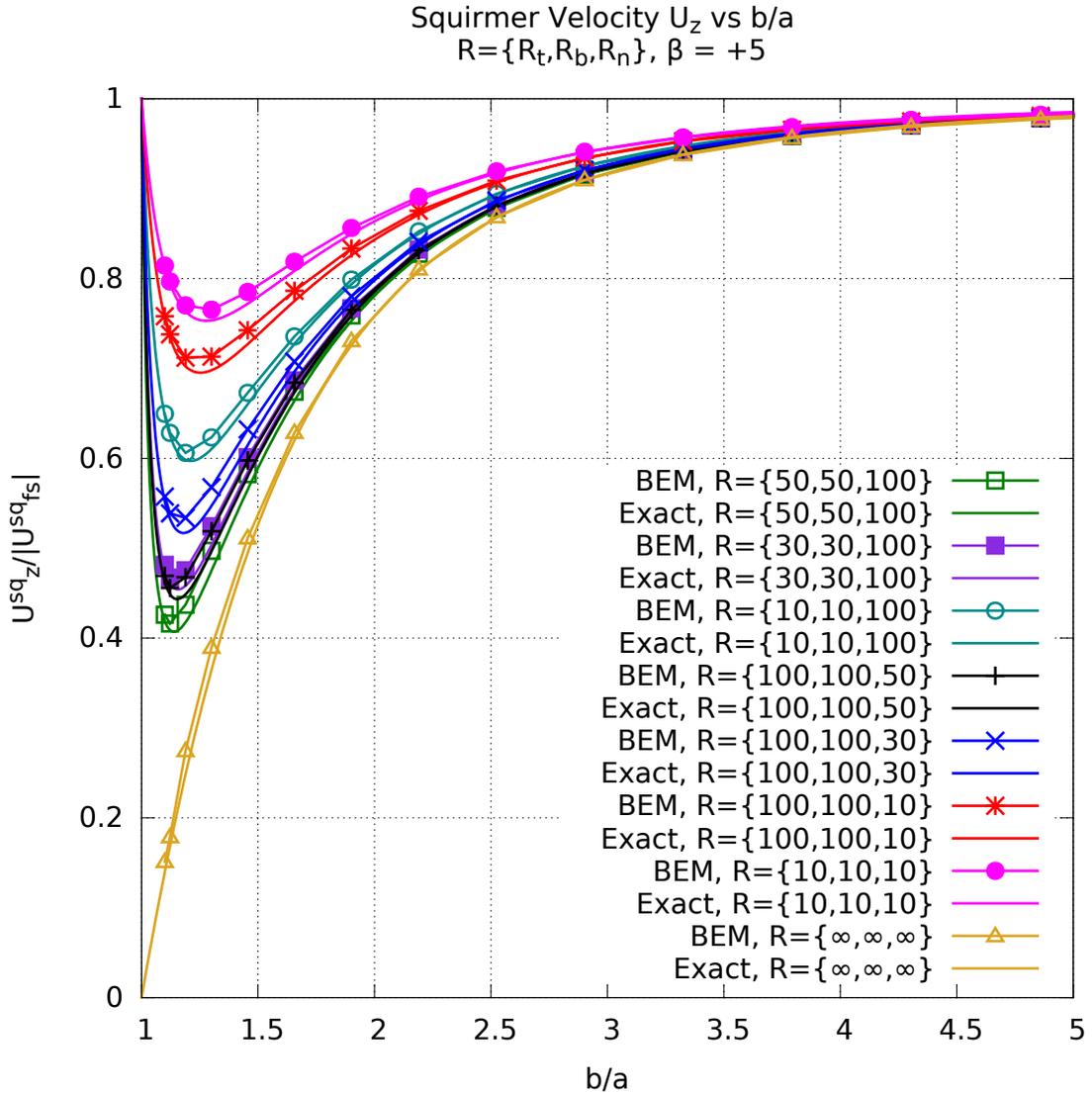


Fig. 5.5 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized squirmer's translational velocity, $U_z^{sq}/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer's translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model.

5.7 Mobility Field Calculations

So far all previous calculations have been for the completely axisymmetric geometry where the squirmer and container are perfectly concentric, i.e. $\mathbf{x}_c^c = \mathbf{x}_c^p$. Axisymmetry

in the geometry is necessary when constructing the analytical solution. Conversely, the Galerkin BEM model can be solved for any placement of the squirmer so long as $|\mathbf{x}_c^{sq} - \mathbf{x}_c^c| < b - a$ meaning that the container and squirmer may not intersect. In practice, in order to have stable and accurate numerical solutions, a further restriction on the center to center separation needs to be enforced, namely,

$$r_{cp} = |\mathbf{x}_c^{sq} - \mathbf{x}_c^c| < b - a(1 + \delta) \equiv r_{cp}^{max} \quad . \quad (5.88)$$

In this way, the surface of squirmer is always kept a distance $a\delta$ away from the container walls. The parameter δ functions as a fractional squirmer to container gap-size.

It is important to make δ small enough so that the mobility solution is sampled in the squirmer-container lubrication region. Very few authors ever attempt to simulate particles in Stokes flow down to these separations. If this is attempted, lubrication analytics are usually incorporated after the far-field has been solved. This is an important distinction since the calculations shown here, which are computed using the author's very efficient GPUGBEM framework, allow full numerical resolution of these solutions down to very small particle gap sizes. Using a novel locally h-adaptive re-meshing algorithm, mobility solutions for small δ may be accurately sampled with little computational cost. The calculations in this section show that reasonable results can be obtained down to $\delta = 0.05$ and likely for smaller values of δ . In all of the following trajectory calculations, computations are performed down to $\delta = 0.05$. Owing to the fact that it is physically impossible for a squirmer to pass through the container surface, squirmers attempting to cross the container boundary simply remain fixed at a center to center separation of r_{cp}^{max} , yet may continue to keep running into the container boundary at the $\delta = 0.05$ level mobility solution.

5.7.1 Basic Trajectory Calculations and Improvements

Subject to an appropriate system specification in parameters \mathbf{P} ,

$$\mathbf{P} = \langle a, b, R_t, R_b, R_n, B_1, B_2, \mathbf{e}_s, \mathbf{x}_c^{sq}, \mathbf{x}_c^c \rangle \quad (5.89)$$

the most obvious way to do a trajectory analysis is to

1. Adaptively mesh both container and squirmer surfaces
2. Solve the mobility problem, Eqn. 5.78, for rigid body motions (RBMs) $\mathbf{U}^{sq}, \mathbf{\Omega}^{sq}$ and $\mathbf{U}^m, \mathbf{\Omega}^m$

3. Integrate the center of mass and orientations of the squirmer and container using the methods of Chapter 2
4. Construct an overall affine transform, (\mathbf{RST}) , describing the overall translation and rotation of the particle
5. Check for container/squirmer collisions
6. Evolve squirmer and container meshes according to (\mathbf{RST}) and appropriate collision conditions
7. Evolve the squirmer orientation according to the rotational part of (\mathbf{RST})
8. Evolve time and repeat

In a CPU only implementation, this procedure may be reasonable for computing a single squirmer trajectory so long as the squirmer does not stay close to the container surface where, solutions of the mobility problem tend to become prohibitively slow for each time step. This computational speed burden may be alleviated using the GPU implementation under the GPUGBEM framework. However, what if one wants to compute hundreds or even thousands of trajectories, sampling a dense 3D grid of squirmer starting positions $\mathbf{x}_c^{sq}(t=0)$, and do so in real time? Even if each trajectory takes milliseconds to fully calculate, simulation of many trajectories is still too slow to be evaluated in real time. The answer is to implement trajectory interpolation over a special set of *standard* mobility solutions.

5.7.2 Interpolation Based Mobility Solutions and Trajectory Sampling

In this section we devise an algorithm that allows for fast squirmer trajectory sampling given a large set of arbitrary starting squirmer parameterizations $\{\mathbf{x}_c^{sq}, \mathbf{e}_s\}$. Since the Stokes equations are time independent and velocity fields are assumed to propagate instantaneously, each spatial location for the squirmer, \mathbf{x}_c^p , relative to the container's center of mass, \mathbf{x}_c^c , produces a mobility solution that is independent of time. Therefore, a squirmer with a given position and orientation produces the same mobility solution regardless of its past trajectory history. This means that one may sample a set of squirmer positions, find the mobility solution, and then use this set of solutions to reconstruct the trajectory.

However, it would seem that for each set of squirmer positions, one would need to sample an entire dense range of orientations in order to reconstruct even a single

arbitrary trajectory (meaning the squirmer may have been initialized with an arbitrary orientation). This is not the case. The squirmer and container geometry is special in the sense that both surfaces are spherical, and so there always exists a symmetry plane that perfectly divides both container and squirmer surfaces, Γ_c and Γ_{sq} . Moreover the parameterizations of the squirmer and container have either an attached explicit orientation, \mathbf{e}_s , or an orientation that is implied when computing the rigid body dynamics. Due to the problems overall spherical geometry, there always exists a rotation from a fixed or standard orientation vector, $\mathbf{e}_s^0 = \mathbf{e}_Z$, to an arbitrary squirmer orientation \mathbf{e}_s . In other words one need only sample mobility solutions from a single orientation, preferably at the same standardized vector for each sampled squirmer position, and then compute the inverse rotation of this standard mobility solution. It also may appear that the mobility solution needs to be sampled on a dense grid in 3D in order to incorporate all squirmer center of mass positions. Again, this is not the case. Owing to the existence of this symmetry plane, a second rotation may always be constructed mapping an arbitrary position vector to the xz plane. Put differently, an arbitrary position vector can always be made to lie in the xz plane by rotating a coordinate system fixed to the center of the container.

In summary, the mobility problem need only be sampled at positions on a 2D grid in the xz plane for a fixed orientation \mathbf{e}_s^0 . To recover the actual mobility solution for a squirmer at an arbitrary position and orientation, one need only rotate the standard mobility solution back by the inverse of the rotation that was used to map $\mathbf{e}_s \rightarrow \mathbf{e}_Z$. To proceed with this algorithm, it is useful to define the standard parameterization of the container and squirmer as,

$$\mathbf{P}^0(x, z) = \langle a, b, R_t, R_b, R_n, B_1, B_2, \mathbf{e}_s = \mathbf{e}_Z, \mathbf{x}_c^{sq} = (x, 0, z), \mathbf{x}_c^c = (0, 0, 0) \rangle \quad (5.90)$$

from which a set of standard mobility solutions may be generated for various squirmer starting positions (x, z) .

5.7.3 Mobility Solution Sampling Grid

The mobility solution is sampled on a circularly masked regular Cartesian grid, in effect producing a discrete 2D sampling of mobility solutions for set of squirmer positions (initialized centers of mass). The curved circular boundary represented by $r = r_{cp}^{max}$ makes obtaining a smooth regular sampling of points near this boundary problematic. This is remedied by using linear extrapolation for $r > r_{cp}^{max}$.

Examples of grids used to sample the mobility solution are shown for $b/a \in \{3, 5, 10\}$ in Figs. 5.6, 5.36, and 5.37. The region of space defined by $b - a > r > r_{cp}^{max}$ is well

inside the Stokes lubrication region, where solutions tend to vary nonlinearly. However, the use of a linear extrapolation in this region is a reasonable first approximation since the lubrication mobility solution has been sampled at r_{cp}^{max} (inside the lubrication region) and must at least vary linearly in its Taylor expansion for a small region near r_{cp}^{max} given by $r_{cp}^{max} + \epsilon, \epsilon \ll a\delta$.

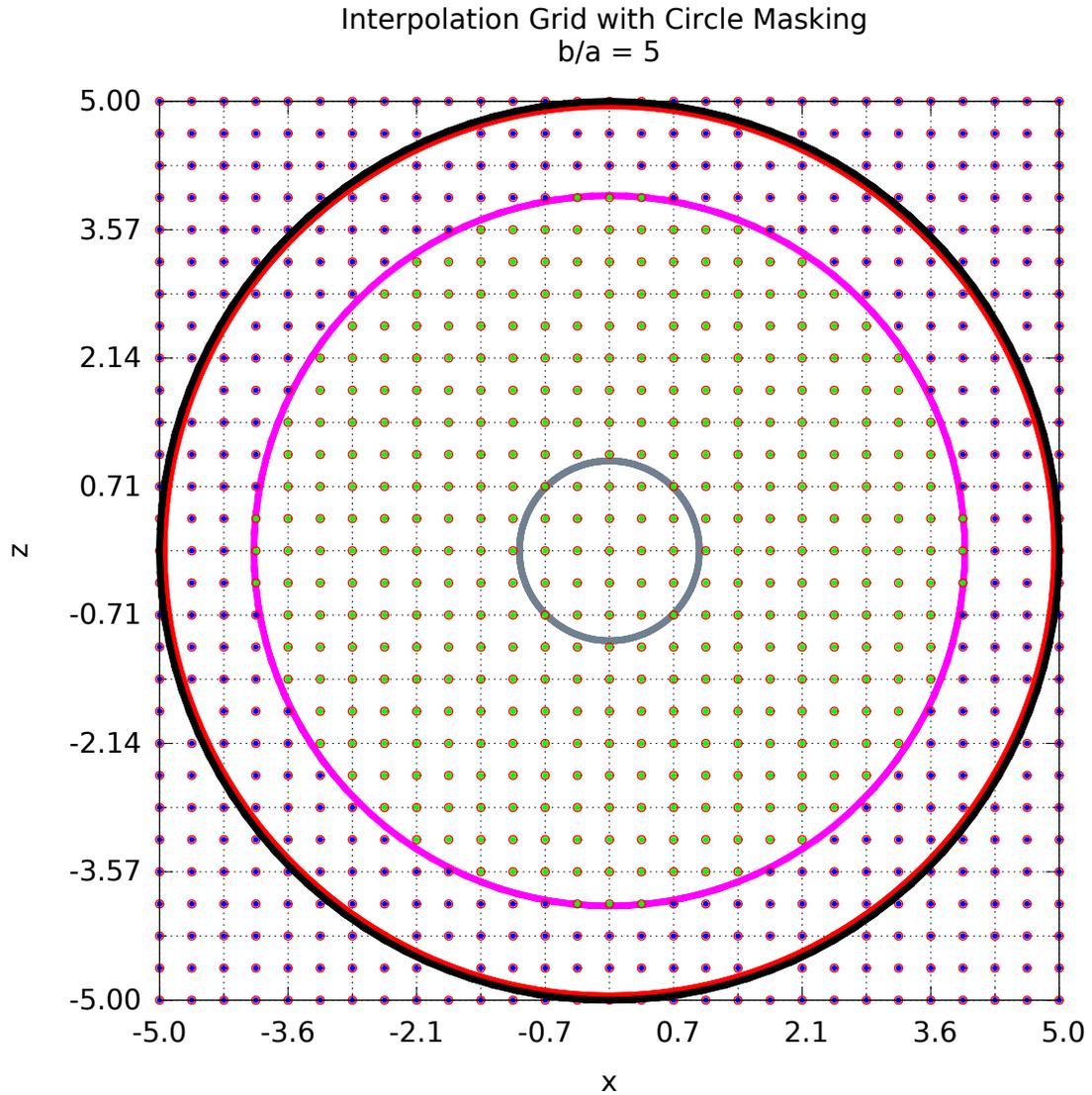


Fig. 5.6 In this plot all regular Cartesian (query) grid points are circled in red. Grid points that are inside (outside) of the region defined by $r < r_{cp}^{max}$ are shown in green (blue). The mobility solution is solved for a squirmer placed at all green points (841 query points and 385 mobility solves). A thick black circle shows the container boundary at $r = b$. An inner red circle shows where $r = b - a\delta$. The pink or fuchsia colored circle shows where $r = r_{cp}^{max}$. The inner gray circle is sized according to $r = a$ giving a sense of the squirmer's size.

In this work we eventually use bicubic interpolation which explicitly requires the use of a regular Cartesian grid that covers all of the circular masking region.

5.7.4 Mobility Solution Fields: Up-sampling to a Finer Grid

The grids with circular maskings for $b/a \in \{3, 5, 10\}$ are likely dense enough for computing squirmer and container trajectories using interpolation. However, all of these grids may be smoothly up-sampled to produce densely packed mobility solutions. We introduce the following notation for a 2D grid, \mathbf{G}^n ,

$$\begin{aligned} \mathbf{G}^n &= \mathcal{X} \times \mathcal{Z} \\ \mathcal{X} &= [-b, \dots, -b + (i-1)\delta_x, -b + i\delta_x = +b], \quad i \in \{0, \dots, n+1\} \\ \mathcal{Z} &= [-b, \dots, -b + (j-1)\delta_z, -b + j\delta_z = +b], \quad j \in \{0, \dots, n+1\} \\ n &= 2b \left\lceil \frac{2b}{(b - r_{cp}^{max})/3} \right\rceil \\ \delta_x = \delta_y &= \frac{2b}{n} \end{aligned} \quad , \quad (5.91)$$

where the Cartesian product is taken over discrete sets \mathcal{X} and \mathcal{Z} . The j 'th row of \mathbf{G}^n is at $z = -b + j\delta_z$ and the i 'th column is at $x = -b + i\delta_x$. Here δ_x and δ_z define grid spacings in x and z . The integer n defines the number of bins along a grid abscissa and the element $\mathbf{G}^n(i, j) = (x, z)_{ij}$ represents a point $(x, z) \in \mathcal{X} \times \mathcal{Z}$. The grid parameter n is constructed to ensure that there are at least 3 grid points in the region $b > r > r_{cp}^{max}$. This creates a reasonable Cartesian based approximation of the circular boundary at $r = r_{cp}^{max}$. Up-sampling proceeds by introducing an interpolation scheme taking $\mathbf{G}^n \rightarrow \mathbf{G}^{\kappa n}$ for an integer $\kappa > 1$. All results have been obtained by interpolating on an up-sampled grid with $\kappa = 3$. Examples of up-sampled grids for $b/a \in \{3, 5\}$ are shown in Figs. 5.38 and 5.39.

Bicubic interpolation is used to calculate the mobility solution on the up-sampled grid. These interpolated mobility solutions may be interpreted as mobility *flow fields*, where each solution represents an element of the set,

$$\{ \langle \mathbf{U}^{sq}, \mathbf{\Omega}^{sq}, \mathbf{U}^m, \mathbf{\Omega}^m \rangle_{ij}^0 : \mathbf{P}((x, z)_{ij}) = \mathbf{P}^0((x, z)_{ij}), \quad \forall (x, z)_{i,j} \in \mathcal{X} \times \mathcal{Z} \} \quad . \quad (5.92)$$

The bicubic interpolation creates a tautologically forced smooth solution. Improvements in the accuracy of the interpolation can be made by numerically approximating the derivatives of the interpolated quantities. Centered difference approximations are used to approximate first derivatives in x and y of the container's and squirmer's center of mass velocities and angular velocities.

Each standard mobility solution depends on the various parameters in \mathbf{P}^0 . The size ratio is varied over the set $b/a \in \{3, 5, 10\}$, B_1 is fixed at $+1$, B_2 is varied over the set $B_2 \in \{-5, 0, +5\}$, and the permeability resistance parameters are each varied over the discrete set $R_{tbn} = \{100, 50, 30, 10\}$ such that $R_t = R_b$. In all plots and data, the permeability resistance parameters are non-dimensionalized by the particle size a .

5.7.4.1 Container Velocity Field, U^m Scalings

Scaling of the container velocity field, U^m , as a function of squirmer positions can be understood by applying known decay characteristics of the squirmer's free space velocity field in combination with the proposed porous container model. The velocity fields of pushers and pullers, defined by $\beta \neq 0$, decay as $O(r^{-2})$ just like the Stokes point force dipole,

$$G_{ijl}^D = \frac{1}{r^3} (\delta_{ij}x_j - \delta_{il}x_j - \delta_{jl}x_i) + \frac{3x_i x_j x_l}{r^5} \quad . \quad (5.93)$$

On the other hand the free space velocity field of a neutral squirmer ($\beta = 0$) decays as $O(r^{-3})$ just like a source dipole,

$$D_{ij} = -\frac{\delta_{ij}}{r^3} + 3\frac{x_i x_j}{r^5} \quad . \quad (5.94)$$

Sometimes, $\beta = 0$ squirmers are loosely grouped together with *quadruple swimmers*, which have the same far field decay.

It is important to note that the presence of the B_2 swimming mode fundamentally changes the far field decay solution. The pressure field for the point force dipole is given by,

$$\Pi_{jl}^{GD} = -2\frac{\delta_{ij}}{r^3} + 6\frac{x_j x_l}{r^5} \quad . \quad (5.95)$$

Pressure fields associated with the point force dipole, Π^{GD} and the source dipole, Π^{SD} decay at one order higher, respectively $O(r^{-3})$ and $O(r^{-4})$.

From the representative boundary integral representation for the pressure,

$$p(\mathbf{x}) = \mu \int_{\Gamma_{sq}} \Pi_{il}^{GD}(\mathbf{x}, \mathbf{y}) n_l(\mathbf{y}) \phi_i(\mathbf{y}) dS(\mathbf{y}) \quad (5.96)$$

the correct pressure scaling at the container wall ($r = b$) are,

$$p = O\left(|\mathbf{U}_{fs}^{sq}|\mu\frac{a^2}{(b-a)^3}\right), \quad \beta \neq 0 \quad (5.97)$$

$$p = O\left(|\mathbf{U}_{fs}^{sq}|\mu\frac{a^3}{(b-a)^4}\right), \quad \beta = 0 \quad . \quad (5.98)$$

Additionally, by using the proposed porous container model we relate velocity and pressure at the container and find that container velocity must scale as,

$$|\mathbf{U}^m| = O\left(\frac{|\mathbf{U}_{fs}^{sq}|}{R_{min}}\frac{a^2}{(b-a)^3}\right), \quad \beta \neq 0 \quad (5.99)$$

$$|\mathbf{U}^m| = O\left(\frac{|\mathbf{U}_{fs}^{sq}|}{R_{min}}\frac{a^3}{(b-a)^4}\right), \quad \beta = 0 \quad . \quad (5.100)$$

Subsequent plots of the container velocity field are all scaled by either Eqn. 5.99 or 5.100 and so magnitudes should be interpreted relative to free space squirmer decay conditions.

5.7.4.2 Effect of Permeability Resistances on \mathbf{U}^m and \mathbf{U}^{sq}

Variations in the permeability resistance parameters have the most effect on the container velocity \mathbf{U}^m . If the squirmer type is held constant, cases in which the normal resistance is weak, $R_t = R_b > R_n$, and in which the tangential resistance is weak $R_t = R_b < R_n$, differ by a mirror flipping of the container velocity across both the x and z planes. This effect is shown in Figs. 5.7 and 5.40 for a size ratio of $a/b = 3$, a $\beta = +5$ squirmer, and using the parameters of, respectively, $R = \{10, 10, 100\}$, $R = \{100, 100, 10\}$. This effects exists for all tested size ratios and squirmer types and means that the motion of the porous container may be changed by modifying its intrinsic porous structure so that either normal or tangential flow is favored.

Effects on the squirmer's translational velocity are not quite as dramatic. In general, weak tangential resistance results in weak (small magnitude) swimming near the posterior and anterior part of the region defined by $r = r_{cp}^{max}$ for $\beta = +5$ and $\beta = -5$. The swimming direction is reversed in these regions very near to where $r = r_{cp}^{max}$. These weak tangential resistance effects for the same parameters can be seen in Figure 5.41. Conversely weak normal resistance does not seem to effect the overall swimming direction or magnitude much though the different squirmers still produce relatively different velocity fields. Weak normal velocity fields for \mathbf{U}^{sq} are shown in Figure 5.43.

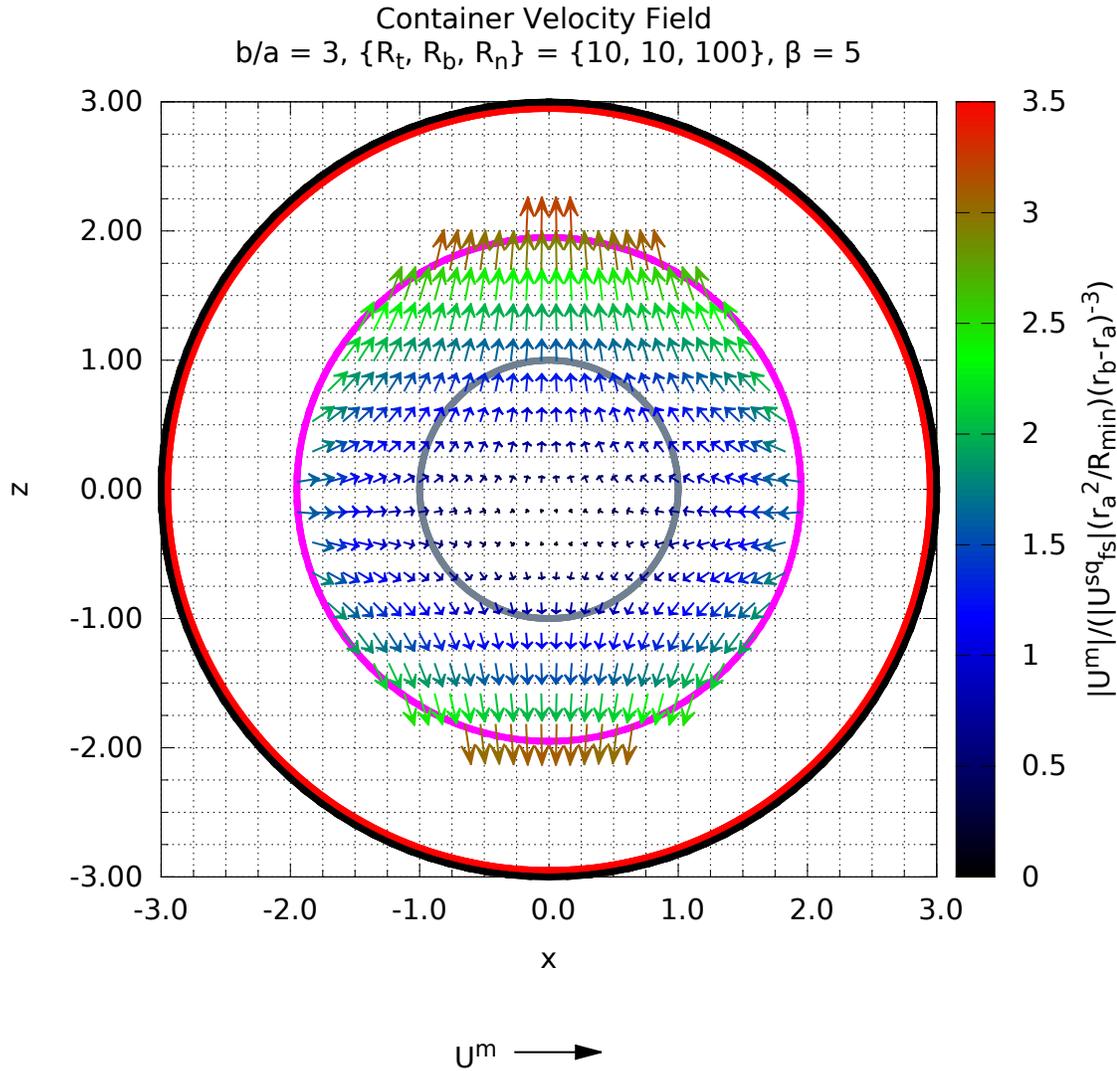


Fig. 5.7 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

5.7.4.3 Effect of Squirmer Type on U^m and U^{sq}

Holding the permeability parameters fixed, each squirmer type (pusher, puller, neutral) produces a different type of mobility velocity field. Generally speaking $\beta < 0$ pushers produce U^{sq} fields that show net motion upwards and directed outwards away from the z axis towards $r = r_{cp}^{max}$ and with mirror symmetry along the xy plane. Pullers with $\beta > 0$ also show net motion upwards, but on the contrary, their velocity is

directed inward towards the z axis and with mirror symmetry along the xy plane. These effects may be seen, for example, by comparing two weak tangential (5.41, 5.8) or weak normal (5.43, 5.44) plots for U^{sq} and $\beta = \pm 5$ squirmers. Neutral squirmers (Figs. 5.42, 5.45) show relative uniform velocity fields which indicate that they are relatively unaffected by the presence of the container.

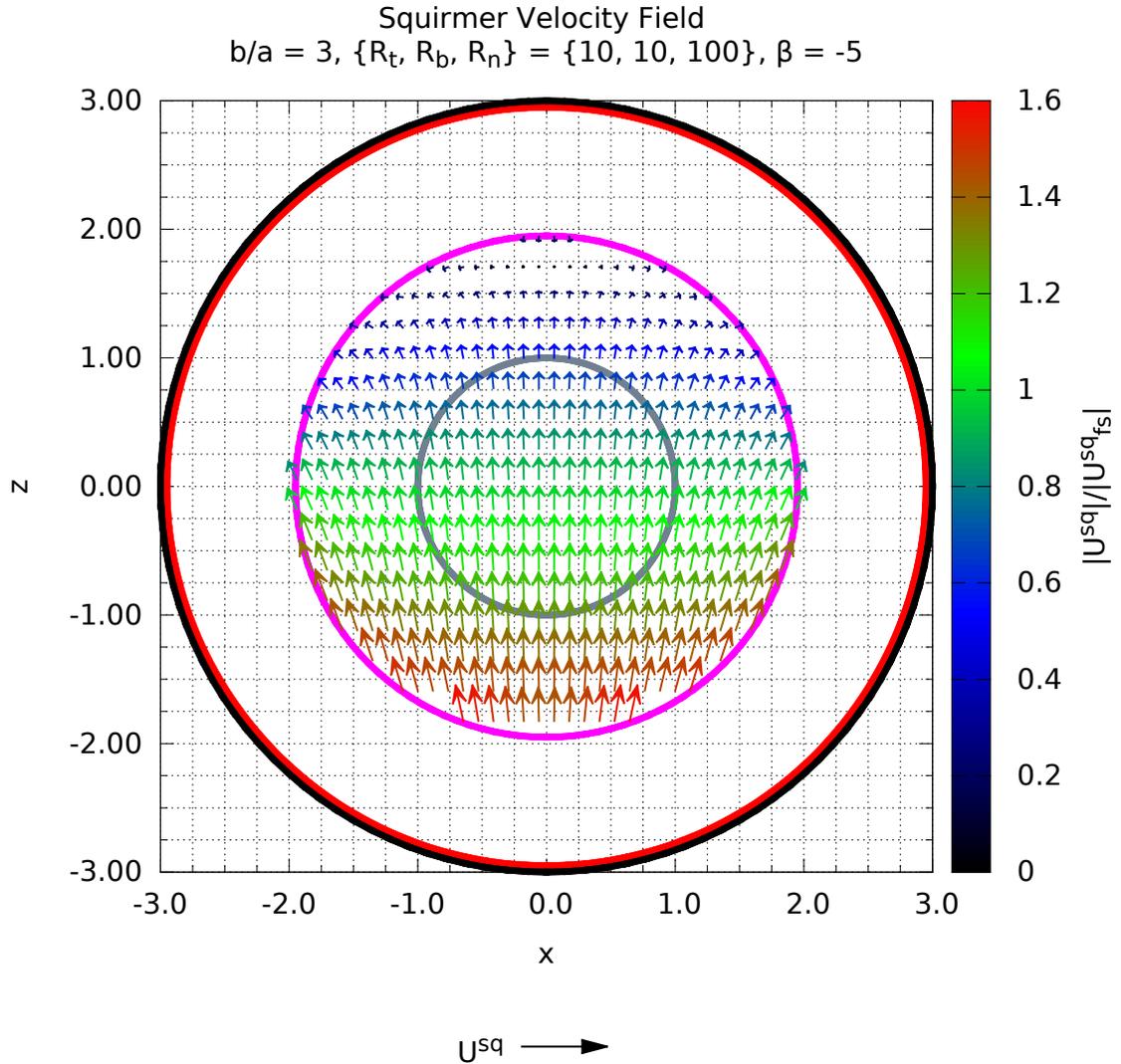


Fig. 5.8 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the tangential direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

The container velocity field shows a similar mirror symmetry reversal as was observed when switching from weak normal to weak tangential resistances. However,

this time this mirror symmetry flipping of \mathbf{U}^m across the x and z planes is caused by changes in $\beta = +5 \rightarrow -5$. This means that the observed translational motion of a container may be reversed simply by changing the type of squirmer inside from a pusher to a puller.

5.7.4.4 Effect of b/a on \mathbf{U}^m and \mathbf{U}^{sq}

The size ratio's effect on the mobility solution field is to make the solutions relatively more uniform near the container's center. Since the squirmer's velocity $\mathbf{U}^{sq} \rightarrow \mathbf{U}_{fs}^{sq}$ for $b/a \rightarrow \infty$, the squirmer tends to move faster especially when near the container's center. Since the velocity flow field decays as $1/r^2$ for pushers and pullers, the velocity of the container \mathbf{U}^m decays in a similar way though in proportion to the pressure field decay which goes like $1/r^3$. These effects simply scale the magnitude of \mathbf{U}^m . All mirror and reversal effects previously describe also manifest themselves for a larger b/a size ratio. Plots for $b/a = 5$, $\beta = \pm 5$ are shown in Figs. 5.46, 5.47, 5.48, 5.49, 5.50, 5.51, 5.52, 5.53. Plots for b/a , $\beta = 0$ are analogous to the $b/a = 3$ case are not shown. These figures have been constructed in the same way, with an equivalent grid resolution, up-sampling scheme, and interpolation procedure. Calculations were also performed for $b/a = 10$ and are available on request.

5.8 Squirmer and Container Trajectory Calculations

With the mobility solution and its up-sampled representation, \mathcal{M}^{P^0} , all the fluid mechanics of the squirmer/porous container problem has been pre-computed and solved. This a priori computation is relatively complex and time consuming, though as previously argued, it facilitates the fast computation of many arbitrary particle trajectories. At a given time, $t = t_k$, a squirmer's position may be calculated by transforming the squirmer's current parameterization \mathbf{P}_k into its standard parameterization \mathbf{P}_k^0 . This allows for the squirmer's position to be calculated in the reference xz plane. The mobility solution is then the interpolated \mathcal{M}^{P^0} field. Afterwards a transformation is computed which rotates and transforms the standard orientation, $\mathbf{e}_s = \mathbf{e}_Z$ to the time t_k orientation, $\mathbf{e}_{s,k}$.

The procedure may be summarized more formally as follows:

1. Given an arbitrary squirmer starting position, \mathbf{x}_c^{sq} and orientation \mathbf{e}_{sq} at time $t = t_k$, find the rotation matrix $\mathbf{R}_{r^{sq}}^{r^{sq},xz}$ such that $[\mathbf{R}_{r^{sq}}^{r^{sq},xz} \mathbf{r}^{sq}] = \mathbf{r}^{sq,xz} \in \text{span}(\mathbf{e}_X, \mathbf{e}_Z)$ for $\mathbf{r}^{sq} = \mathbf{x}_c^c - \mathbf{x}_c^c$.

2. Rotate the squirmer's orientation, $\mathbf{e}_{s,k}$, into the xz plane such that $\mathbf{e}_s \rightarrow \mathbf{e}_s^{xz}$. Then find the rotation matrix $\mathbf{R}_{\mathbf{e}_s^{xz}}^{\mathbf{e}_s^z}$ that transforms $\mathbf{e}_s^{xz} \rightarrow \mathbf{e}_s^z$.
3. Rotate $\mathbf{r}^{sq,xz}$ by $\mathbf{R}_{\mathbf{e}_s^{xz}}^{\mathbf{e}_s^z}$ to find the squirmer's reference position, $\mathbf{r}_0^{sq,xz}$, in \mathbf{P}_k^0 .
4. Search the κ up-sampled grid for $\mathbf{r}_0^{sq,xz}$ using an efficient method (ordered list bisection) and find spatial location bounds in the grid coordinates (i_l, j_l) , $l \in 0, 1$ such that $G^{\kappa n}(i_0, j_0) < \mathbf{x}_c^{sq,xz} \cdot \mathbf{e}_X < G^{\kappa n}(i_1, j_0)$ and $G^{\kappa n}(i_0, j_0) < \mathbf{x}_c^{sq,xz} \cdot \mathbf{e}_Z < G^{\kappa n}(i_0, j_1)$.
5. Use bicubic interpolation on the bounding grid square indexed by the set $\{(i_l, j_l) : l \in 0, 1\}$ (with counterclockwise ordering) to interpolate the standard mobility solution field at $\mathbf{r}_0^{sq,xz}$. Store this interpolated solution as the quantity $\mathcal{M}^{\mathbf{P}^0}(\mathbf{r}_0^{sq,xz})$.
6. Compute the rotation matrix, $\mathbf{R}_{\mathbf{e}_Z}^{\mathbf{e}_{s,k}}$ that transforms the standard orientation, \mathbf{e}_Z , to the squirmer's current orientation, $\mathbf{e}_{s,k}$.
7. Rotate the standard interpolated mobility solution by $\mathbf{R}_{\mathbf{e}_Z}^{\mathbf{e}_{s,k}}$ to find the time $t = t_k$ mobility solution, $\mathbf{R}_{\mathbf{e}_Z}^{\mathbf{e}_{s,k}} \mathcal{M}^{\mathbf{P}^0}(\mathbf{r}_0^{sq,xz}) = \mathcal{M}^{\mathbf{P}_k}(\mathbf{r}^{sq})$
8. Integrate both squirmer's and container's center of mass and orientation and repeat this procedure for time $t = t_{k+1}$.

The construction of the various rotation matrices results from asking the question of, "How can a given vector, \mathbf{a} , be rotated into a different vector \mathbf{b} ." One answer to this question is to use the well-known Rodrigues rotation formula, which is essentially an angle-axis rotation representation. Put differently, given a vector \mathbf{a} , an angle θ , and a unit vector axis of rotation $\hat{\mathbf{k}}$, the Rodrigues rotation formula rotates \mathbf{a} around axis $\hat{\mathbf{k}}$ by an angle θ . In component form the formula is given by,

$$\begin{aligned}
 a_i &= \mathbf{R}_{ik} b_k \\
 &= \left[\delta_{ik} + \sin(\theta) \epsilon_{ijk} \hat{k}_j + (1 - \cos(\theta)) [\epsilon_{ijk} \hat{k}_j]^2 \right] b_k \\
 &= \left[\delta_{ik} + |\hat{\mathbf{a}} \times \hat{\mathbf{b}}| \epsilon_{ijk} \hat{k}_j + (1 - |\hat{\mathbf{a}} \cdot \hat{\mathbf{b}}|) [\epsilon_{ijk} \hat{k}_j]^2 \right] b_k
 \end{aligned} \tag{5.101}$$

There are several cases where the formula fails during numerical implementation, but these cases may be avoided with proper implementation. It should also be understood that $\mathbf{k} = \hat{\mathbf{a}} \times \hat{\mathbf{b}}$ and $\hat{\mathbf{k}} = \mathbf{k}/|\mathbf{k}|$. This formula may be used to build $\mathbf{R}_{\mathbf{e}_Z}^{\mathbf{e}_{s,k}}$ by letting $\mathbf{a} = \mathbf{e}_Z$ and $\mathbf{b} = \mathbf{e}_{s,k}$. One may build $\mathbf{R}_{\mathbf{r}^{sq}}^{\mathbf{r}^{sq,xz}}$ by letting $\mathbf{a} = \hat{\mathbf{r}}_k^{sq}$ and setting \mathbf{b} to the unit vector of the projection of \mathbf{a} onto the xz plane.

In practice, Euler integration is used to simulate trajectories. A squirmer run-time may be defined as $t_{run} = a/|\mathbf{U}_{fs}^{sq}|$ from which a total simulation time may be calculated as $t_{tot} = t_{run}b/a$. This total time is roughly the time that it takes the squirmer to move a distance b . Since the squirmer's translational velocity is bounded above by \mathbf{U}_{fs}^{sq} , in practice, simulations are run for a constant multiple of t_{run} so that the squirmer has a chance to run across the container. A convenient way to generate trajectories, which densely sample squirmer starting positions $\mathbf{x}_{c,0}^{sq}$ is to initialize these positions with grid points $G^{\kappa n}(i, j)$. In total this produces $[(\kappa + 1)(n + 1)]^2$ starting positions. Under the standard parameterization \mathbf{P}^0 these positions all lie in the xz plane with orientation \mathbf{e}_z .

5.8.1 Squirmer Container Collisions

The long time behavior of all squirmer trajectories always shows squirmer and porous container contact. In all simulations shown here, squirmers will eventually end up moving towards the container walls, and tend to group at certain container boundary accumulation points. The existence of these accumulation points is real since the near field lubrication interactions have been rigorously sampled near the container walls at a gap size $a\delta$ for $\delta = 0.05$. The mechanism for squirmer and container contact is that at the accumulation points, the squirmer always maintains a non-zero radially outward velocity component relative to the container's motion. Various types of squirmers find these boundary accumulation points by sliding along the container walls. At these accumulation points, the squirmer orientation aligns with the outward radial vector.

Appropriate collision models are always a subject of intense debate since using different collision models often drastically changes near field system dynamics. In these simulations a very simple collision model is used. The squirmer surface is never allowed to move closer than $a\delta$ to the container surface. In other words, we enforce a constraint on the container to squirmer center of mass vector such that $|\mathbf{r}_c^{sq}| = |(\mathbf{x}_c^{sq} - \mathbf{x}_c^c)| \leq r_{cp}^{max}$. When a squirmer violates this condition, its position is moved backwards along the unit radial vector $\hat{\mathbf{r}}_c^{sq}$ (in general this would be the normal vector) by a scalar amount γ such that $|\gamma\hat{\mathbf{r}}_c^{sq} + \mathbf{r}_c^{sq}| = r_{cp}^{max}$. The squirmer's velocity and angular velocity (mobility solution) both remain unchanged. This idea is similar to what is normally used for reflecting collisions, where the velocity component parallel to the wall normal vector is reflected inward. This equation is quadratic in γ and may be easily solved. Keeping the squirmer's overall mobility solution unchanged allows the squirmer to explore a small space of mobility solutions near the collision point. This leads to a re-orientation mechanism by which the squirmer slides along the container wall.

5.8.2 3D Squirmer Trajectories

Full 3D squirmer trajectory plots for the \mathbf{P}^0 parameterization are shown in Figs. 5.62, 5.60, 5.61, 5.65, 5.63, 5.64, for various types of squirmers, permeability resistances, and a size ratio of $b/a = 5$. These plots provide insight into how squirmers behave globally at all positions inside of the container. They also illustrate an interesting and highly symmetric trajectory structure that is testament to the accuracy and precision of the Galerkin BEM numerics. It is important to note that these trajectories show the position of the particle relative to the container. In other words, the squirmer positions are shown in a coordinate frame that moves with the container. The relative position is used so that end points of a squirmer's trajectory directly represent the squirmers net translocation across the container.

Several general trends can be observed from these trajectory plots. A container with weak tangential resistance tends to cause stronger radial migrations in individual squirmers as they move upwards. Pullers ($\beta > 0$) tend to move radially inwards relative to the z axis as they swim up in z . On the other hand, pushers ($\beta < 0$) move radially outwards as they swim upwards in z . Neutral squirmers ($\beta = 0$) swim along relatively straight trajectories. If the container has weak normal resistance to flow, the squirmer trajectories show less of a tendency to move in the radial direction. However, pullers still tend to migrate radially inwards and pusher move radially outwards. Viewed differently, pushers tend to be attracted towards the container wall and pullers tend move away from the container walls. This squirmer and container attraction is purely a hydrodynamic effect.

These trajectories are slightly easier to understand and analyze if they are plotted in the xz plane (note that the 3D trajectories have a negligible y -component owing to the fact that component velocities $U_y^{sq} \approx 0$ and $U_y^m \approx 0$). To aid in this analysis, the squirmer trajectories are sorted by the initial x and z components of the starting position $\mathbf{x}_{c,0}^{sq}$. The trajectories are then sampled at several discrete values in z , starting as close as possible to $z = -b$, which span the radial direction from the container wall near $-x$ to the container wall at $+x$. Only a small subset of the full mobility field's trajectories are shown so as to aid in clear visualization.

Squirmer trajectories for weak tangential resistance to flow and $\beta \in \{-5, 0, 5\}$ are respectively shown in Figs. 5.54, 5.55, 5.56. For this type of membrane resistance there exists certain starting positions where a puller may become trapped and be unable to swim away from its initial position. These *sticky* initial positions are near the $-z$ axial pole. The puller's velocity reverses sign in this region forcing it to swim towards the container wall. Above a certain elevation in z , all pullers finish their swim and completely translocate to the other side of the container. Once at the container

wall, they proceed to slide inward along the container wall and migrate towards the z axis from both sides. There are two accumulation points for pullers when there is weak tangential resistance to permeable flow. These accumulation points are at $(0, 0, \pm b)$ in the \mathbf{P}^0 parameterization. On the other hand, pushers, in containers with weak tangential resistance to flow, nearly all successfully translocate several run lengths across the container. However, all trajectories tend to move away from the pole at $(0, 0, +b)$ towards two different poles that are symmetric with respect to the yz plane. Neutral squirmers migrate towards the $+z$ pole behaving like pullers but without *sticky* initial positions.

Squirmers trajectories for weak normal resistance to flow and $\beta \in \{-5, 0, 5\}$ are respectively shown in Figs. 5.57, 5.58, 5.59. For weak normal resistance to flow, all starting positions of pullers are now able to completely translocate the container. All puller trajectories end up running into the container wall and accumulate at the axial node $(0, 0, +b)$. These trajectories move inwards towards the z -axis and proceed upwards in z until they run into the container wall. Pushers again migrate towards two container nodes, though this time at sharper (more acute) angles relative to the yz plane. Pusher trajectories tend to spread in z and move away from the z -axis. All pushers are able to translocate the container. Finally, neutral squirmers show a relatively straight trajectory and all migrate towards the axial node at $(0, 0, +b)$.

With the fully 3D trajectories, angular velocity, which is predominantly in the \mathbf{e}_Y direction, may be attached to the trajectory and visualized. The squirmer angular velocity $\boldsymbol{\Omega}^{sq}$ is scaled by $b|\mathbf{U}^m|$ using either Eqn. 5.99 or 5.100. Trajectory fields supplemented with angular velocity data are shown in for weak tangential flow in Figs. 5.66, 5.67, 5.68, and for weak normal resistance to flow in Figs. 5.69, 5.70, 5.71. The most important observation from these plots is that squirmers on opposite sides of the yz plane have anti-parallel angular velocities and so rotate in opposite directions. This anti-parallel rotation either results in a squirming orientation that causes accumulation towards a point or depletion away from a point. This is the fundamental mechanism that drives squirmer trajectories towards the $+z$ pole (or towards any other accumulation point).

5.8.3 3D Container Trajectories

Permeable container walls give rise to finite container velocities and thus finite container motion. Full 3D container trajectories in world space are shown for weak tangential resistance to flow and $\beta \in \{-5, 0, 5\}$ in Figs. 5.72, 5.73, 5.74. Analogous plots for weak normal resistance to flow are shown in Figs. 5.75, 5.76, 5.77. These plots correspond to the 3D squirmer trajectories seen previously. These trajectory plots

serve to illustrate the unique, often non monotonic motion of the container. General trends which influence net container parallel swimming or anti-parallel swimming can be observed by examining the net displacement of the container. Parallel swimming or co-swimming means that the container and squirmer both end up swimming a net-positive distance in the sense that they both move upwards in z . Anti-parallel swimming means that the container swims a net-negative (net-positive) distance and the squirmer swims a net-positive (net-negative) distance in the sense that the container moves down (up) in z and the container moves up (down) in z . These notions of co-swimming and anti-parallel swimming are to be understood only up to the point where the squirmer's motion stagnates at the container boundary.

For weak normal permeable flow, container trajectory plots show that pullers induce net anti-parallel swimming of the container. Pushers on the other hand, induce co-swimming. Even though the container trajectories are non-monotonic, the net motion is monotonic across all cases. Neutral squirmers always induce anti-parallel swimming of the container. Though not obvious from these trajectory plots, out of the three types of squirmers, pullers will always translocate the container the fastest since they enjoy the added benefit of net-negative container swimming distances.

For weak tangential flow, the situation is more complex. Pullers now induce co-swimming of the container except for the so-called *sticky* squirmer positions. Certain *sticky* squirmer positions that are very near the pole at $-z$ however do result in container co-swimming in the $-z$ direction though some less *sticky* positions give rise to net anti-parallel swimming. Pushers always induce anti-parallel swimming of the container. Neutral squirmers always show co-swimming of the container. These results are summarized in Table 5.1.

Squirmer Type	Weak Normal Flow	Weak Tangential Flow
$\beta = +5$	$\uparrow\downarrow$	$\uparrow\uparrow, \uparrow\downarrow, \downarrow\downarrow$
$\beta = 0$	$\uparrow\downarrow$	$\uparrow\uparrow$
$\beta = -5$	$\uparrow\uparrow$	$\uparrow\downarrow$

Table 5.1 Net squirmer and container swimming directions are shown for the \mathbf{P}^0 parameterization relative to z . An up arrow, \uparrow indicates net motion in $+z$. A down arrow, \downarrow , indicates net motion in $-z$. The first and second arrows represent the net directional motion of the squirmer and container respectively. Arrows are to be understood as the overall direction of the net motion of the squirmer and container for sufficiently long trajectories.

5.8.4 Net and Total Distance, Fastest Squirmer Translocation

The net translocation distance that the squirmer travels relative to the container is also an interesting quantity to study. Given a certain type of squirmer, this quantity gives insight into which container resistance conditions allows this squirmer to translocate most efficiently over a fixed net distance d_n^β . Naturally this net distance is taken to be $2r_{cp}^{max}$. The total distance traveled, d_t^β , relative to world space is also plotted versus time. This quantity is nearly always equal to the net distance traveled, even over several squirmer run times since the container velocities are much smaller than the squirmer velocities at these b/a size ratios. However these two quantities start to diverge as soon as the squirmer starts to repeatedly collide with the container's boundary. For weak tangential resistance to permeable flow, the squirmer's net and relative distances are shown in Figs. 5.78, 5.79, and 5.80. The cases where there is weak normal resistance to permeable flow are shown in Figs. 5.81, 5.82, and 5.83. Each distance plot uses the same colors as the trajectory plots.

One must be careful when comparing these plots, especially at long times. Each trajectory shows unique spatial dynamics and is neither globally convex nor concave. Also, at long times, not all trajectories make it to the container walls. This is particularly evident in the case of weak tangential resistance to permeable flow where pushers tend to *stick* near the negative axial pole at $-z$. The 3D container trajectories show that for weak tangential resistance to permeable flow, pushers swim net anti-parallel and pullers show net co-swimming relative to the container. However, this understanding is not enough to make conclusions about which squirmers will translocate most quickly.

The quantity $\Delta d_n^{\beta_1, \beta_2} = (d_n^{\beta_1} - d_n^{\beta_2})/a$ compares net swimming distances between two squirmer types. This may be computed for all combinations of weak normal and weak tangential resistances and $\beta \in \{-5, 0, 5\}$. The quantity $\Delta d_n^{-5, 5}$ is shown in Figs. 5.84 and 5.85 for respectively weak tangential and weak normal resistances. These two figures show a general trend where pushers, at early times, translocate over a much larger relative distance than pullers. In the long time, pullers become more efficient at translocating the container, in part due to the fact that pushers tend to attract towards the container wall.

Holding the type of squirmer fixed, Fig. 5.86 shows that at early times, a pusher swims most efficiently when there is weak tangential resistance. At long times, the pusher swims more efficiently in a container with weak normal resistance to permeable flow. On the other hand, Fig. 5.87 shows that pullers almost always are most efficient when the container has weak normal resistance to permeable flow.

5.9 Discussion and Conclusions

The dynamics of an active squirmer particle inside of a rigid container are incredibly rich and complex. In this study we analyzed this system in great detail starting with the simple concentric geometry case. An analytical solution of the Stokes equations was obtained for several types of container boundary conditions. An active squirming particle was first studied in the context of a non-porous container. The obtained analytical solution describing the squirmers motion was compared with the solution for the case where a forced particle moves inside of a rigid non-porous container. The velocity of the squirmer is much less influenced by the container's hydrodynamic effects when compared to the velocity of a forced particle. This interesting phenomena is attributed to the fact that the velocity field associated with a squirmer decays more rapidly in free space $\propto 1/r^{2,3}$ than the point force free space solution $\propto 1/r$, and so squirmer will tend to move in a less hindered manner sooner than the forced particle as $b/a \rightarrow \infty$.

Next, a new novel model was proposed for describing fluid flow across an infinitesimal porous rigid container. The underlying basis of this model is Darcy's law, but this porous container model is more general and also allows for discontinuous jumps in both the normal and tangential stresses. Fluid is thus allowed to pass through the container surface with both finite normal and tangential velocity components (relative to the container surface). In the concentric geometry, this model also admits an exact analytical solution. This solution is parameterized by tangential (parallel) and normal (perpendicular) resistance coefficients, which are respectively given by R_{\parallel} and R_{\perp} . By varying the magnitudes of these resistance coefficients various limiting cases can be successfully recovered. In particular, the previously solved non-porous container and squirmer case is given by the limit where $R = \{R_{\parallel}, R_{\perp}\} \rightarrow \{\infty, \infty\}$. Plots of these solutions for various values of the container to squirmer size ratio, b/a , showed non-trivial and interesting results. In all resistance combinations, $\{R_{\parallel}, R_{\perp}\}$, there exists a global minimum value in the squirmer's translational velocity at a particular size ratio $(b/a)_{min}$. Roughly speaking, this is understood as the size ratio where the squirmer is most inefficient in expelling fluid through the porous container. Velocity flow fields with streamlines for the squirmer and porous container were generated from the exact analytical solutions. Each flow field was seen to be heavily influenced by the type of squirmer (described by the parameter β). Each flow field has non-trivial and highly interesting features. Pushers, defined by $\beta < 0$ tend to draw fluid in along the radial direction and expel it axially. They do so asymmetrically so as to push more net flow towards the $-z$ direction and generate net thrust in $+z$. Pullers, defined by $\beta > 0$, swim by the opposite mechanism and tend to draw fluid in along

the axial directions and expel it out radially. Again this is done axisymmetrically so as to generate net thrust in $+z$. However, pushers and pullers both create vortical flows in the axial z -anterior and z -posterior portions of the container. This effect is purely of hydromechanical origin. For weak resistance to tangential permeable flow, these vortical flows were seen to extend outside of the membrane. These vortical flows caused the container to move in the same direction as the squirmer. Conversely for weak normal permeable resistances to flow, the container was seen to translate in the direction opposite of the squirmer's motion. In this case, fluid exits the container with a predominantly normal component and the net thrust of the squirmer in $+z$ acts to convectively push the container in the opposite direction.

Next, a rigorous and novel boundary integral formulation was derived for the squirmer and porous container problem. A completed double-layer boundary integral representation was used to describe the squirmer dynamics. However, it is important to point out that no force or torque completion is necessary. Since the squirmer is by definition, a force and torque free body, it can be perfectly represented by the Stokes double layer potential \mathcal{K} . The container BIE representation was expressed using a Stokes single layer potential, \mathcal{G} , and a surface traction discontinuity $\Delta\mathbf{f}$. This term may be written using the newly proposed porous membrane model. However, a more general description of the flow needs to be made to account for a fluid flow in a second tangential basis direction. The porous resistances were subsequently generalized under the identification $R = \{R_t, R_b, R_n\} = \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$ with tangent, bitangent, and normal permeable resistances given respectively by R_t, R_b, R_n . This description of the container yields a second kind BIE equation making the overall description of the coupled system second kind although with container force and torque free constraints. It is likely that these constraints can be built efficiently into the container description and still yield an integral equation of the second kind. This may be attempted in future work.

The coupled system of boundary integral equations describing the dynamics of the porous container and squirmer system was discretized under the Galerkin method. While this discretization method has been well-known since the inception of the finite element method, it is almost never applied to BEM and never to multi-body hydrodynamics problems in Stokes flow. A detailed construction of this discretization is described. The successful implementation of this discretization and overall solution procedure is highly non-trivial. The squirmer porous container problem is solved under this discretization using the authors GPUGBEM framework. Without this framework, solution of this problem would be highly inefficient and very error prone. All boundary element calculations are performed on CUDA enabled GPUs. Accurate calculations

are enforced using a novel adaptive local meshing algorithm that preserves the mesh manifold property and thus C^0 continuity of all underlying unknown boundary datums. Detailed matrix representations of all BEM operators are given and discussed in detail. Each operator construction is novel and highly non-trivial.

After forming the overall GBEM linear system, numerical solutions are computed for the perfectly concentric geometry. Numerical results are found to be in excellent agreement with the exact analytical model. Numerical solutions are obtained down to the size ratio $b/a = 1.1$. The squirmer's translational velocity is seen to be bounded below by the solution for the non-porous container. Global minimums in the squirmer's velocity, seen in the analytical model, are recovered in the GBEM numerics.

In order to fully characterize the squirmer's and container's dynamics, trajectories are computed using a highly accurate, efficient, and fast interpolation procedure. An important symmetry observation of the system is made, namely that the container and squirmer always share a symmetry plane that contains both of their individual centers of mass, and that the mobility solution is only unique up to how the squirmer is oriented. If the fluid mechanics can be resolved in this single plane at a standard squirming orientation, then the fluid mechanics is known in all space up to a rigid body rotation. This allows for simulation and calculation of trajectories for arbitrary squirmer orientations and positions and is independent of concurrent fluid mechanics calculations. The planer grid containing all the relevant fluid mechanics information is referred to as a *mobility solution field*. The mobility solution fields are constructed in a standard reference configuration, namely the xz plane, and with a standard squirmer orientation \mathbf{e}_z , though these two selections are completely arbitrary. Next mobility solution fields are used to interpolate squirmer and container trajectories for a variety of size ratios, resistance parameters, and squirmer types. Full three dimensional trajectories are constructed for both the squirmer and the contained.

Squirmer and container trajectories are examined in great detail. The trajectories move in ways that could only be predicted by the numerics. Non-intuitive but smooth motions result when the squirmer and container come in close contact. Several general trends in the trajectories are observed, namely that weak tangential resistance causes radial migrations relative to the z -axis. Under this type of resistance pullers move radially inward and up towards the axial $+z$ pole. However there are some *sticky* starting positions where a puller may become trapped in the container's hydrodynamic grasp and possibly never be able to make it past the axial equatorial plane (defined here by the xy plane). Conversely, pushers tend to move radially outward. For weak normal resistance to flow, the *sticky* positions in the puller case disappear and all

pullers successfully translocate across the container. Weak normal resistance to flow has the general effect of reducing radial spread in the squirmer trajectories.

The net distance that a squirmer travels relative to the container is examined as a function of time. This net distance is used as a metric measuring a squirmers efficiency in exploring the fluid inside the container. With respect to this metric, it is shown that pushers move most efficiently at short times. Pushers tend to run into the container walls, or slow down as they cross the equatorial container plane. Pullers then become more efficient in exploring the container. This swimming efficiency is asymmetric in the total trajectory time, showing that pushers are more efficient over longer portions of their trajectories.

References

- [1] Ando, T., Chow, E., and Skolnick, J. (2013). Dynamic simulation of concentrated macromolecular solutions with screened long-range hydrodynamic interactions: Algorithm and limitations. *The Journal of Chemical Physics*, 139(12):121922.
- [2] Ando, T. and Skolnick, J. (2010). Crowding and hydrodynamic interactions likely dominate in vivo macromolecular motion. *Proceedings of the National Academy of Sciences*, 107(43):18457–18462.
- [3] Ando, T. and Skolnick, J. (2013). On the Importance of Hydrodynamic Interactions in Lipid Membrane Formation. *Biophysical Journal*, 104(1):96–105.
- [4] Baraff, D. (2001). Physically Based Modeling Rigid Body Simulation Rigid Body Simulation. *Simulation*, 2(1):2–1.
- [5] Barnes, J. and Hut, P. (1986). A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(6096):446–449.
- [6] Beavers, G. S. and Joseph, D. D. (1967). Boundary conditions at a naturally permeable wall. *Journal of Fluid Mechanics*, 30(01):197.
- [7] Blake, J. R. (1971). A spherical envelope approach to ciliary propulsion. *Journal of Fluid Mechanics*, 46(01):199.
- [8] Boedec, G., Leonetti, M., and Jaeger, M. (2011). 3D vesicle dynamics simulations with a linearly triangulated surface. *Journal of Computational Physics*, 230(4):1020–1034.
- [9] Brebbia, C. A. (1978). *The boundary element method for engineers*. John Wiley & Sons, Incorporated.
- [10] Cignoni, P., Montani, C., Rocchini, C., and Scopigno, R. (2003). External memory management and simplification of huge meshes. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):525–537.
- [11] Corona, E., Greengard, L., Rachh, M., and Veerapaneni, S. (2016). An integral equation formulation for rigid bodies in Stokes flow in three dimensions. *Journal of Computational Physics*, 332:1–20.
- [12] Elcock, A. H. (2010). Models of macromolecular crowding effects and the need for quantitative comparisons with experiment. *Current opinion in structural biology*, 20(2):196–206.

- [13] Erichsen, S. and Sauter, S. A. (1998). Efficient automatic quadrature in 3-d Galerkin BEM. *Computer Methods in Applied Mechanics and Engineering*, 157(97):215–224.
- [14] Frembgen-Kesner, T. and Elcock, A. H. (2009). Striking Effects of Hydrodynamic Interactions on the Simulated Diffusion and Folding of Proteins. *Journal of Chemical Theory and Computation*, 5(2):242–256.
- [15] Geller, A., Mondy, L., Rader, D., and Ingber, M. (1993). Boundary element method calculations of the mobility of nonspherical particles—1. Linear chains. *Journal of Aerosol Science*, 24(5):597–609.
- [16] Greengard, L. (1988). *The Rapid Evaluation of Potential Fields in Particle Systems*. ACM distinguished dissertations. MIT Press.
- [17] Greengard, L. and Rokhlin, V. (1997). A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 135(2):280–292.
- [18] Hackbusch, S. B. . L. G. . W. (2003). Hierarchical matrices. Technical report, Max Planck Institute for Mathematics in the Sciences.
- [19] Happel, J. and Brenner, H. (1981). *Low Reynolds number hydrodynamics*, volume 1 of *Mechanics of fluids and transport processes*. Springer Netherlands, Dordrecht.
- [20] Hestenes, M. R. and Stiefel, E. (1952). Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436.
- [21] Hiptmair, R. and Kielhorn, L. (2012). BETL – A generic boundary element template library.
- [22] Hsiao, G. C. and Wendland, W. L. (2008). *Boundary Integral Equations*, volume 164 of *Applied Mathematical Sciences*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [23] Ingber, M., Womble, D., Geller, A., Rader, D., and Mondy, L. (1999). Boundary element method calculations of the mobility of nonspherical particles—3. Parallel implementation for long chains. *Journal of Aerosol Science*, 30(1):127–130.
- [24] Ingber, M. S. and Mondy, L. A. (1993). Direct second kind boundary integral formulation for Stokes flow problems. *Computational Mechanics*, 11(1):11–27.
- [25] Ishikawa, T., Locsei, J. T., and Pedley, T. J. (2008). Development of coherent structures in concentrated suspensions of swimming model micro-organisms. *Journal of Fluid Mechanics*, 615:401.
- [26] Ishikawa, T. and Pedley, T. J. (2007). The rheology of a semi-dilute suspension of swimming model micro-organisms. *Journal of Fluid Mechanics*, 588:399–435.
- [27] Ishikawa, T., Simmonds, M. P., and Pedley, T. J. (2006). Hydrodynamic interaction of two swimming model micro-organisms. *Journal of Fluid Mechanics*, 568:119.
- [28] Jones, I. (1973). Low Reynolds number flow past a porous spherical shell. *Mathematical Proceedings of the Cambridge Philosophical Society*, 73(01):231–238.

- [29] Karr, J. R., Sanghvi, J. C., Macklin, D. N., Gutschow, M. V., Jacobs, J. M., Bolival, B., Assad-Garcia, N., Glass, J. I., and Covert, M. W. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389–401.
- [30] Karrila, S. J., Fuentes, Y. O., and Kim, S. (1989). Parallel Computational Strategies for Hydrodynamic Interactions Between Rigid Particles of Arbitrary Shape in a Viscous Fluid. *Journal of Rheology (1978-present)*, 33(6):913–947.
- [31] Karrila, S. J. and Kim, S. (1989). Integral Equations of the Second Kind for Stokes Flow: Direct Solution for Physical Variables and Removal of Inherent Accuracy Limitations. *Chemical Engineering Communications*, 82(1):123–161.
- [32] Keaveny, E. E. and Shelley, M. J. (2011). Applying a second-kind boundary integral equation for surface tractions in Stokes flow. *Journal of Computational Physics*, 230(5):2141–2159.
- [33] Kim, S. and Karrila, S. J. (2013). *Microhydrodynamics: Principles and Selected Applications*. Dover Civil and Mechanical Engineering. Dover Publications.
- [34] Kirk, B. S., Peterson, J. W., Stogner, R. H., and Carey, G. F. (2006). libMesh : a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3-4):237–254.
- [35] Kress, R. (2014). *Linear Integral Equations*, volume 82 of *Applied Mathematical Sciences*. Springer New York, New York, NY.
- [36] Lamb, S. H. (1932). *Hydrodynamics*.
- [37] Lighthill, M. J. (1952). On the squirming motion of nearly spherical deformable bodies through liquids at very small reynolds numbers. *Communications on Pure and Applied Mathematics*, 5(2):109–118.
- [38] Martys, N., Bentz, D. P., and Garboczi, E. J. (1994). Computer simulation study of the effective viscosity in Brinkman’s equation. *Physics of Fluids*, 6(4):1434–1439.
- [39] McGuffee, S. R. and Elcock, A. H. (2010). Diffusion, Crowding & Protein Stability in a Dynamic Molecular Model of the Bacterial Cytoplasm. *PLoS Computational Biology*, 6(3):e1000694.
- [40] Mitchell, W. F. (2016). 30 years of newest vertex bisection. volume 11, page 020011.
- [41] Mitchell, W. F. and McClain, M. A. (2014). A Comparison of hp -Adaptive Strategies for Elliptic Partial Differential Equations. *ACM Transactions on Mathematical Software*, 41(1):1–39.
- [42] Mondy, L. A., Geller, A. S., Rader, D. J., and Ingber, M. (1996). Boundary element method calculations of the mobility of nonspherical particles—II. Branched chains and flakes. *Journal of Aerosol Science*, 27(4):537–546.
- [43] Nazockdast, E., Rahimian, A., Zorin, D., and Shelley, M. (2016). A fast platform for simulating flexible fiber suspensions applied to cell mechanics.

- [44] Nguyen, H. and Corporation, N. (2008). *GPU Gems 3*. Number v. 3 in Lab Companion Series. Addison-Wesley.
- [45] Pak, O. S. and Lauga, E. (2014). Generalized squirring motion of a sphere. *Journal of Engineering Mathematics*, 88(1):1–28.
- [46] Phan-Thien, N. and Fan, X. J. (1996). A boundary integral formulation for elastically deformable particles in a viscous fluid. *ZAMP Zeitschrift for angewandte Mathematik und Physik*, 47(5):672–694.
- [47] Phan-Thien, N. and Kim, S. (1994). *Microstructures in Elastic Media: Principles and Computational Methods*. Oxford University Press, Incorporated.
- [48] Phan-Thien, N. and Tullock, D. (1994). Completed double layer boundary element method in elasticity and stokes flow: Distributed computing through PVM. *Computational Mechanics*, 14(4):370–383.
- [49] Phan-Thien, N., Tullock, D., Ilic, V., and Kim, S. (1992a). Completed double layer boundary element method: a numerical implementation and some experimental results. *Computational Mechanics*, 10(6):381–396.
- [50] Phan-Thien, N., Tullock, D., and Kim, S. (1992b). Completed double layer in half-space: a boundary element method. *Computational Mechanics*, 9(2):121–135.
- [51] Power, H. and Miranda, G. (1987). Second Kind Integral Equation Formulation of Stokes’ Flows Past a Particle of Arbitrary Shape. *SIAM Journal on Applied Mathematics*, 47(4):689–698.
- [52] Power, H. and Wrobel, L. C. (1995). *Boundary integral methods in fluid mechanics*. Computational Engineering Series. Computational Mechanics Publications.
- [53] Pozrikidis, C. (1992). *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge Texts in Applied Mathematics. Cambridge University Press.
- [54] Pozrikidis, C. (2002). *A Practical Guide to Boundary Element Methods with the Software Library BEMLIB*. Taylor & Francis.
- [55] Reigh, S. Y., Zhu, L., Gallaire, F., and Lauga, E. (2017). Swimming with a cage: low-Reynolds-number locomotion inside a droplet. *Soft Matter*, 13(17):3161–3173.
- [56] Saad, Y. and Schultz, M. H. (1986). GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869.
- [57] Saffman, P. G. (1971). On the Boundary Condition at the Surface of a Porous Medium. *Studies in Applied Mathematics*, 50(2):93–101.
- [58] Sauter, S. a. and Schwab, C. (2011). *Boundary Element Methods*, volume 39 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [59] Shelley, M. J. (2016). The Dynamics of Microtubule/Motor-Protein Assemblies in Biology and Physics. *Annual Review of Fluid Mechanics*, 48(1):487–506.

-
- [60] Śmigaj, W., Betcke, T., Arridge, S., Phillips, J., and Schweiger, M. (2015). Solving Boundary Integral Problems with BEM++. *ACM Transactions on Mathematical Software*, 41(2):1–40.
- [61] Solin, P., Segeth, K., and Dolezel, I. (2003). *Higher-Order Finite Element Methods*. Studies in Advanced Mathematics. CRC Press.
- [62] Solín, P. and Solín, P. (2006). *Partial Differential Equations and the Finite Element Method*.
- [63] Sonneveld, P. (1989). CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear systems. *SIAM Journal on Scientific and Statistical Computing*, 10(1):36–52.
- [64] Steinbach, O. (2008). *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer New York, New York, NY.
- [65] Stone, H. A. and Samuel, A. D. T. (1996). Propulsion of Microorganisms by Surface Distortions. *Physical Review Letters*, 77(19):4102–4104.
- [66] van der Vorst, H. A. (1992). Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644.

Fig. 5.9 In this figure, an example output of the h-adaptive meshing algorithm, based on the nearly singular distance ration $d_{xy} \leq 1$ is shown for two spheres. Colors are related to triangle diameters. This application of NVB to BEM results in a conformal mesh (i.e. there are no hanging nodes or so-called T-edges).

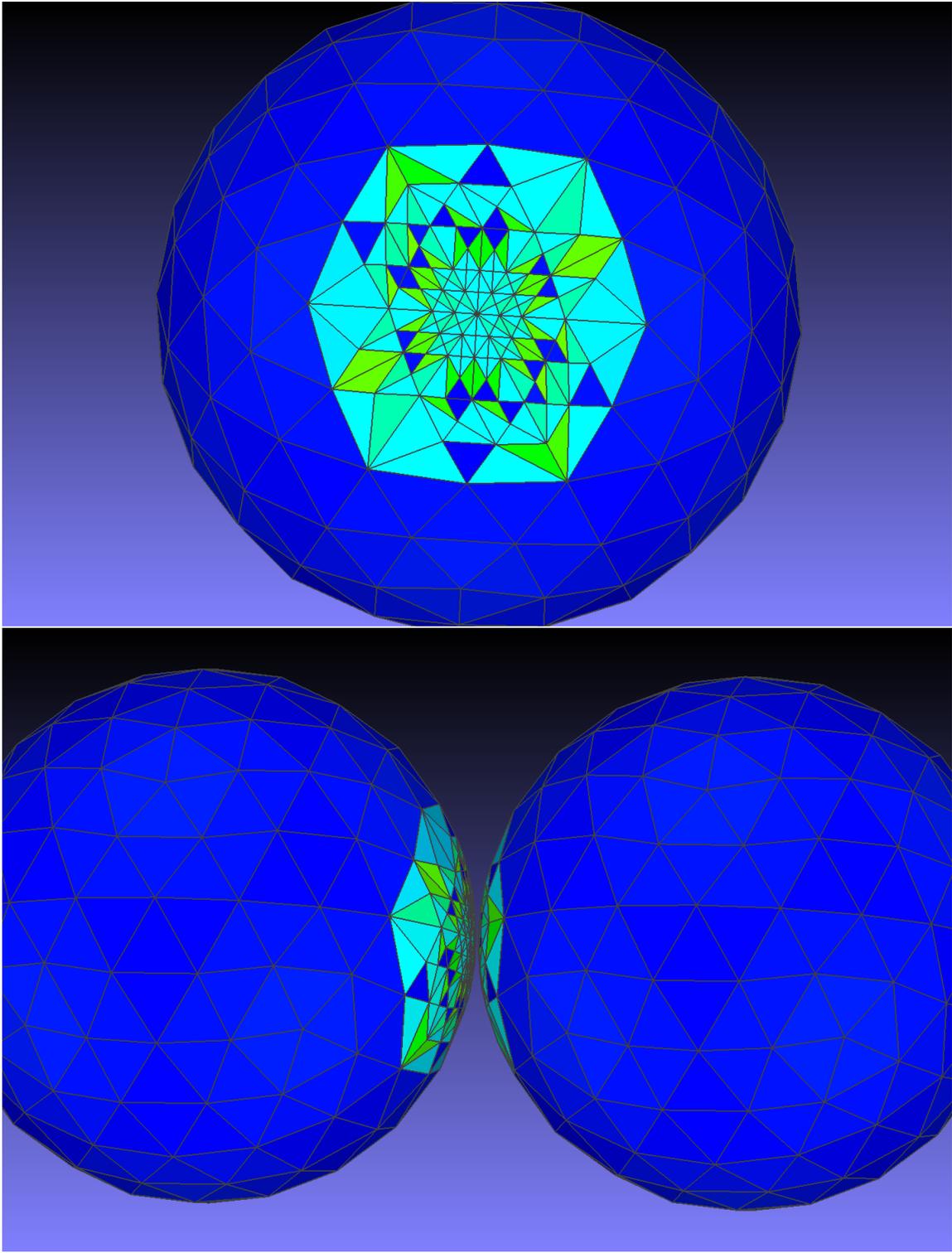


Fig. 5.10 The normalized z-component of the squirmer’s translational velocity, U_z^{sq} , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} > R_{\perp}$.

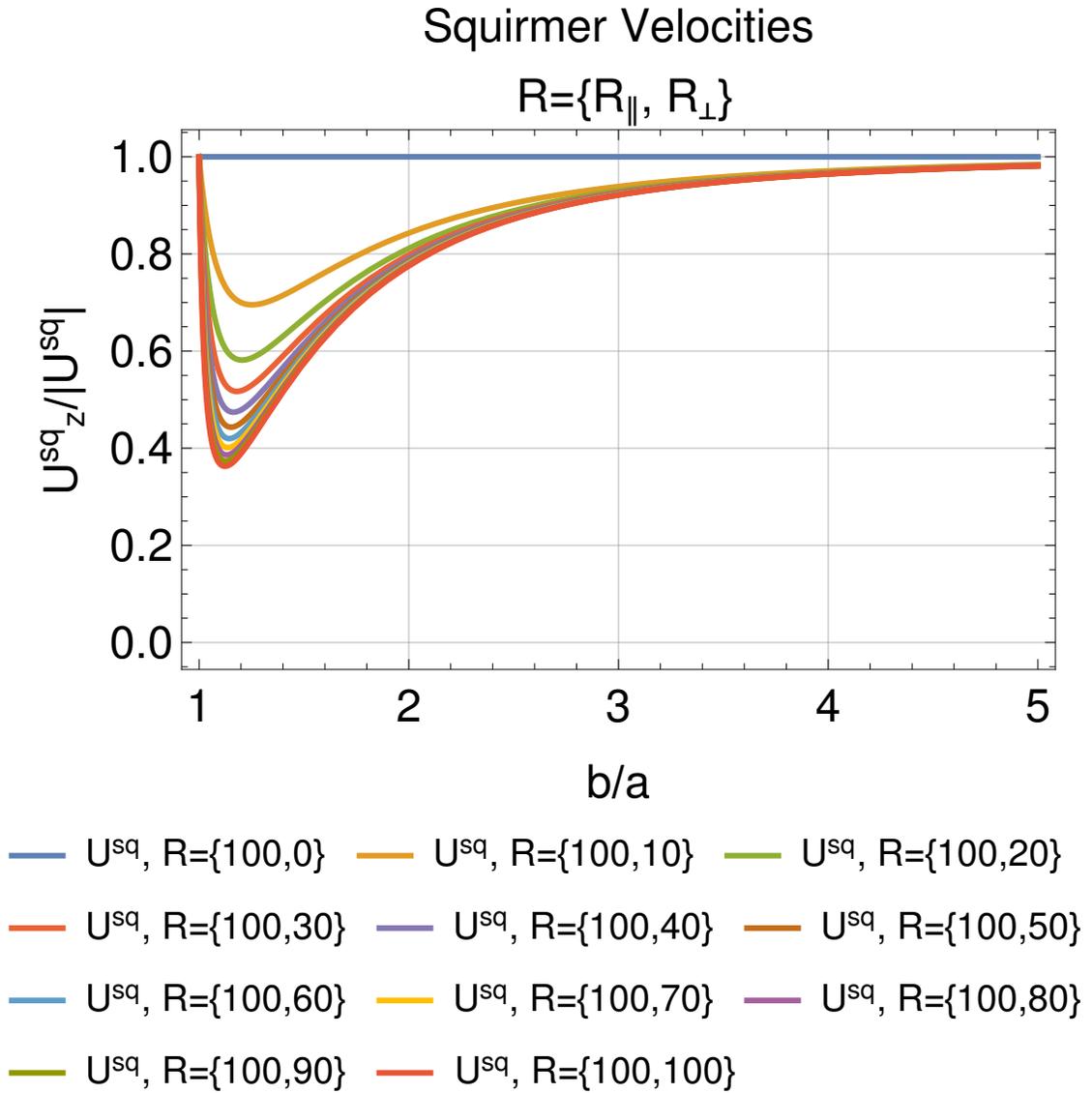


Fig. 5.11 The normalized z-component of the container's translational velocity, U_z^m , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} > R_{\perp}$.

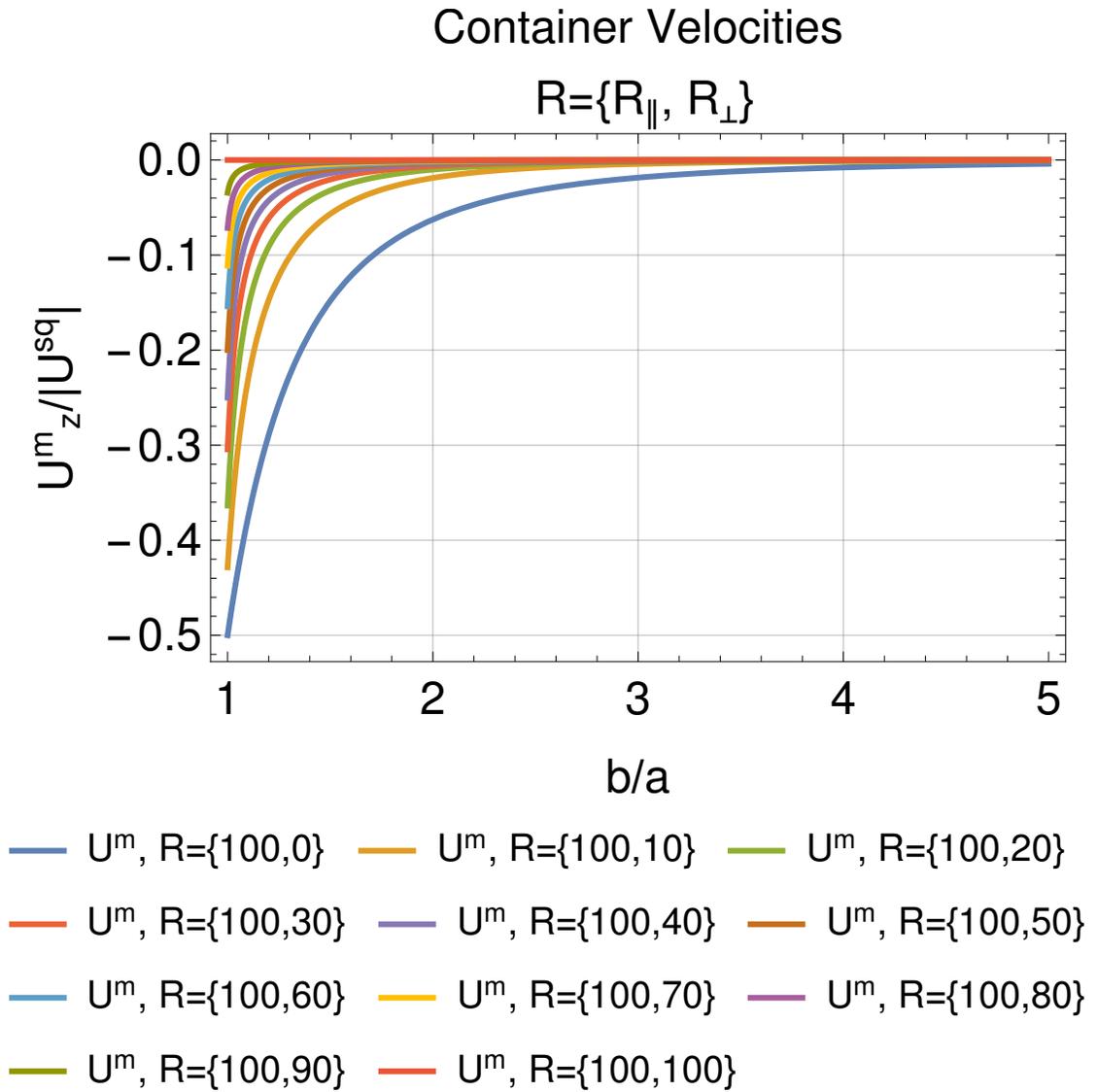


Fig. 5.12 The normalized z-component of the squirmer’s translational velocity, U_z^{sq} , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} < R_{\perp}$.

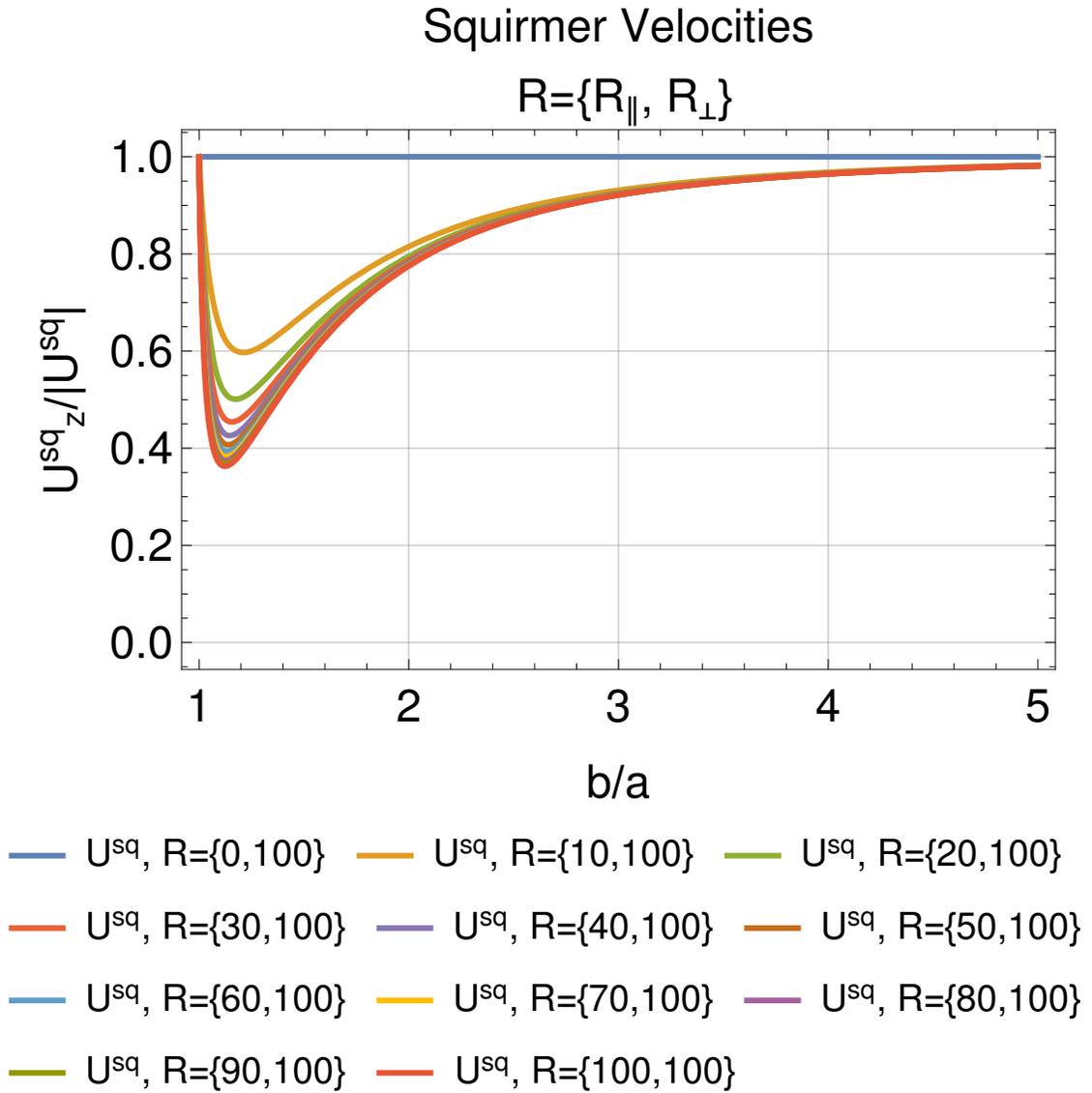


Fig. 5.13 The normalized z-component of the container’s translational velocity, U_z^m , is shown versus the container to squirmer size ratio b/a . Curves are shown for various sets of permeable resistances, $R = \{R_{\parallel}, R_{\perp}\}$ where $R_{\parallel} < R_{\perp}$.

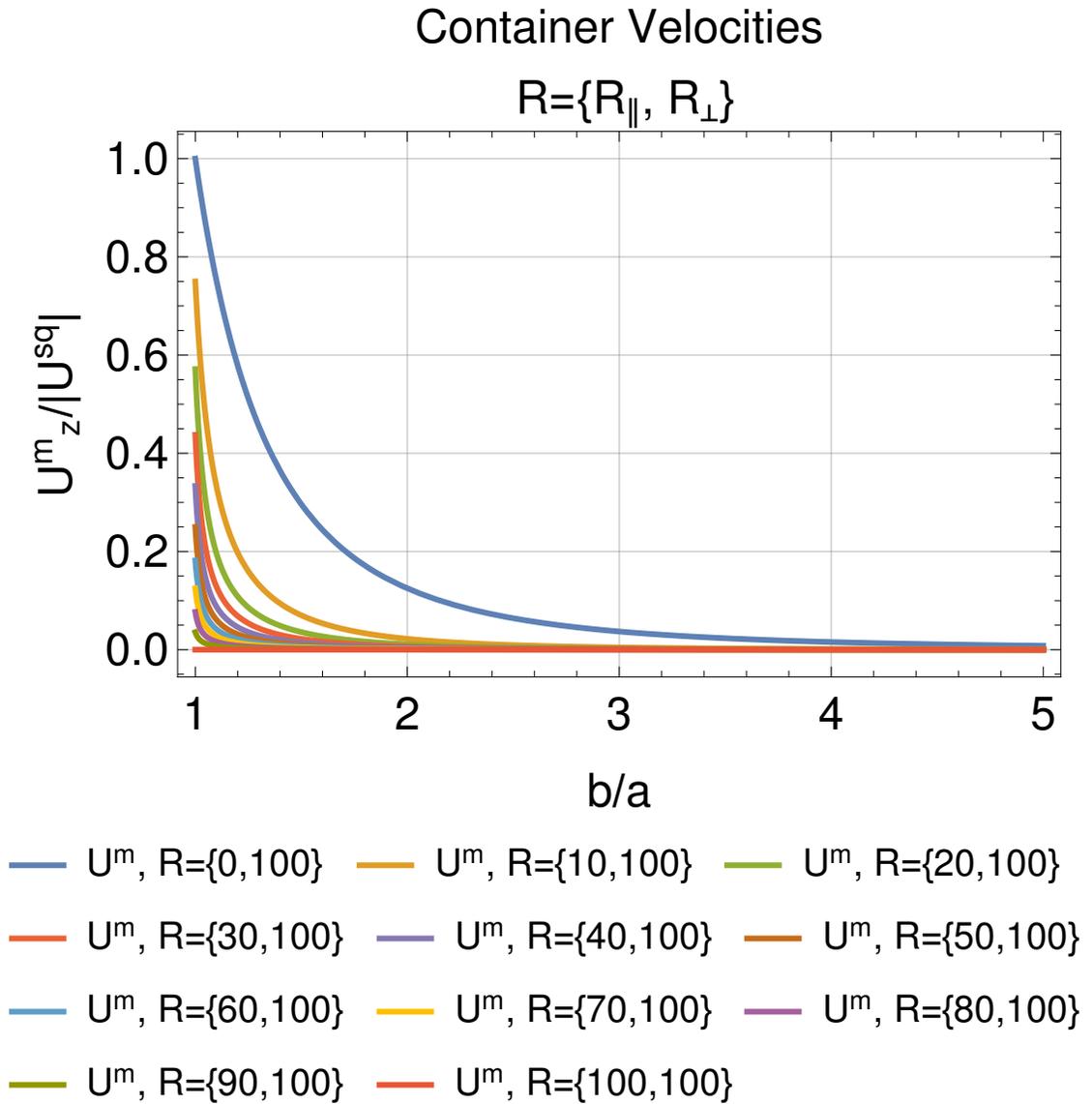


Fig. 5.14 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

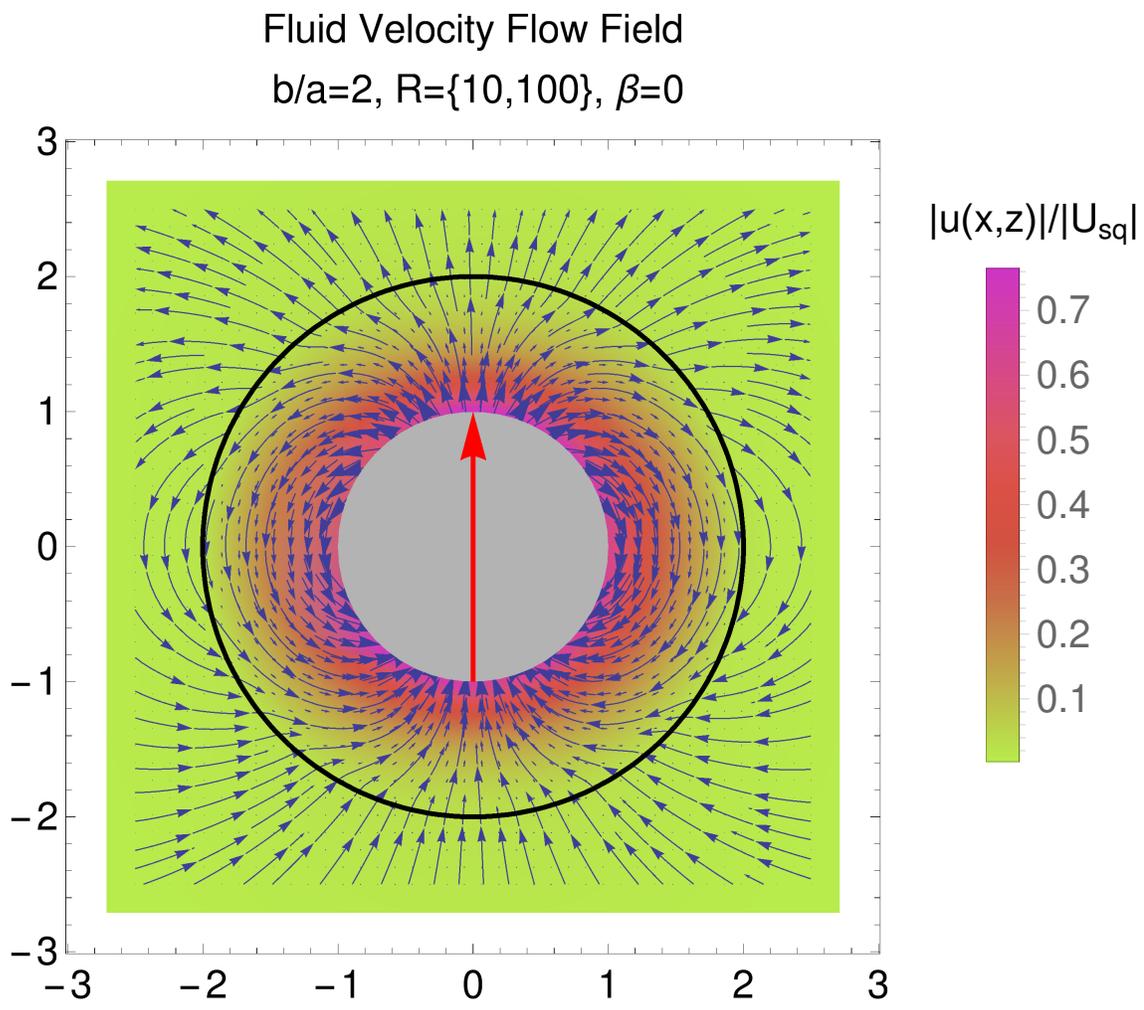


Fig. 5.15 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

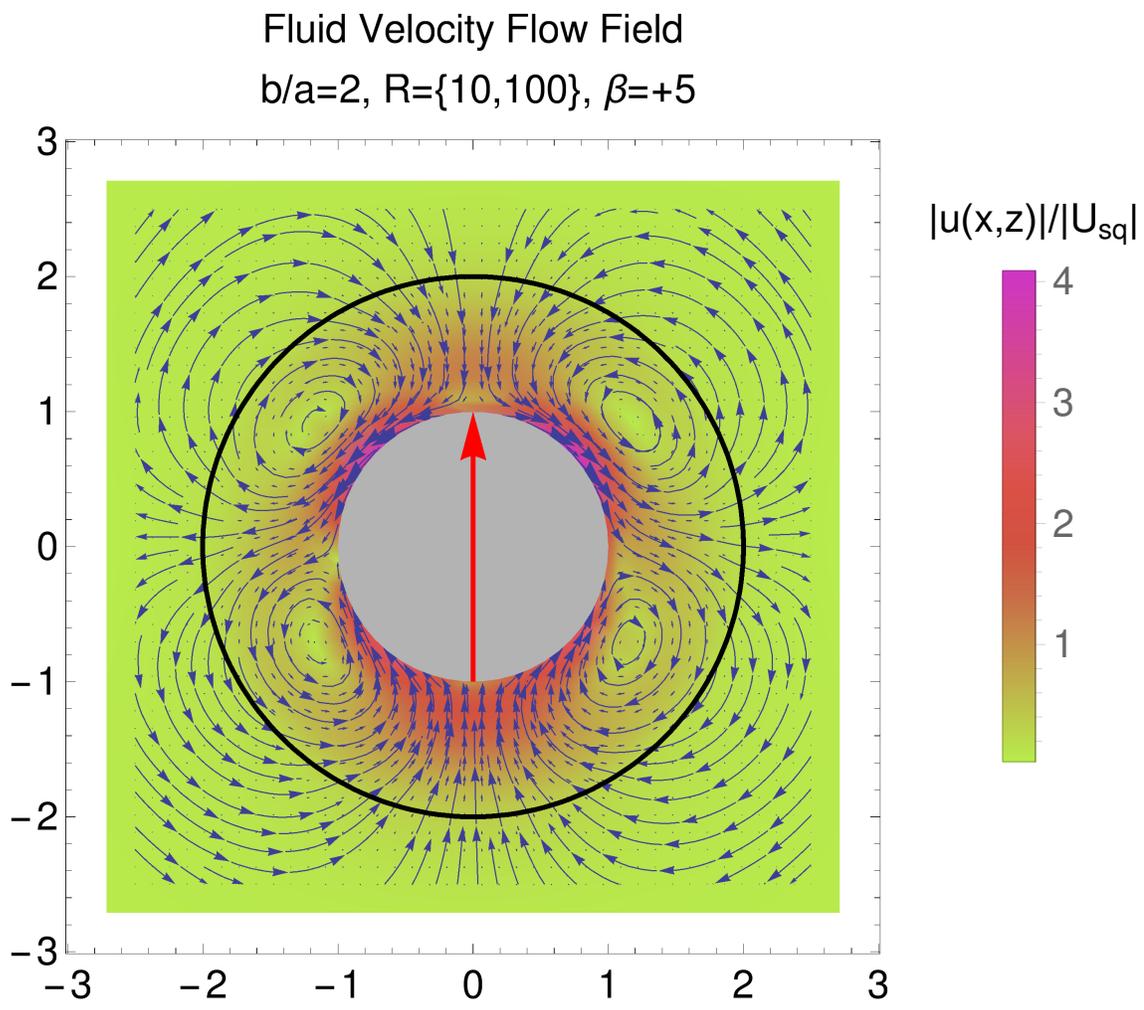


Fig. 5.16 The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

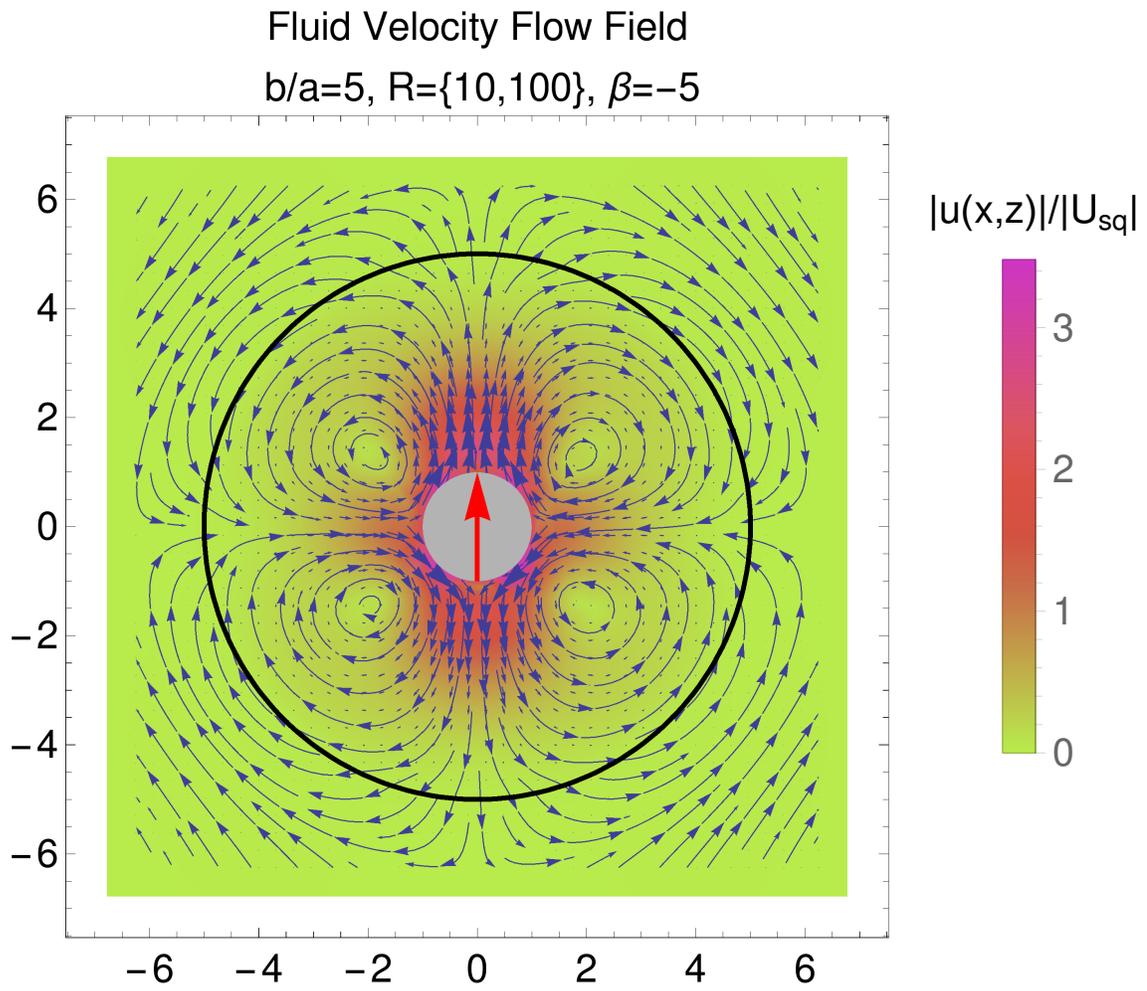


Fig. 5.17 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

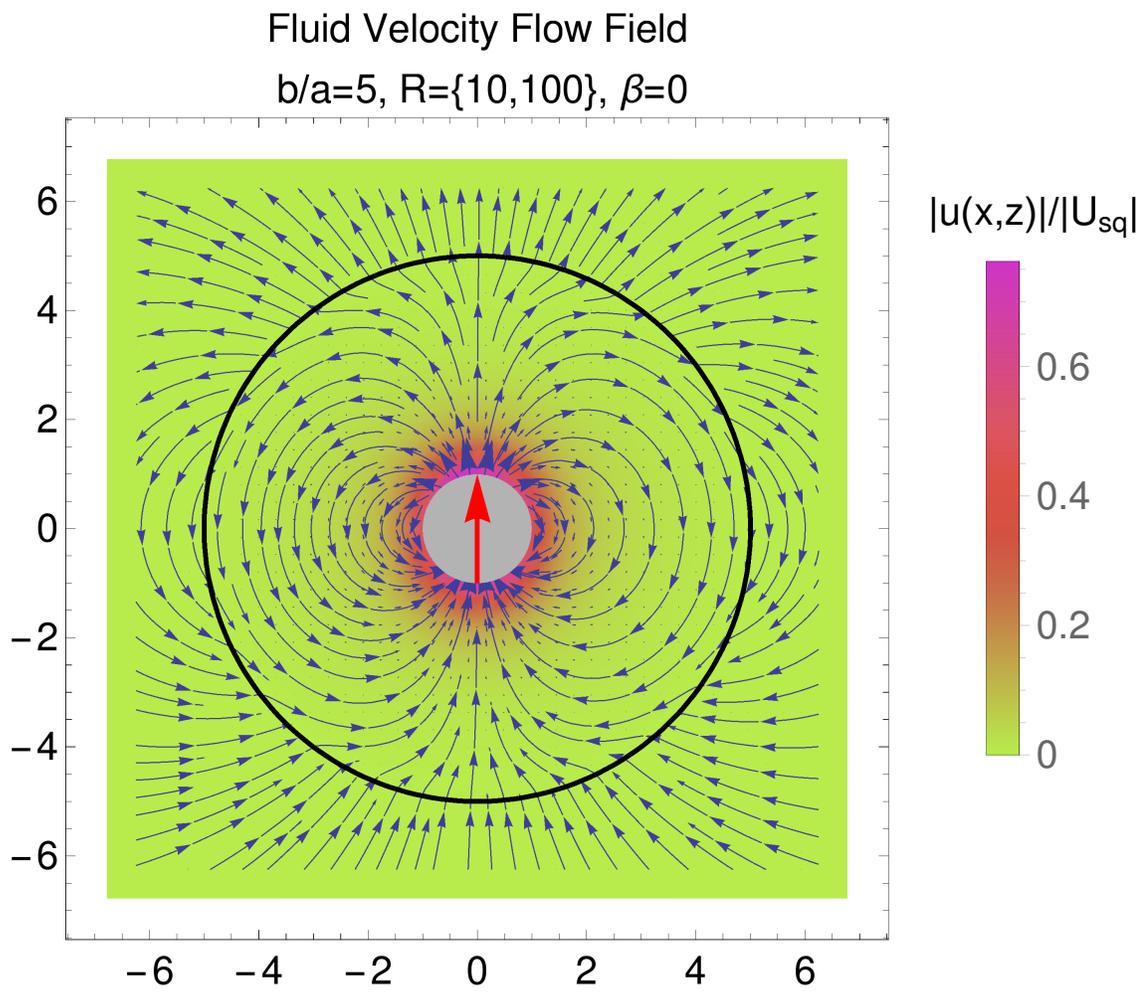


Fig. 5.18 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

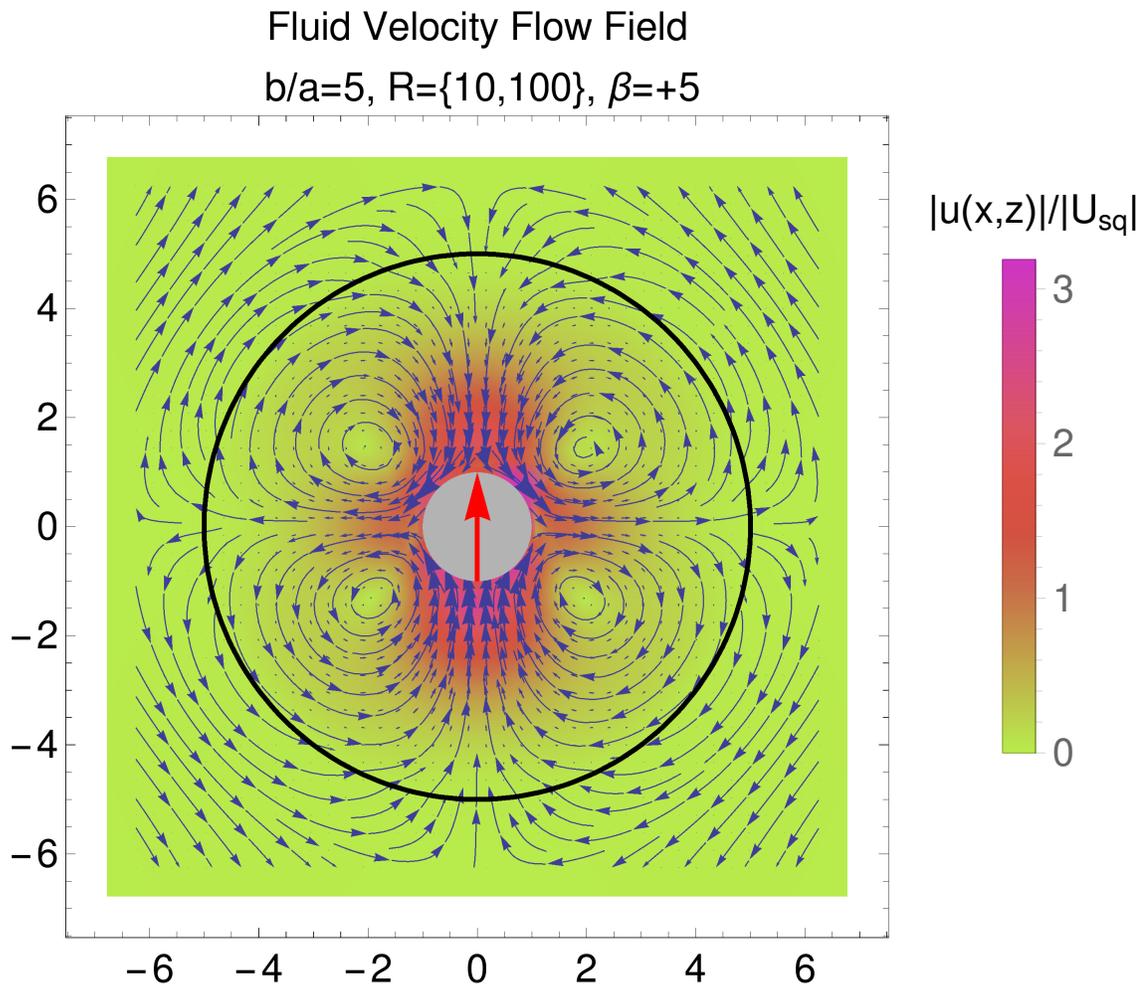


Fig. 5.19 The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

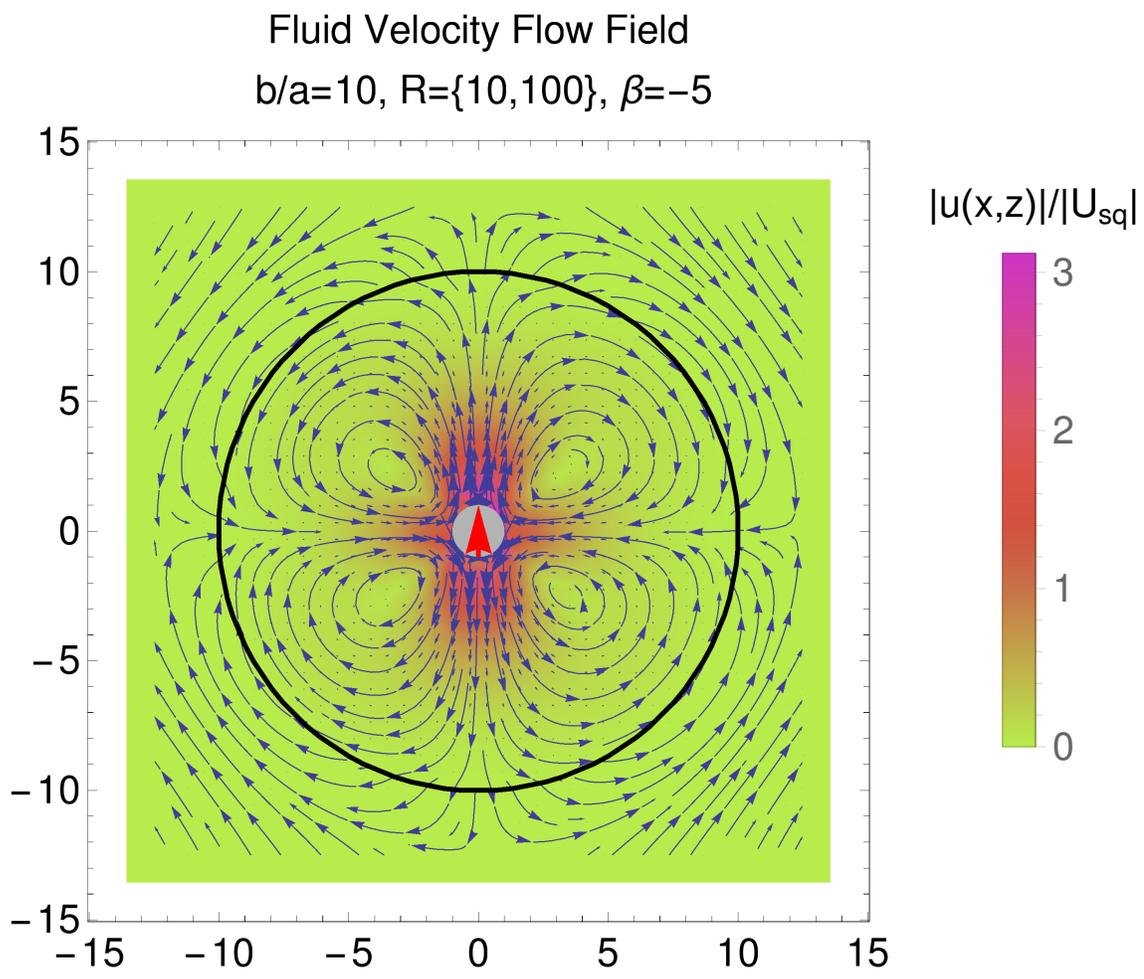


Fig. 5.20 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

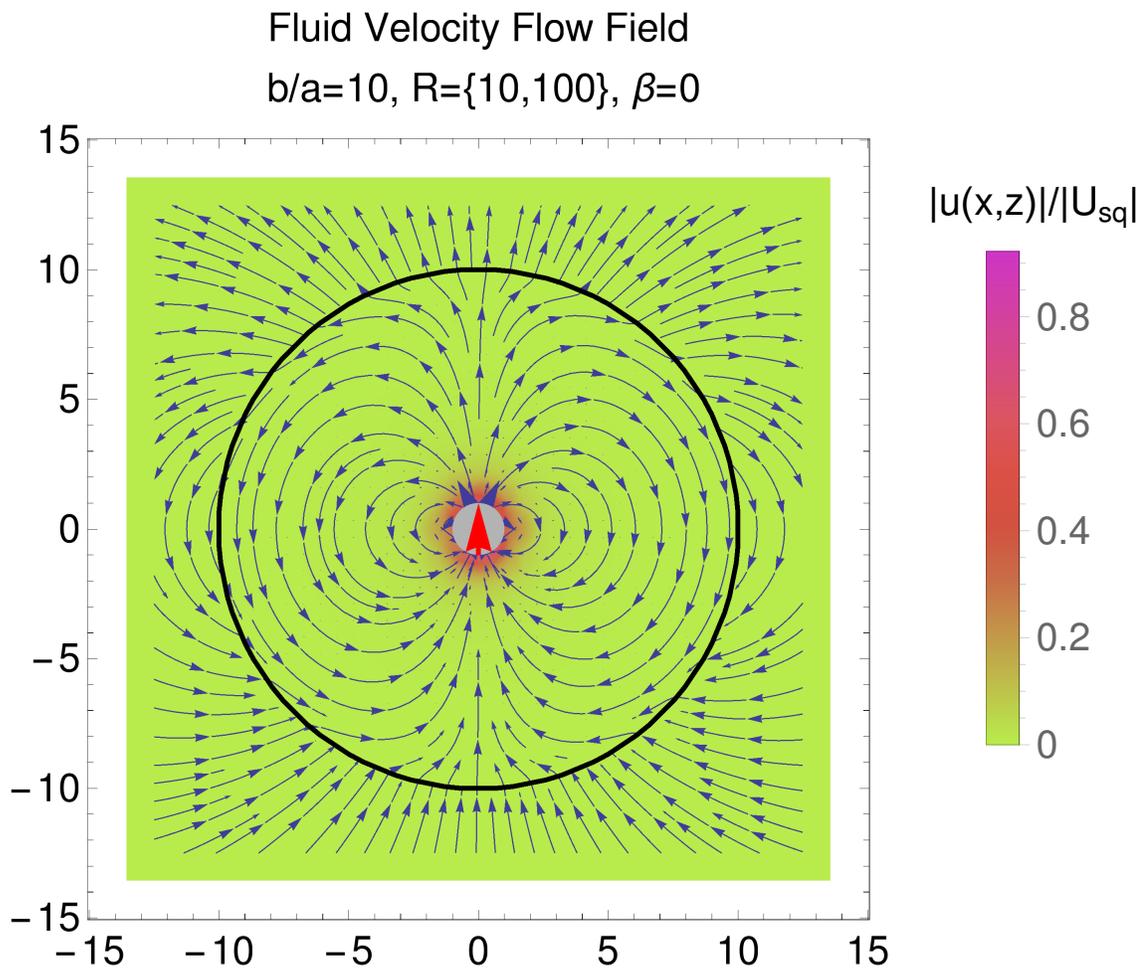


Fig. 5.21 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{10, 10, 100\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\parallel} promotes flow passing through the membrane in the tangential direction.

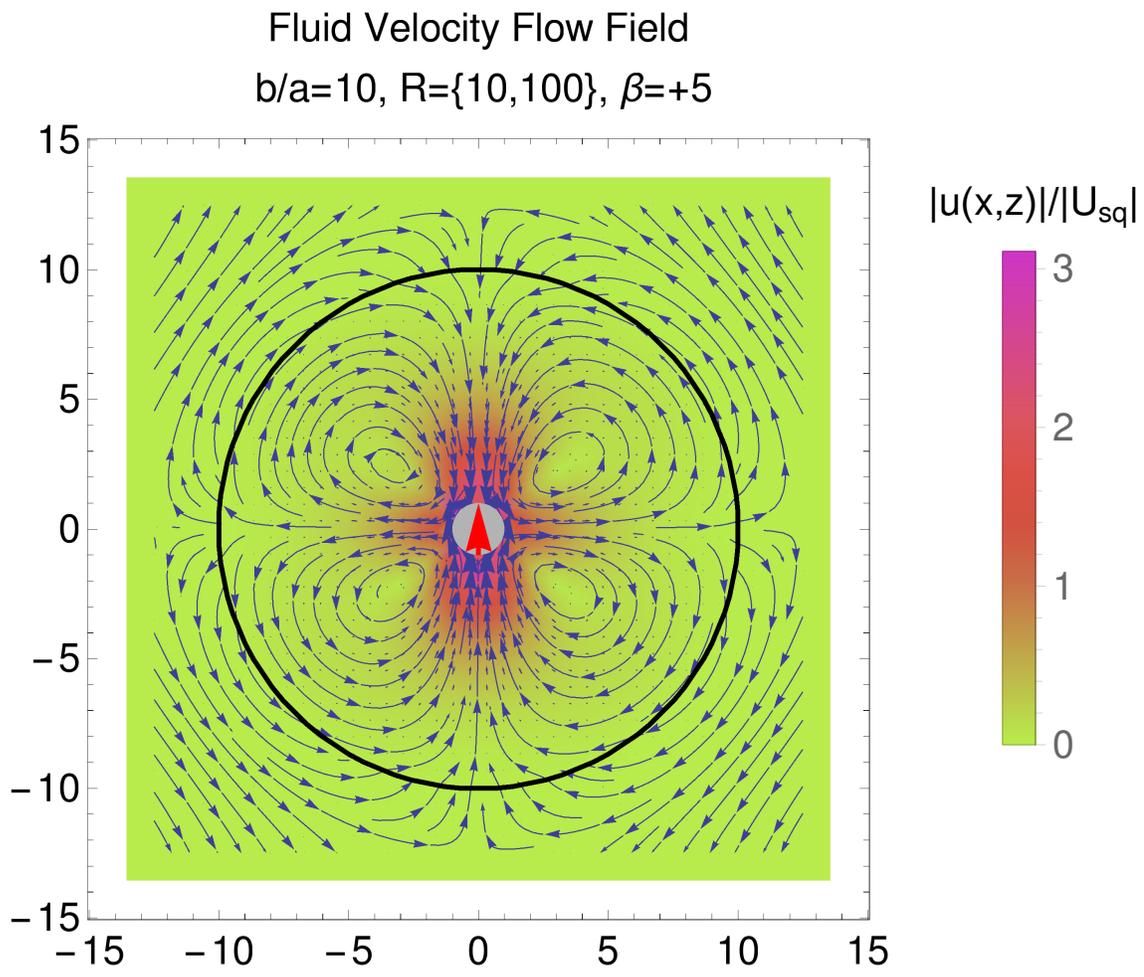


Fig. 5.22 The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

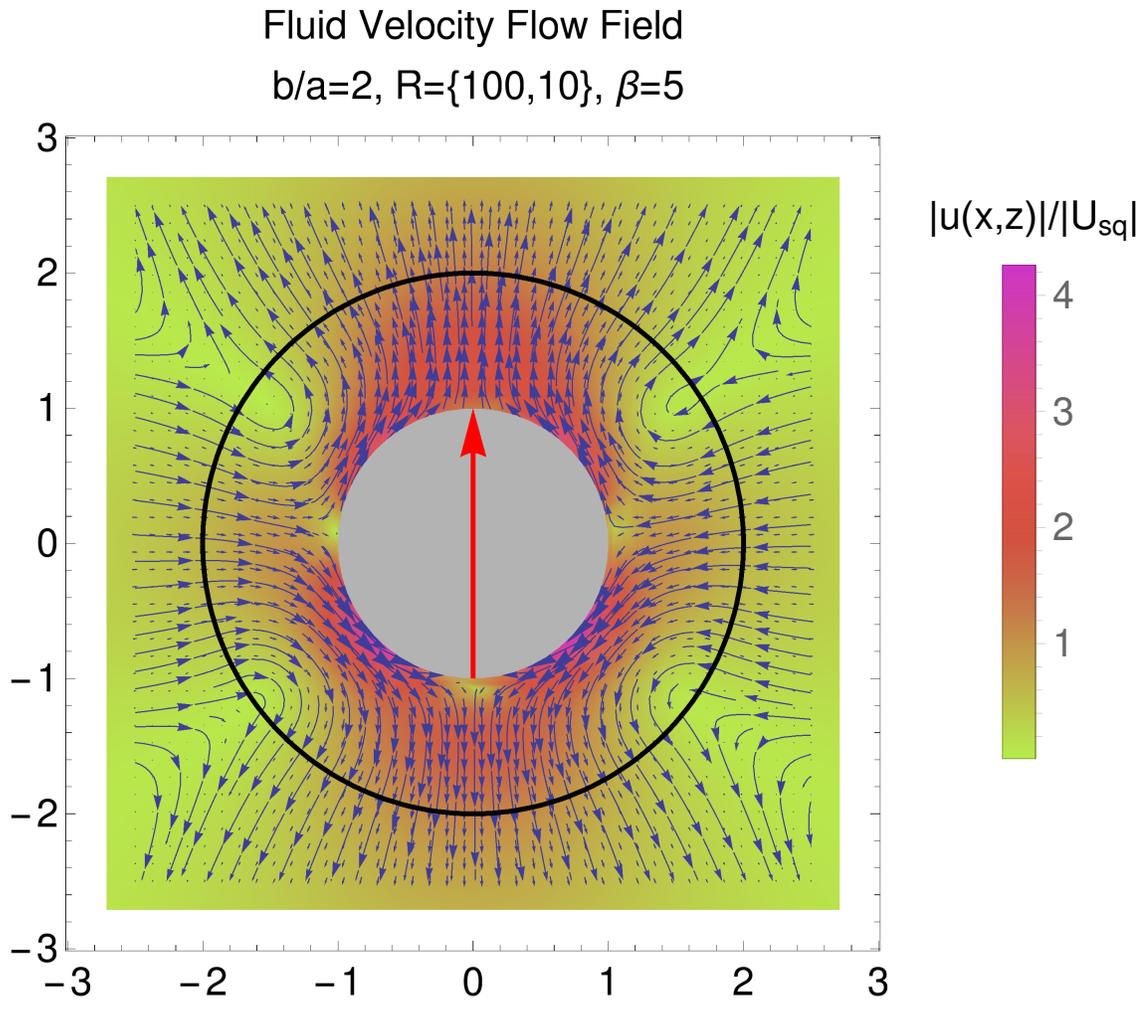


Fig. 5.23 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

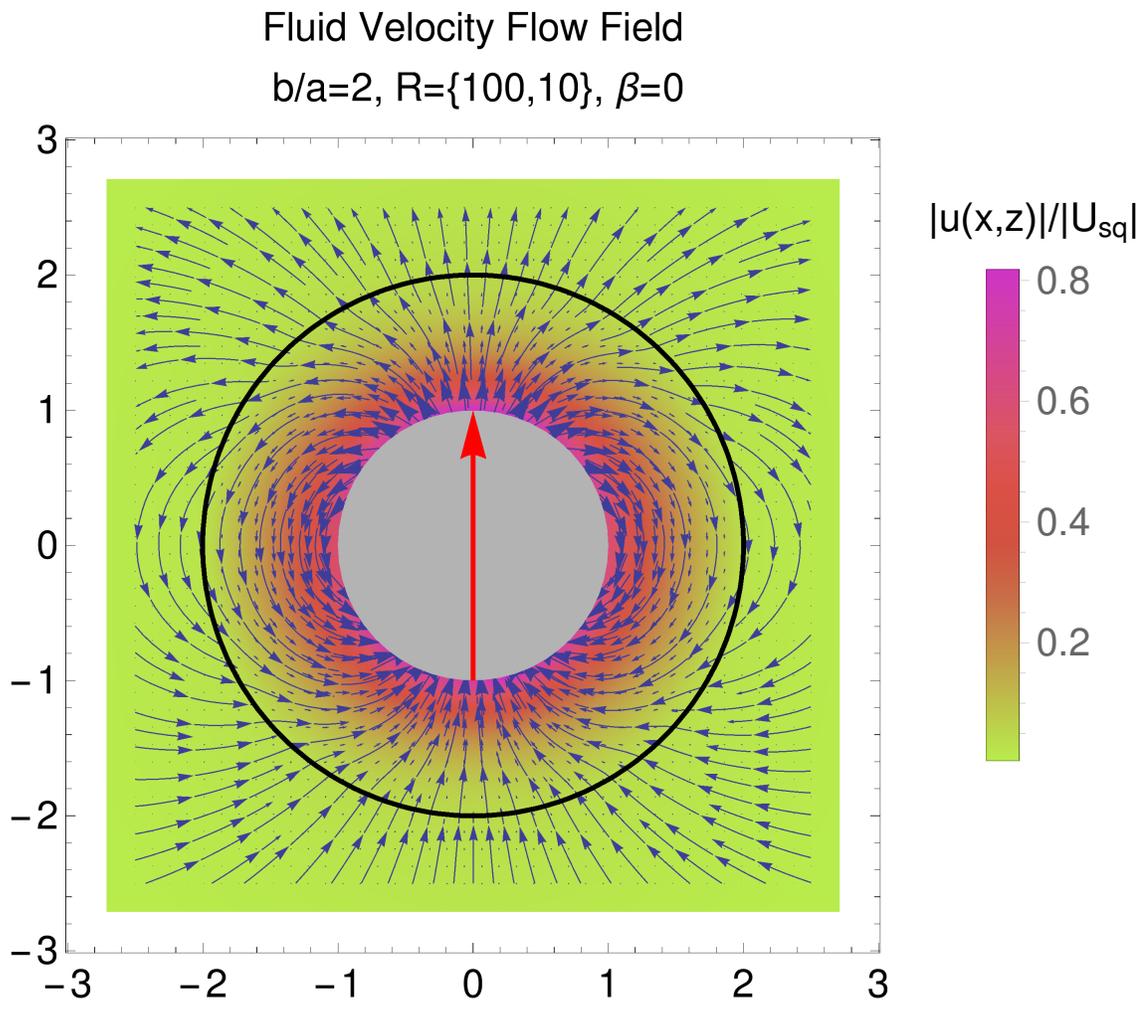


Fig. 5.24 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 2$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

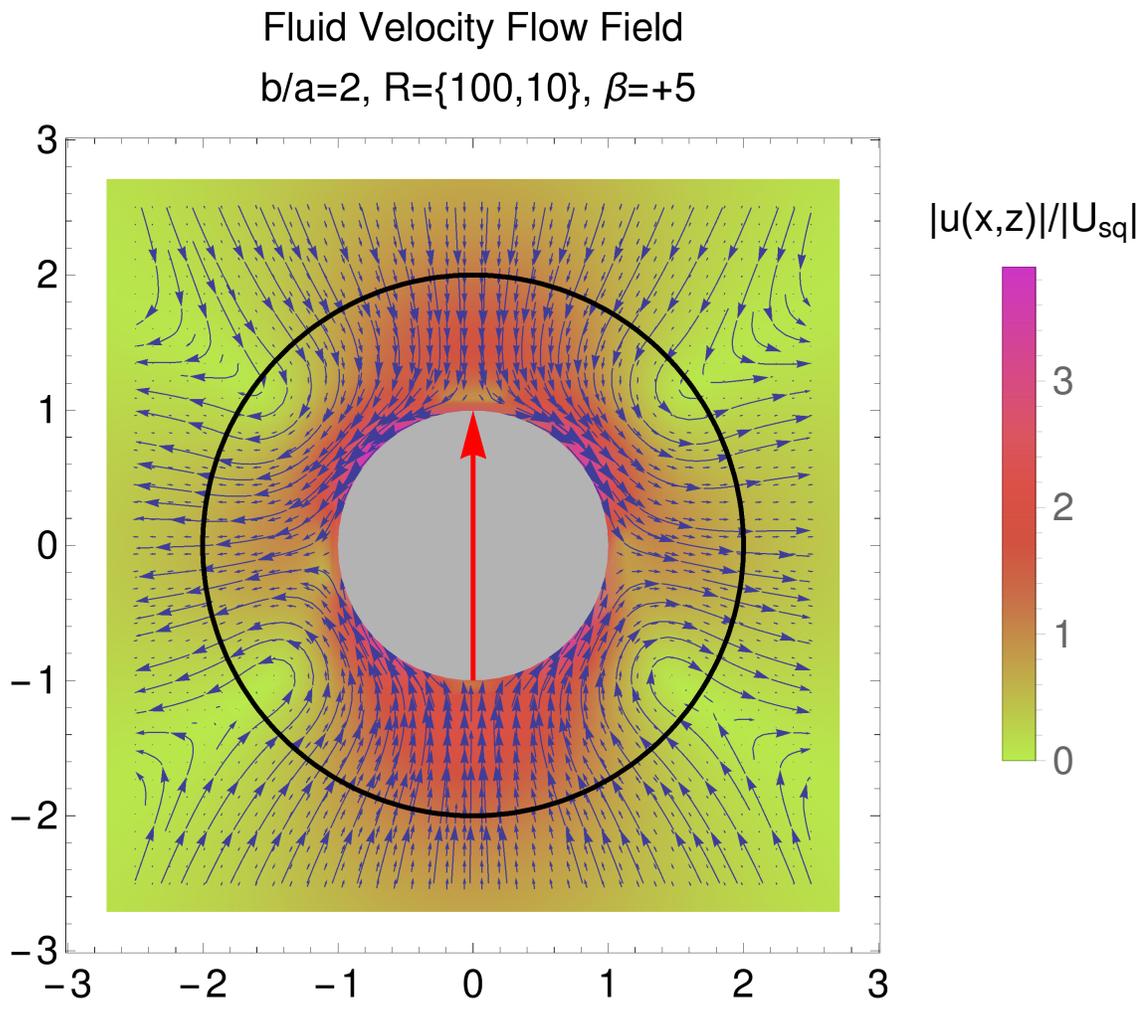


Fig. 5.25 The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

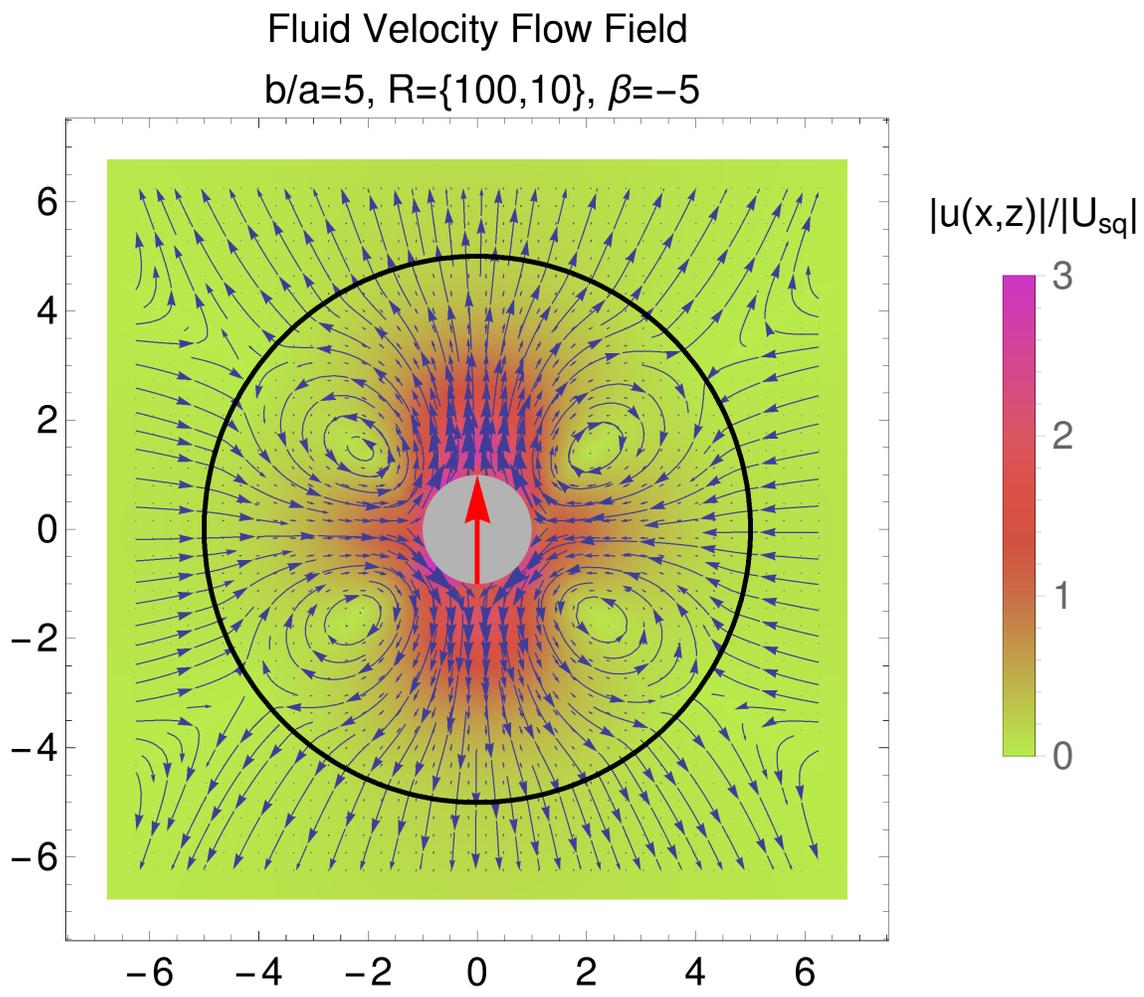


Fig. 5.26 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

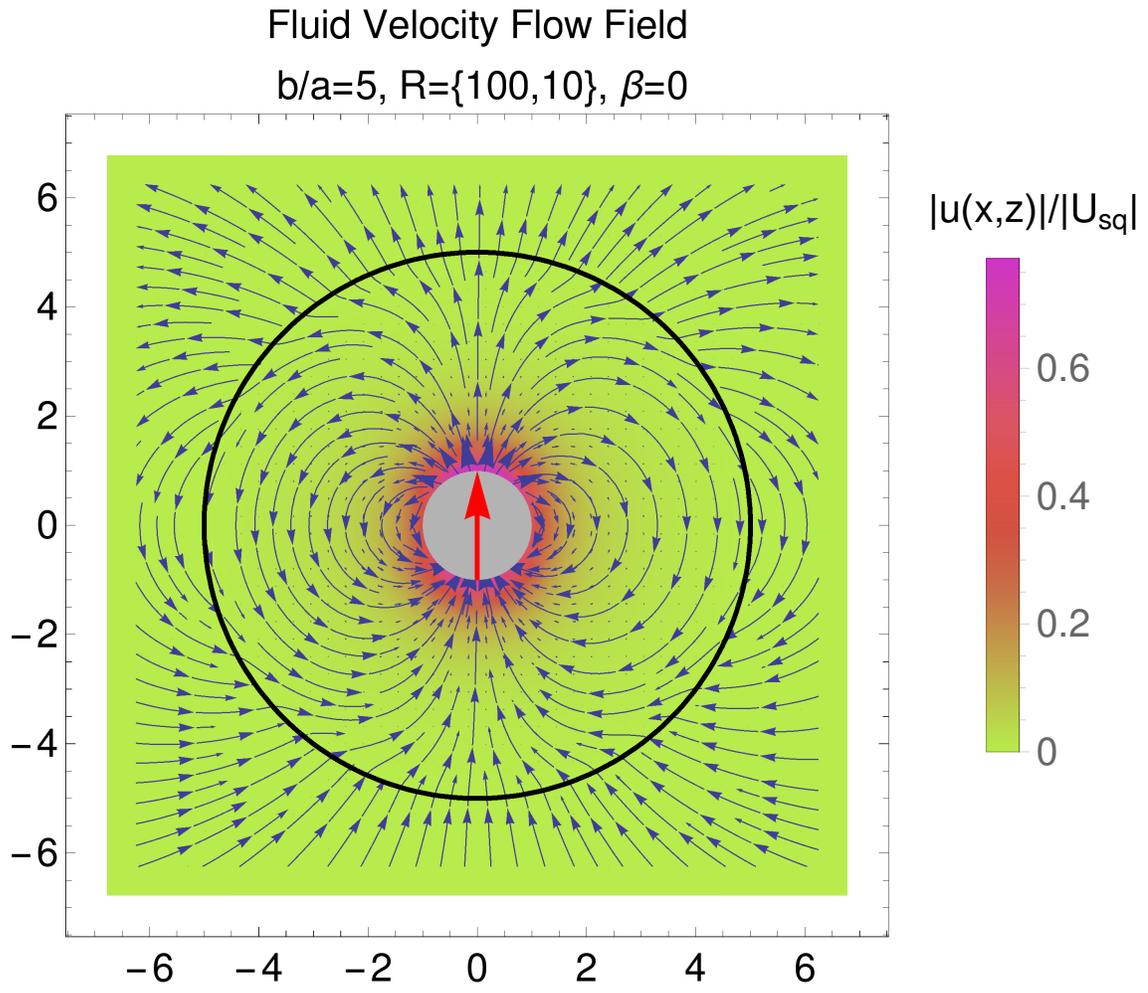


Fig. 5.27 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 5$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

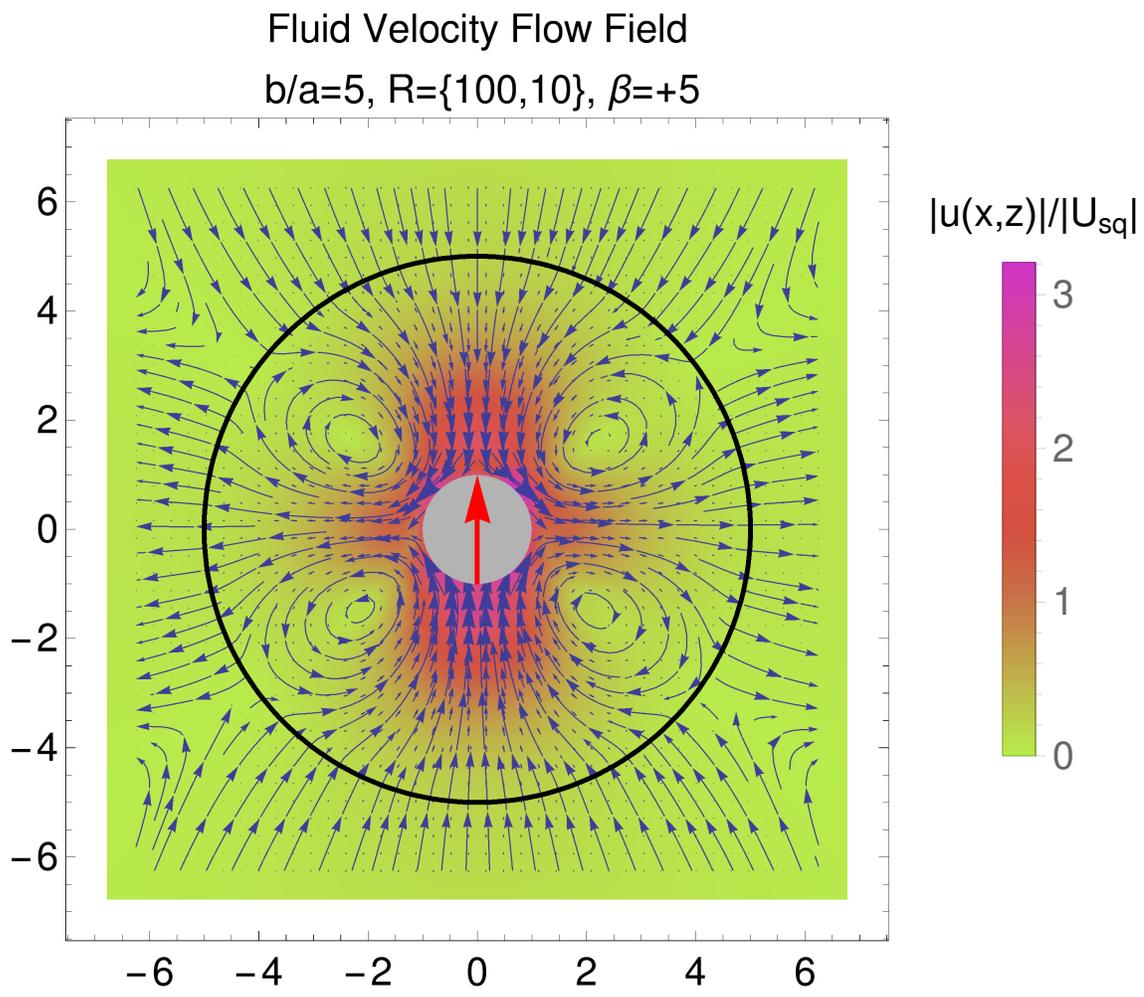


Fig. 5.28 The velocity field of a $\beta = -5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

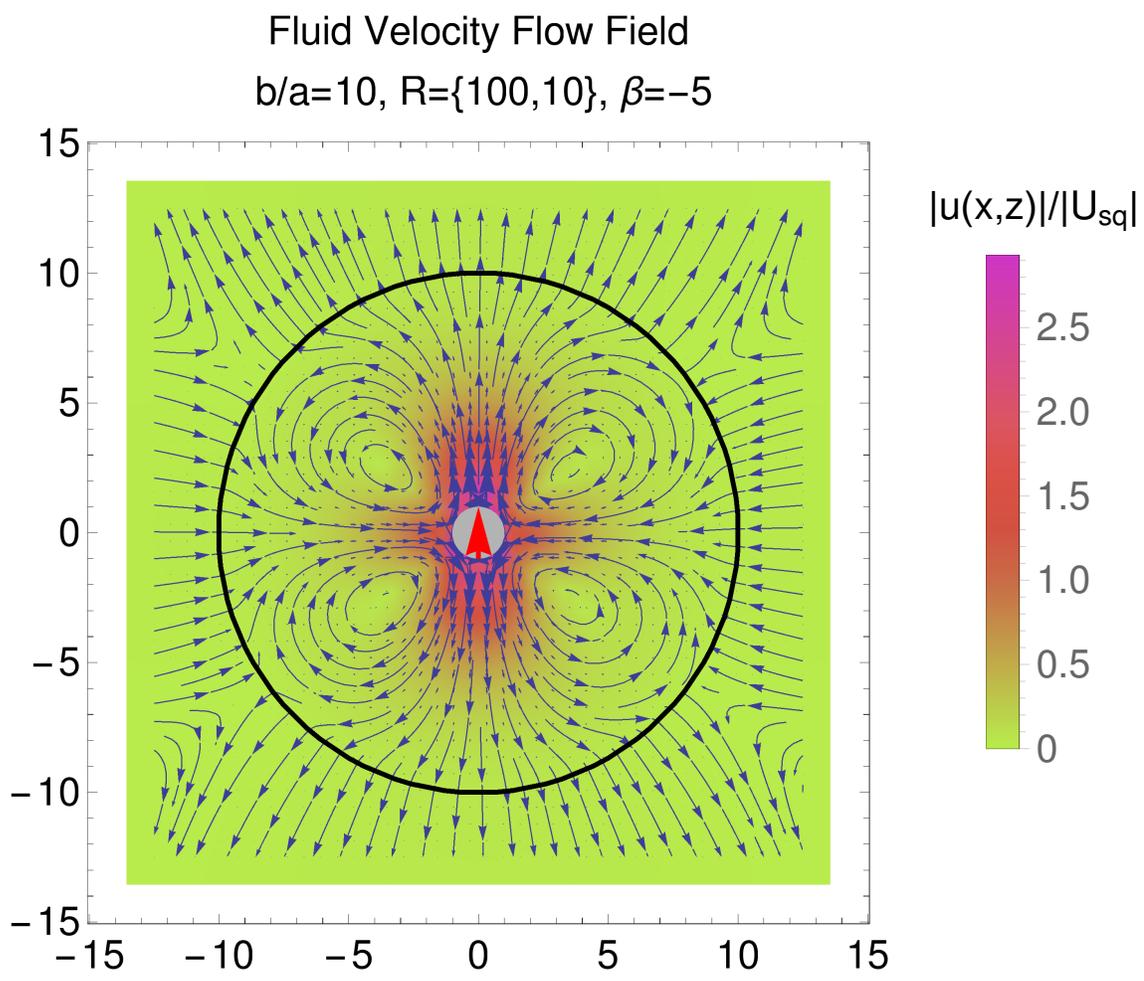


Fig. 5.29 The velocity field of a $\beta = 0$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

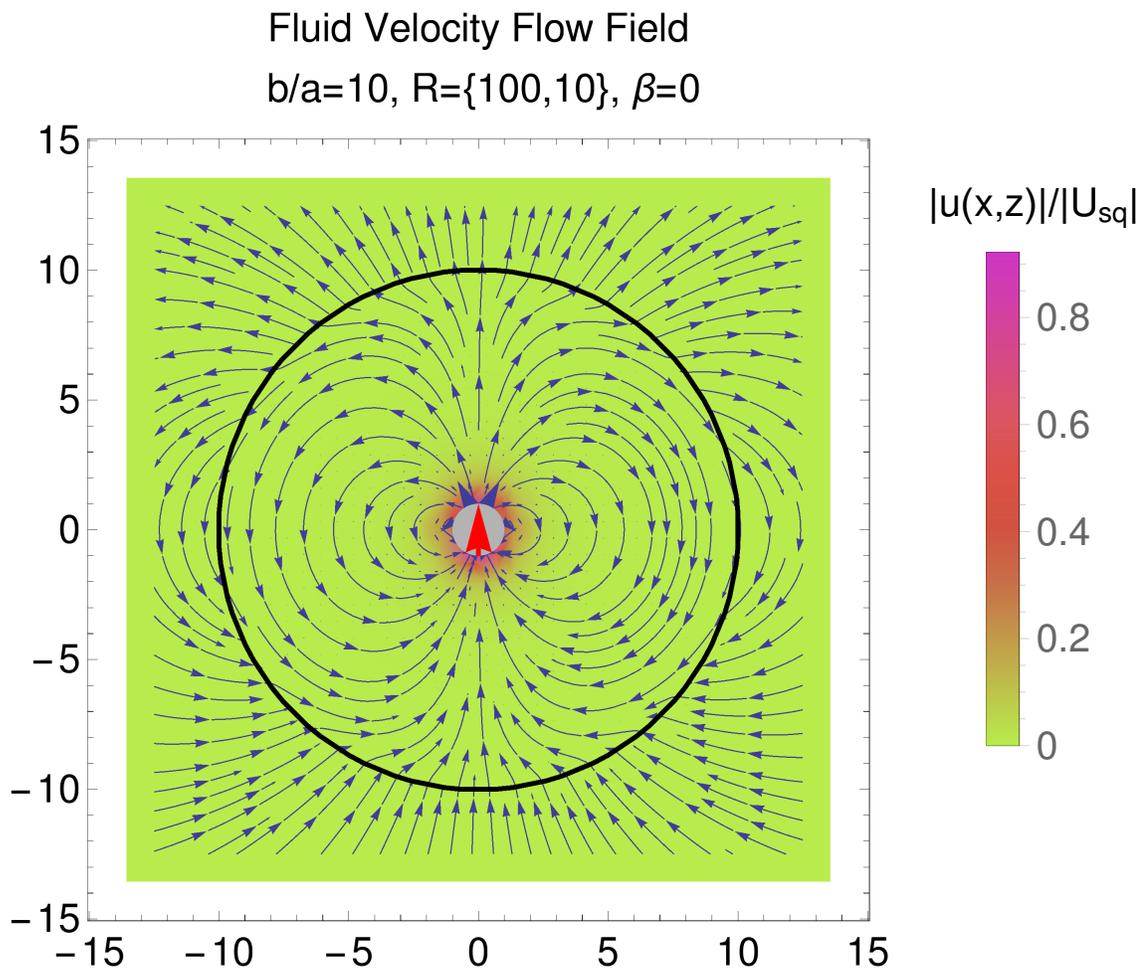


Fig. 5.30 The velocity field of a $\beta = +5$ squirmer is shown under the parameterization $R = \{100, 100, 10\}$ and for a container to particle size ratio $b/a = 10$. The weaker resistance in R_{\perp} promotes flow passing through the membrane in the normal direction.

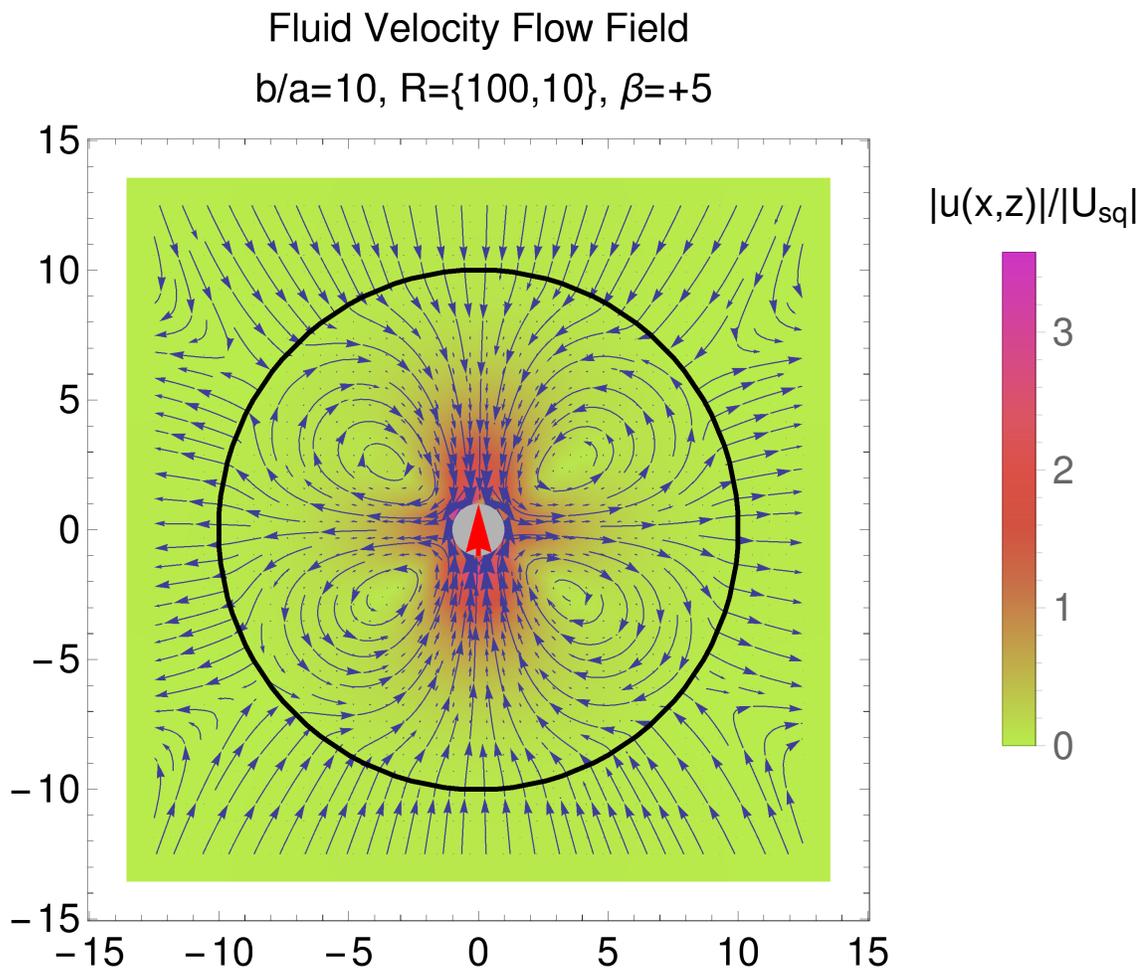


Fig. 5.31 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized squirmer’s translational velocity, $U_z^{sq}/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer’s translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model.

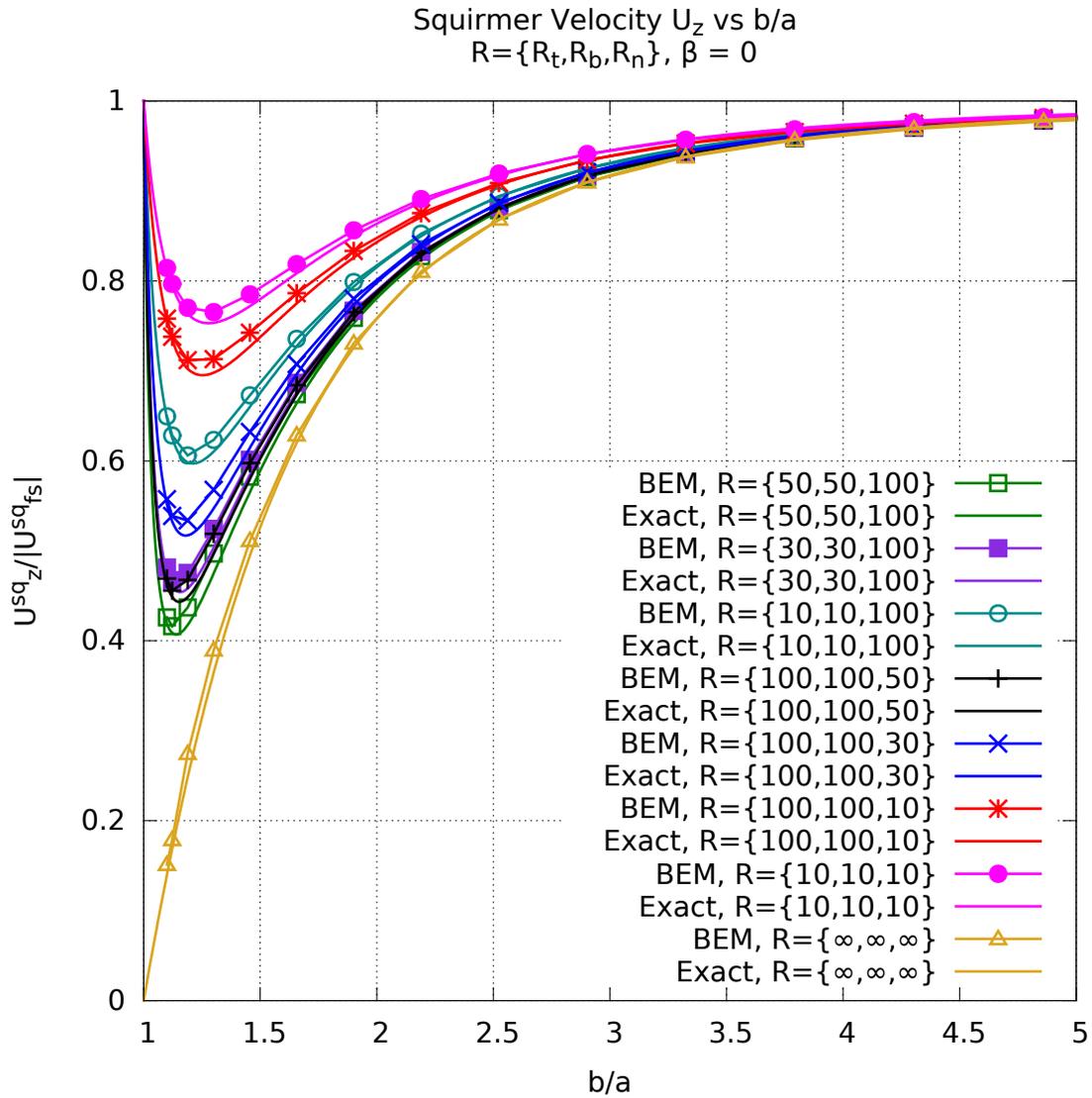


Fig. 5.32 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized squirmer’s translational velocity, $U_z^{sq}/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow is made to dominating (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer’s translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model.

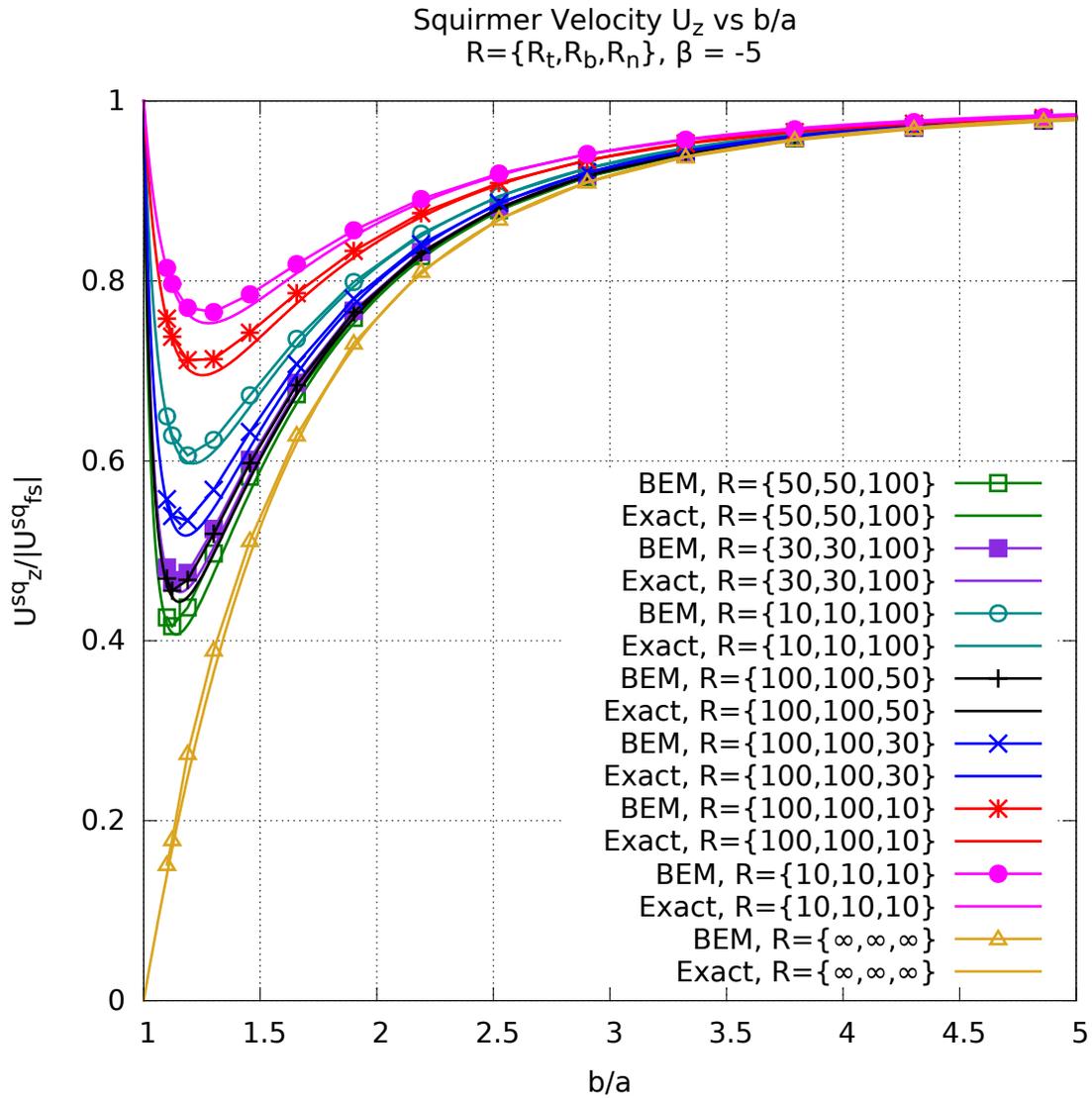


Fig. 5.33 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized container’s translational velocity, $U_z^m/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer’s translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model.

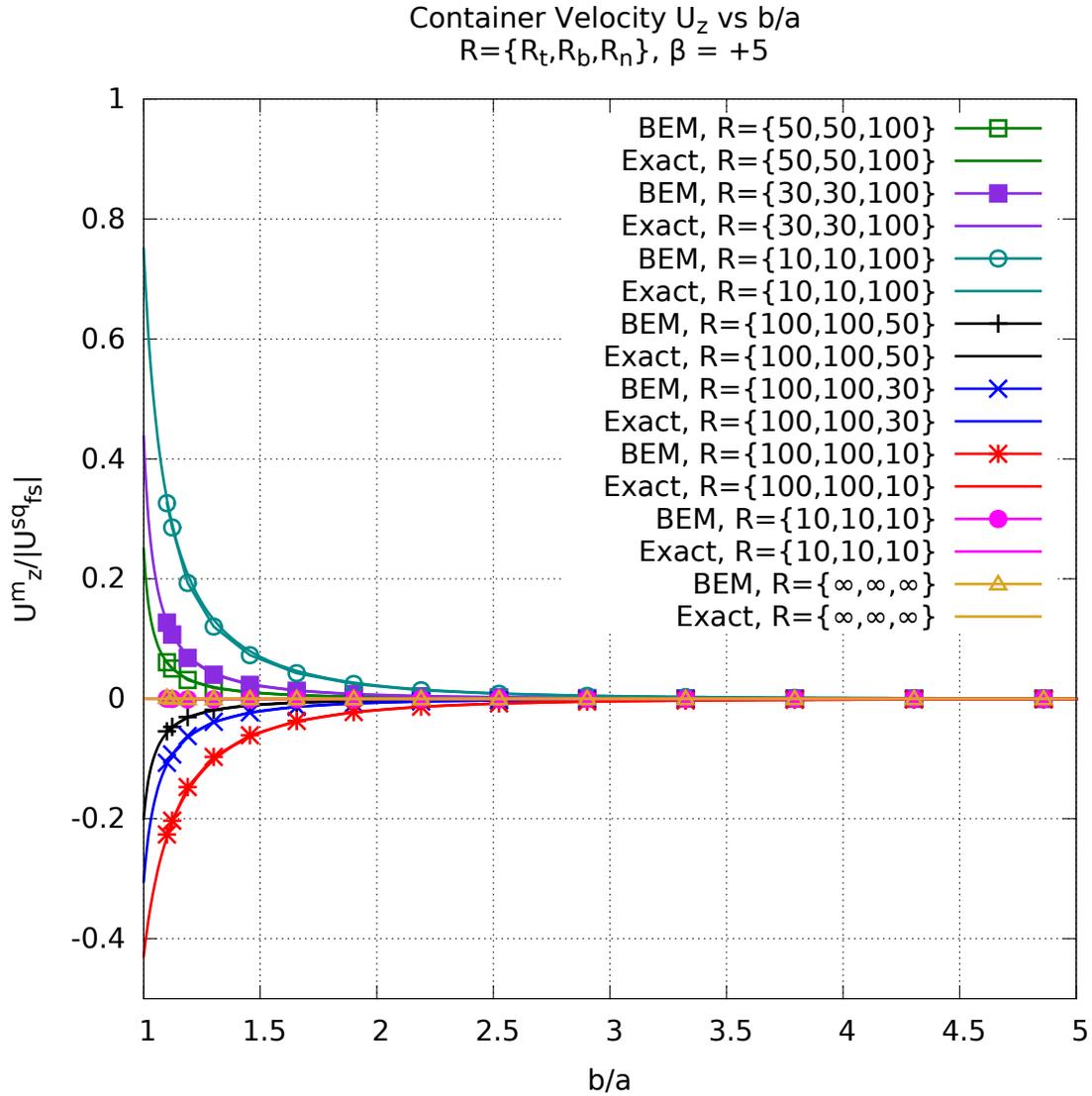


Fig. 5.34 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized container’s translational velocity, $U_z^m/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer’s translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model.

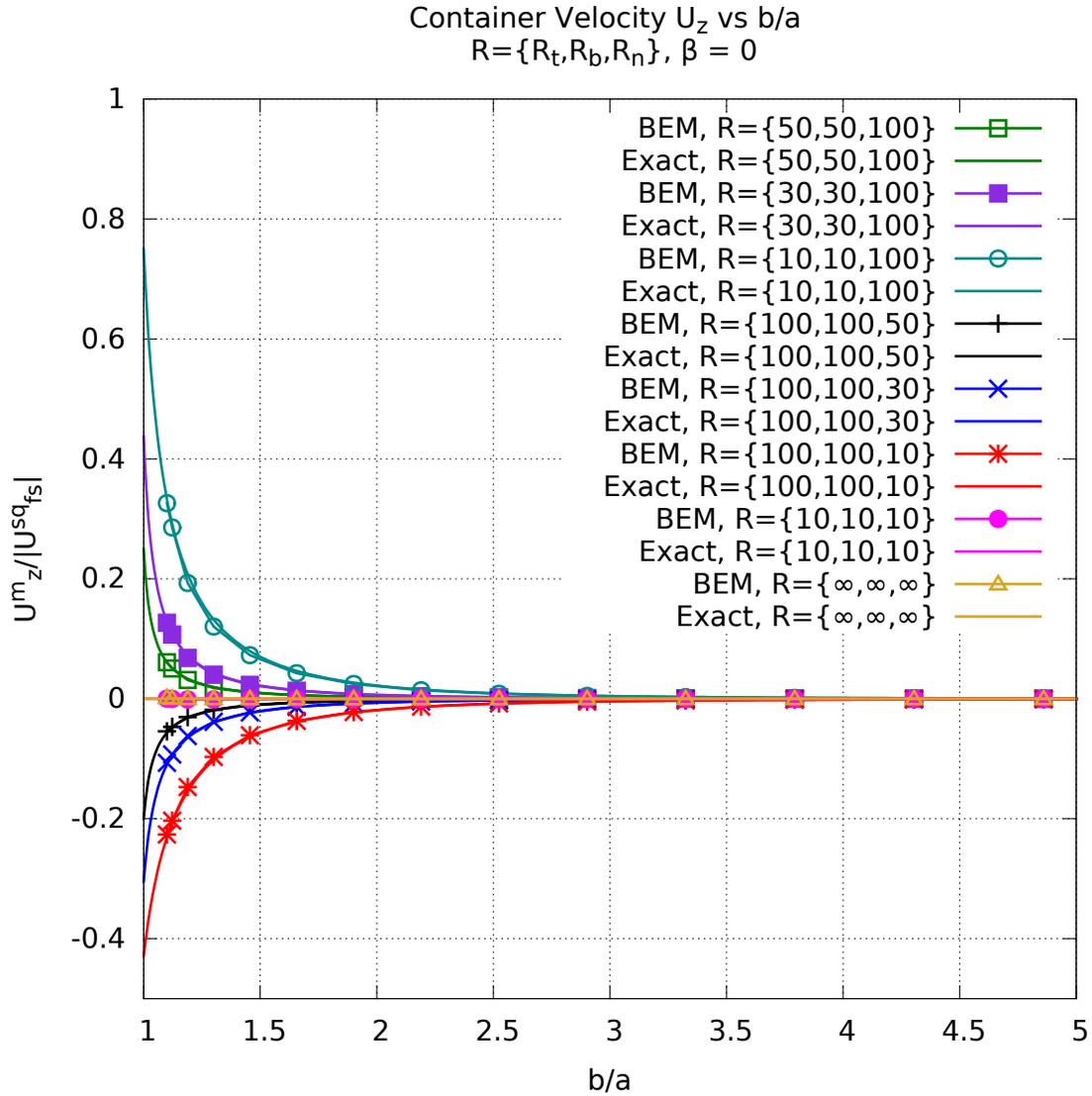


Fig. 5.35 This plot shows a comparison between the analytical, *Exact*, and numerical, *BEM*, models for the normalized container’s translational velocity, $U_z^m/(2/3B_1)$, versus the container/squirmer size ratio b/a . This comparison is made for various sets of permeability resistances where tangential or normal flow across the container is made to dominate (i.e. has a lower resistance). The correspondence between these resistances is given by $R = \{R_t, R_b, R_n\} \equiv \{R_{\parallel}, R_{\parallel}, R_{\perp}\}$. The squirmer’s translational velocity is seen to have a minimum at a particular value of b/a . This minimum is reproduced accurately by the Galerkin BEM model.

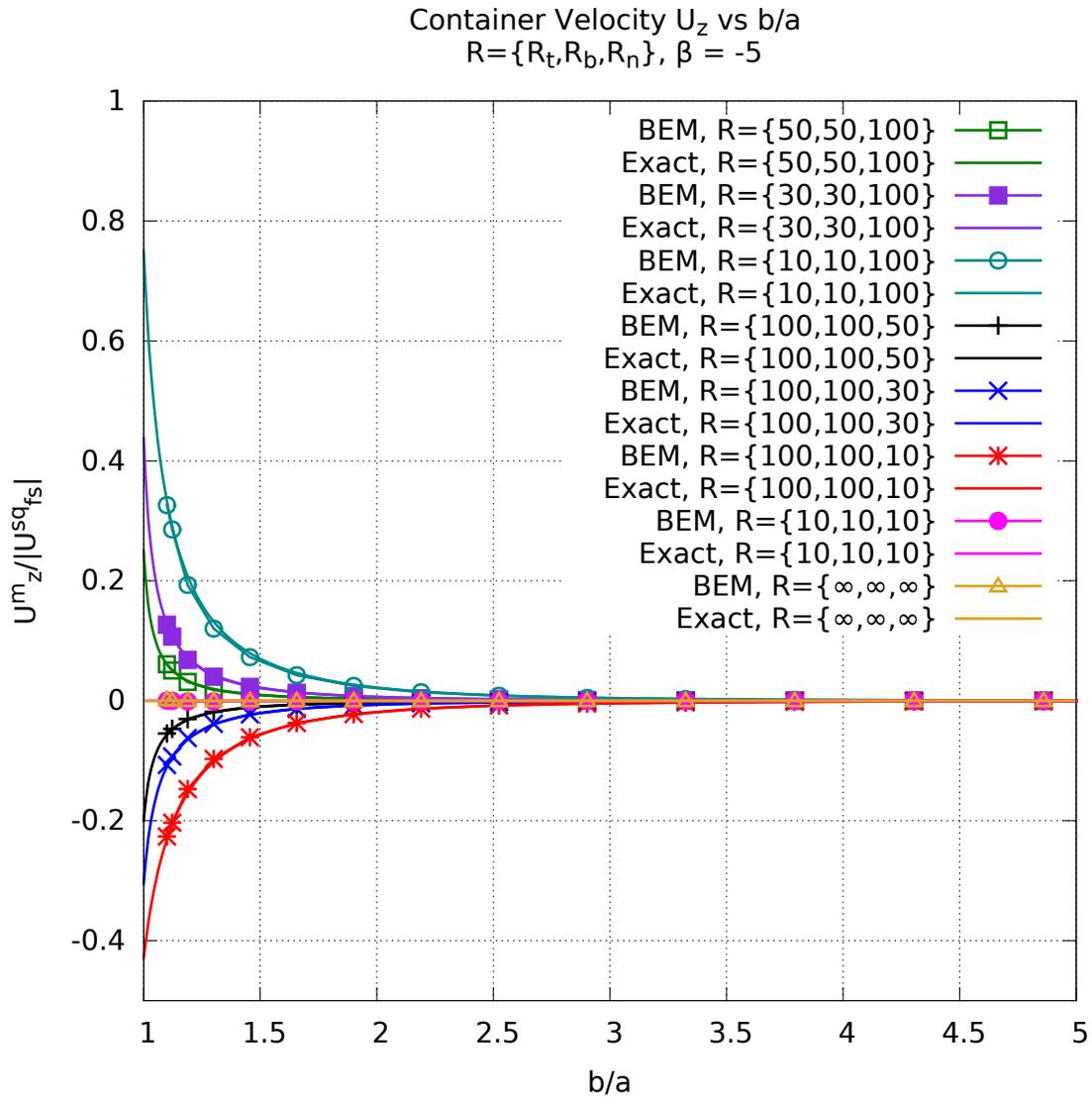


Fig. 5.36 In this plot all regular Cartesian (query) grid points are circled in red. Grid points that are inside (outside) of the region defined by $r < r_{cp}^{max}$ are shown in green (blue). The mobility solution is solved for a squirmer placed at all green points (289 query points and 96 mobility solves). A thick black circle shows the container boundary at $r = b$. An inner red circle shows where $r = b - a\delta$. The pink or fuchsia colored circle shows where $r = r_{cp}^{max}$. The inner gray circle is sized according to $r = a$ giving a sense of the squirmer's size.

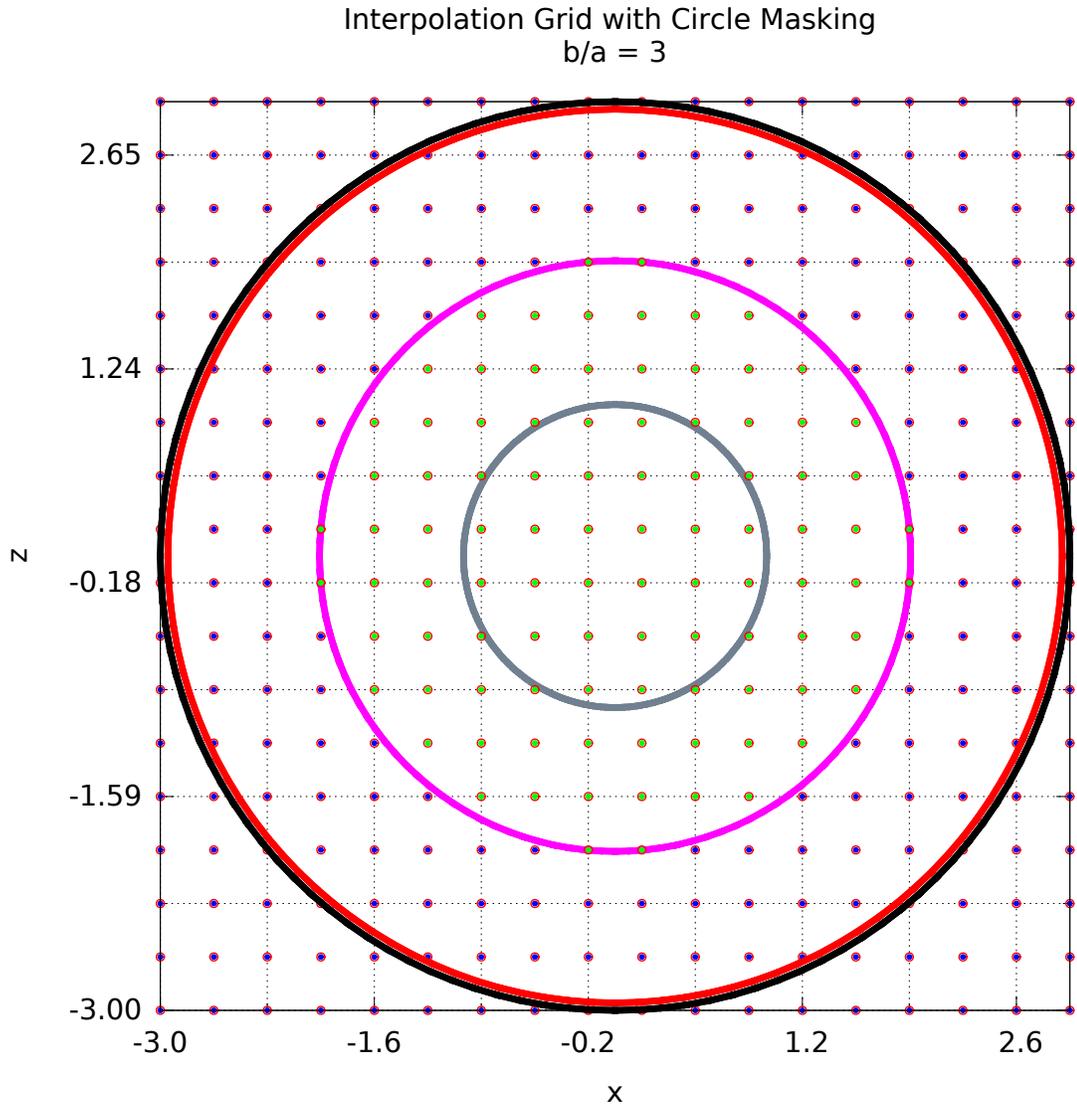


Fig. 5.37 In this plot all regular Cartesian (query) grid points are circled in red. Grid points that are inside (outside) of the region defined by $r < r_{cp}^{max}$ are shown in green (blue). The mobility solution is solved for a squirmer placed at all green points (3249 query points and 2040 mobility solves). A thick black circle shows the container boundary at $r = b$. An inner red circle shows where $r = b - a\delta$. The pink or fuchsia colored circle shows where $r = r_{cp}^{max}$. The inner gray circle is sized according to $r = a$ giving a sense of the squirmer's size.

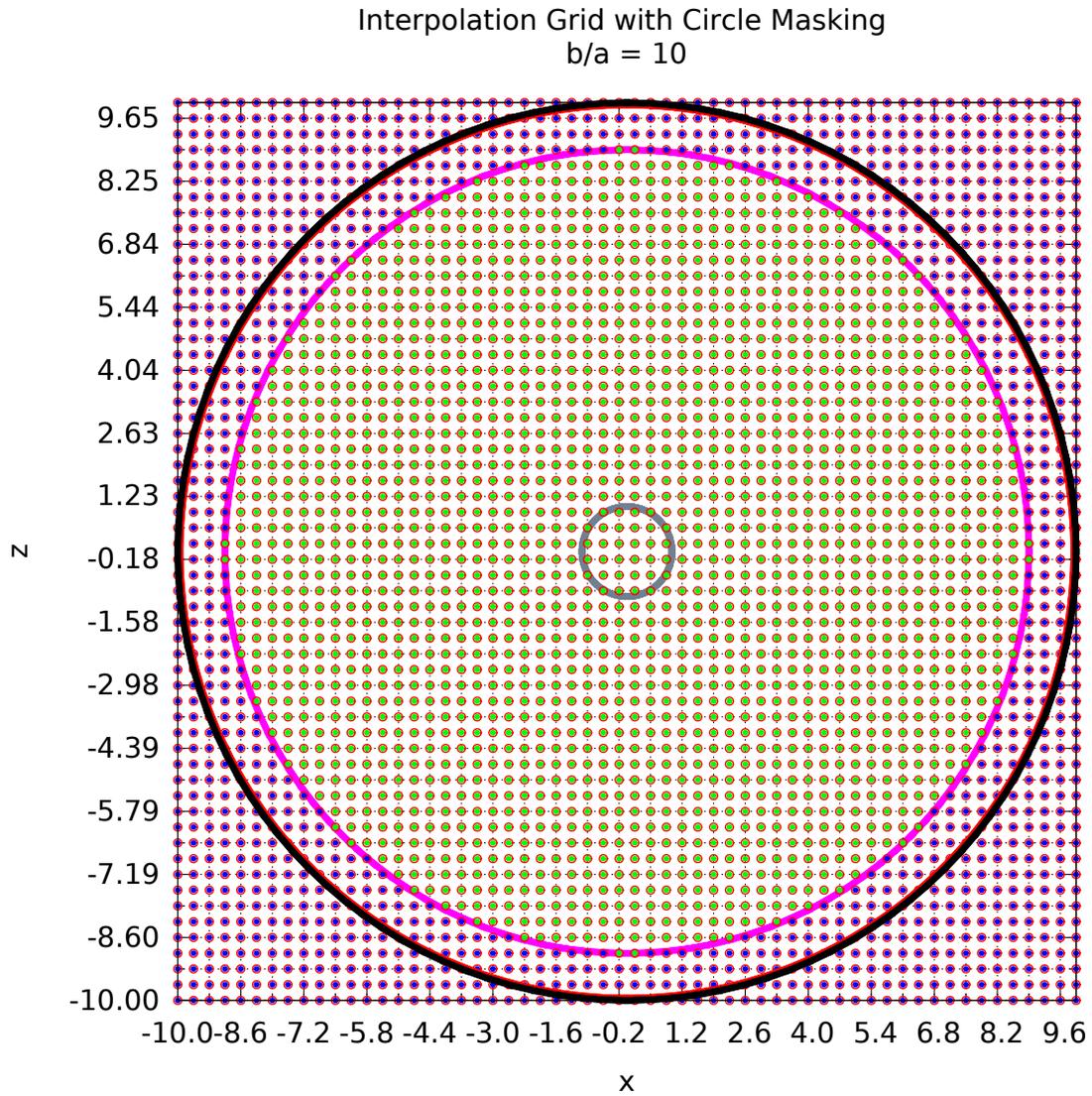


Fig. 5.38 This plot shows an up-sampled grid for $b/a = 3$ and $\kappa = 3$. Up-sampled grid points are shown in orange. Original grid points are shown in green (inside the circle mask defined by $r = r_{cp}^{max}$) and blue outside the circle mask.

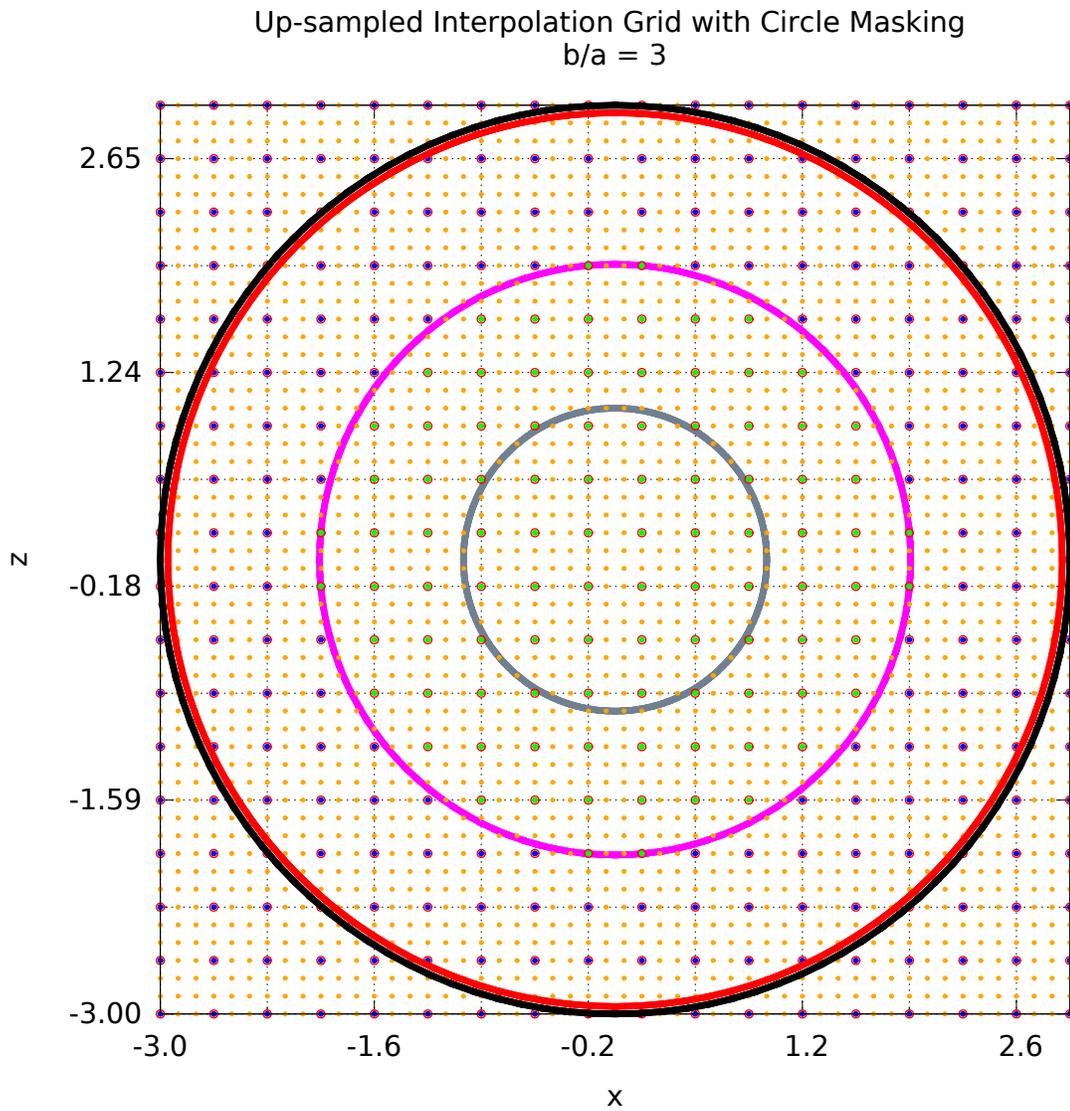


Fig. 5.39 This plot shows an up-sampled grid for $b/a = 5$ and $\kappa = 3$. Up-sampled grid points are shown in orange. Original grid points are shown in green (inside the circle mask defined by $r = r_{cp}^{max}$) and blue outside the circle mask.

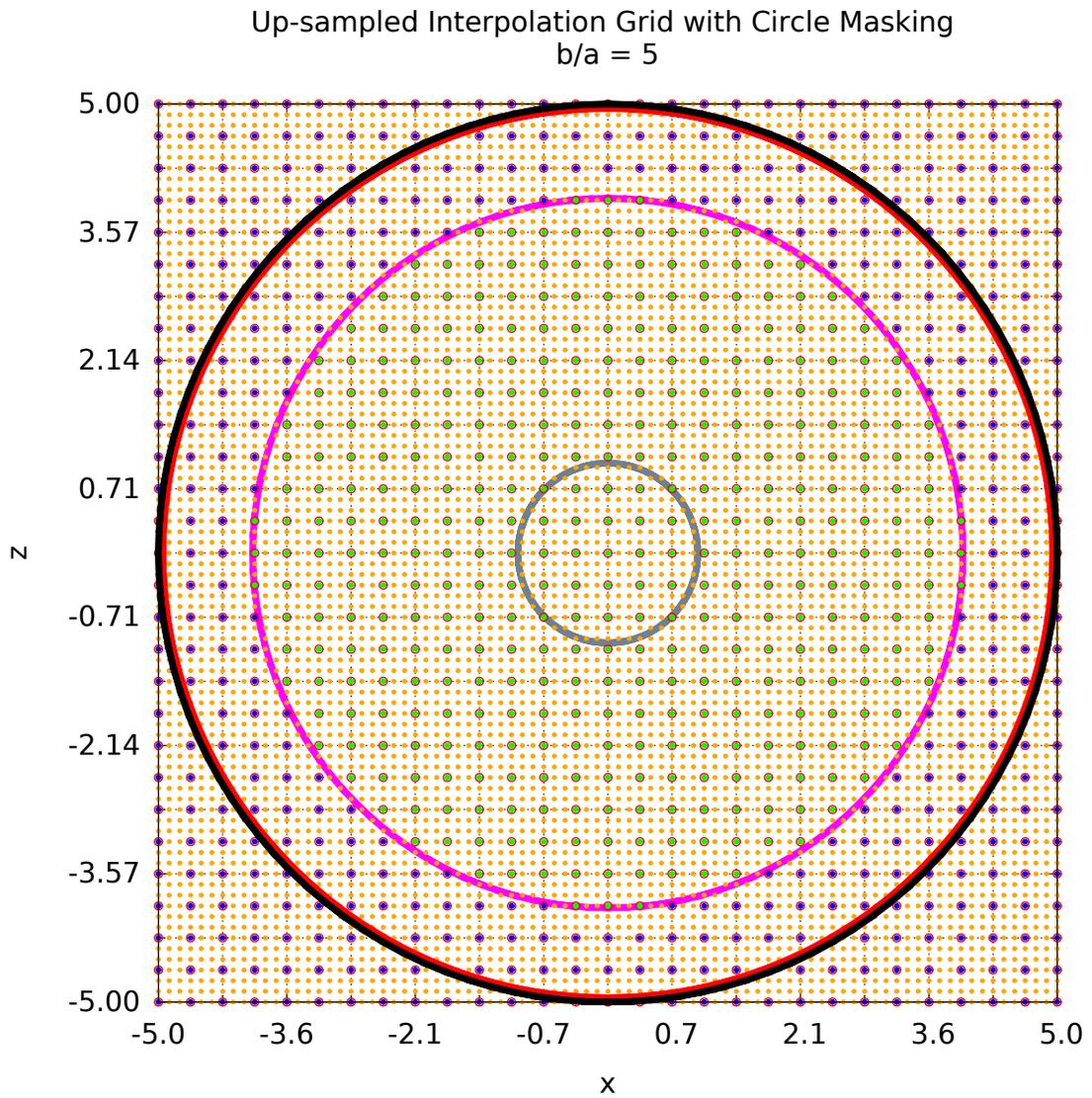


Fig. 5.40 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

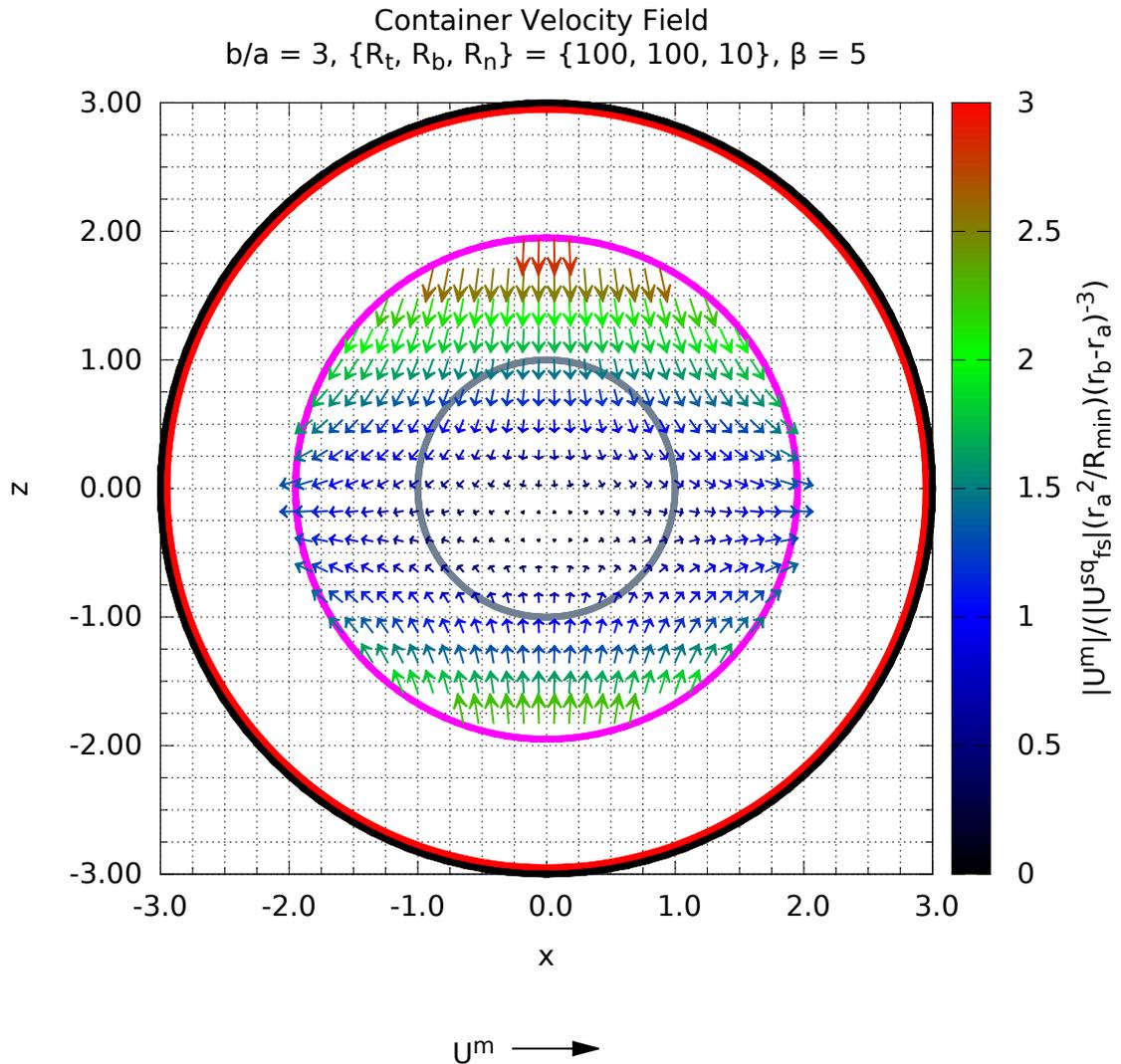


Fig. 5.41 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

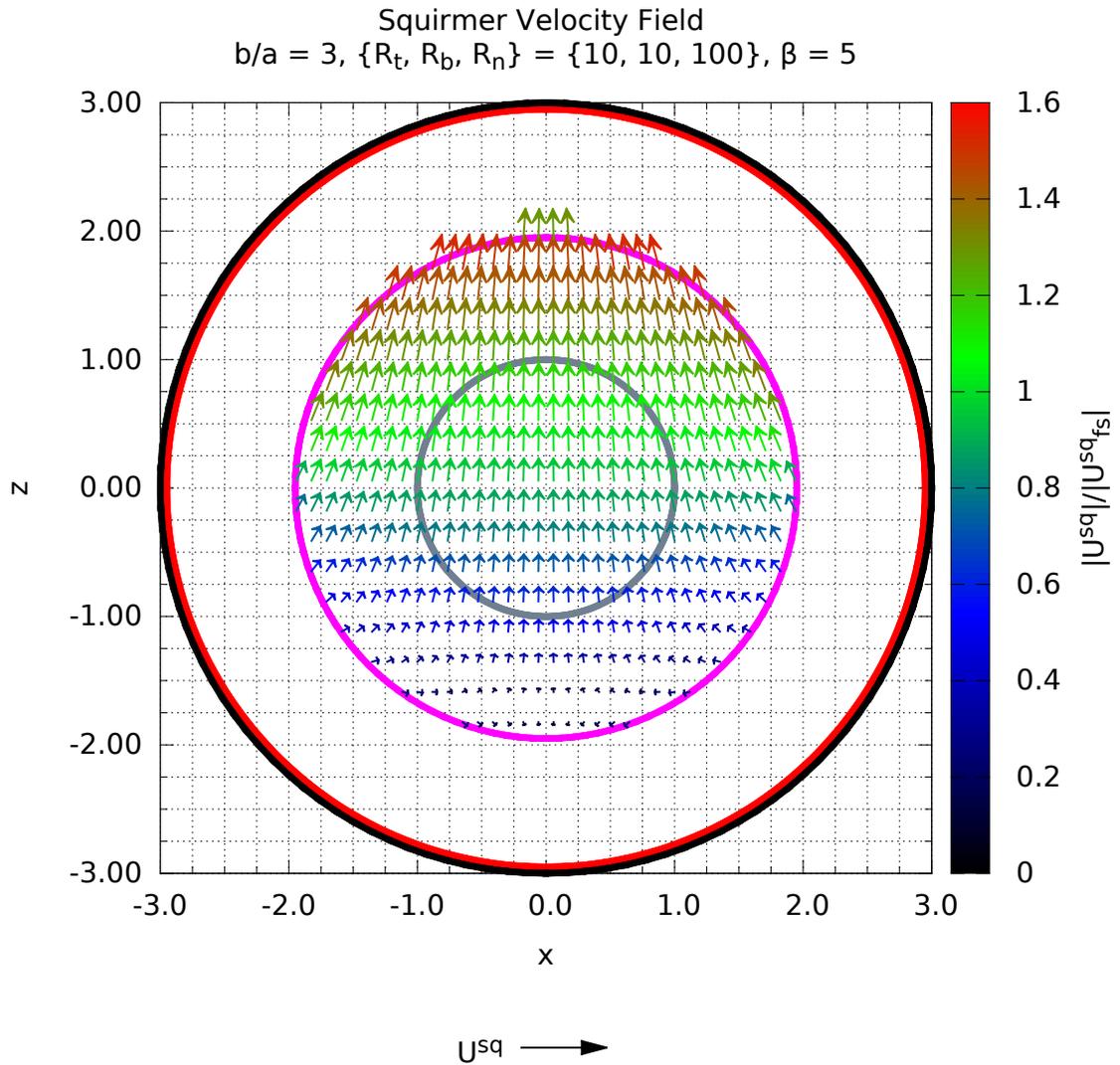


Fig. 5.42 This plot shows the squirmer velocities that would result if a $\beta = 0$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

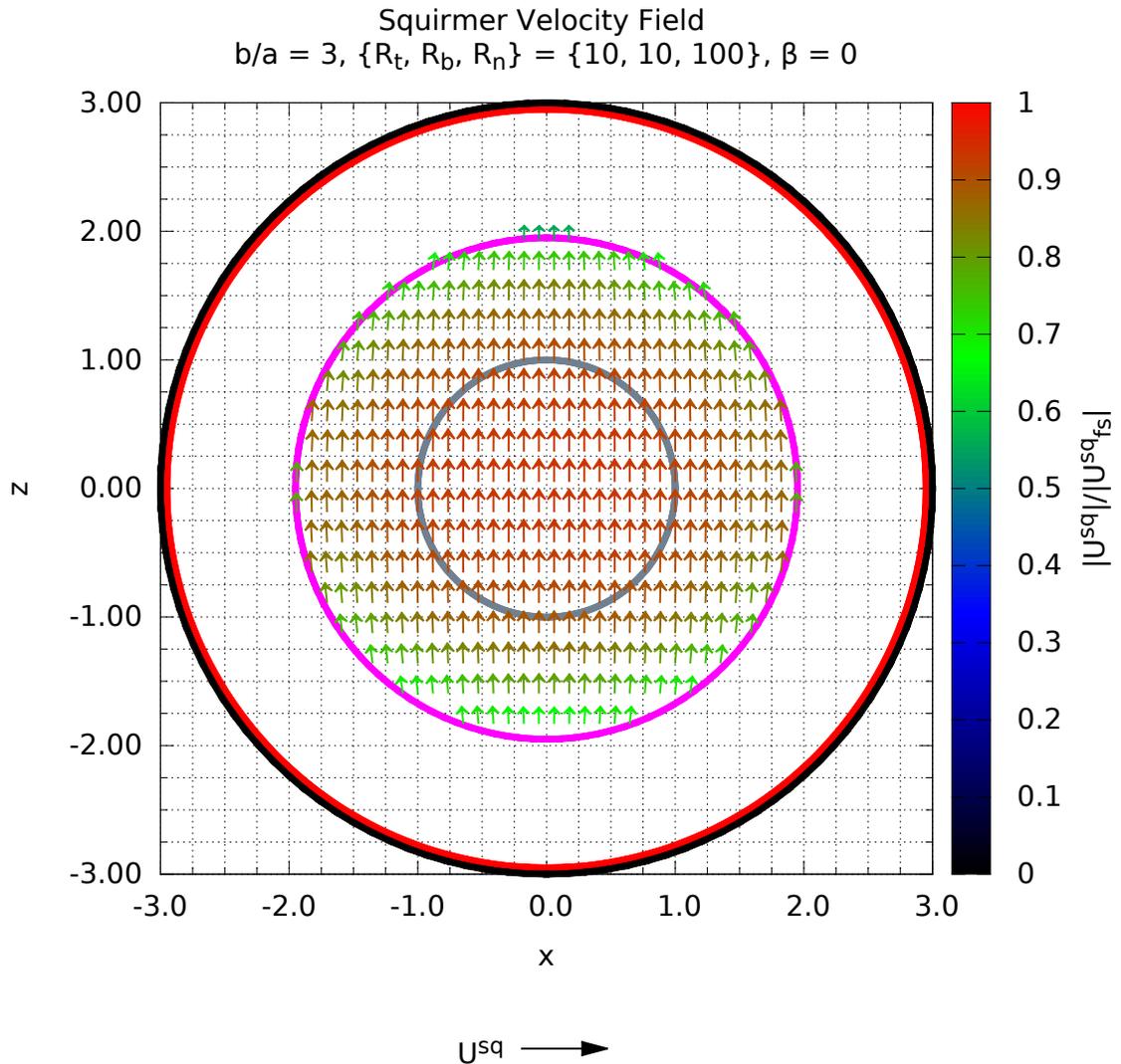


Fig. 5.43 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

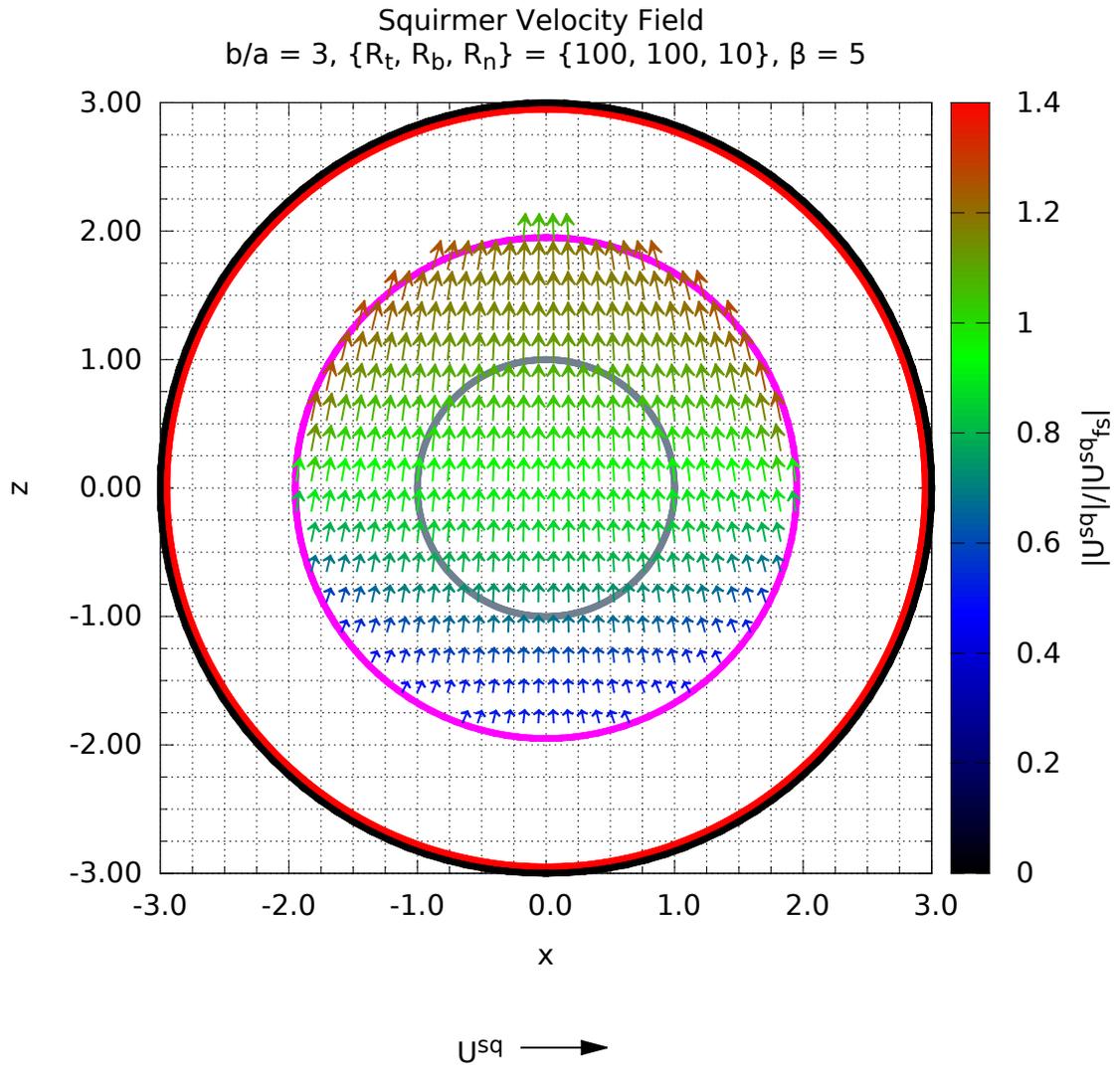


Fig. 5.44 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

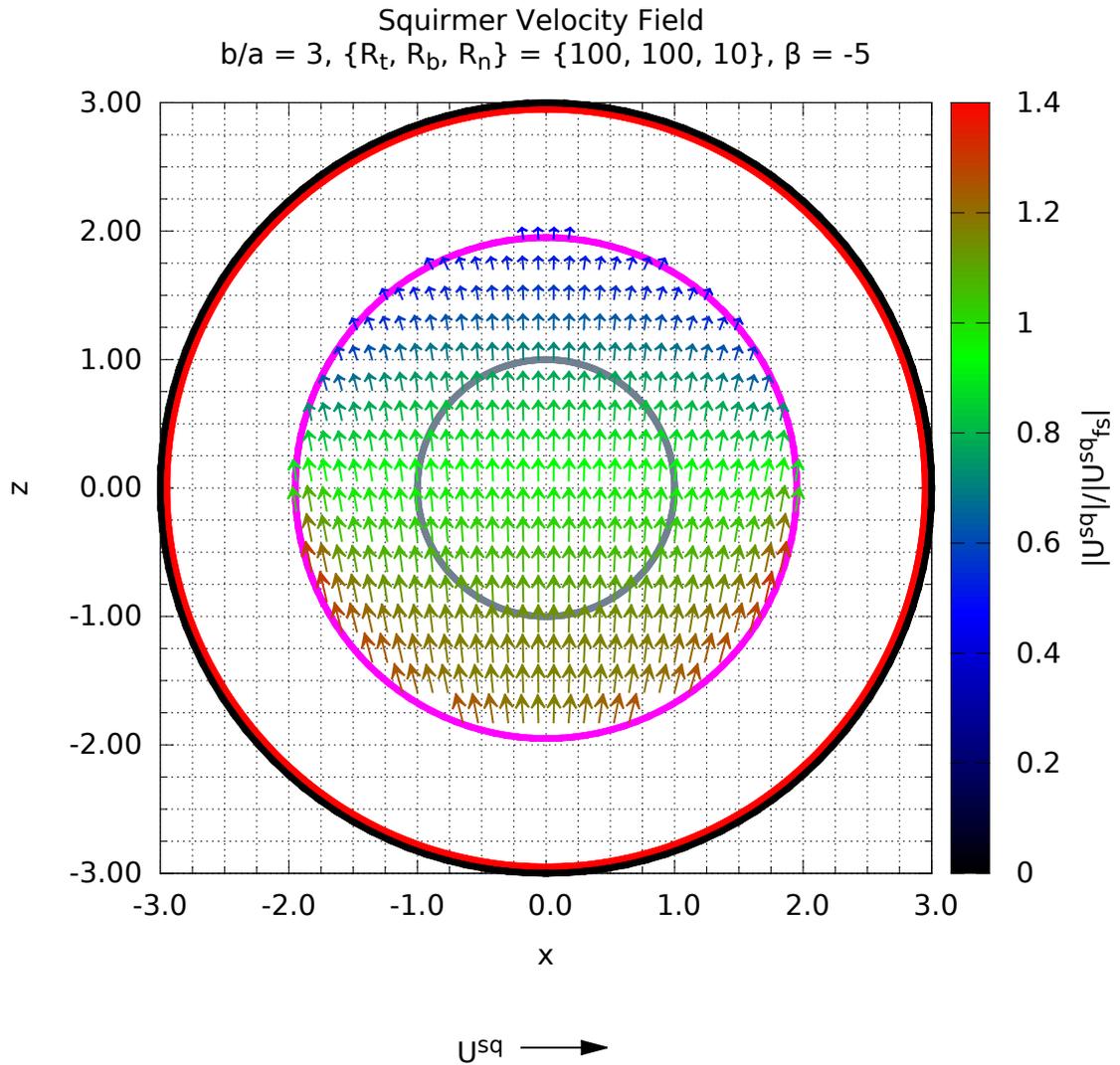


Fig. 5.45 This plot shows the squirmer velocities that would result if a $\beta = 0$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 3$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

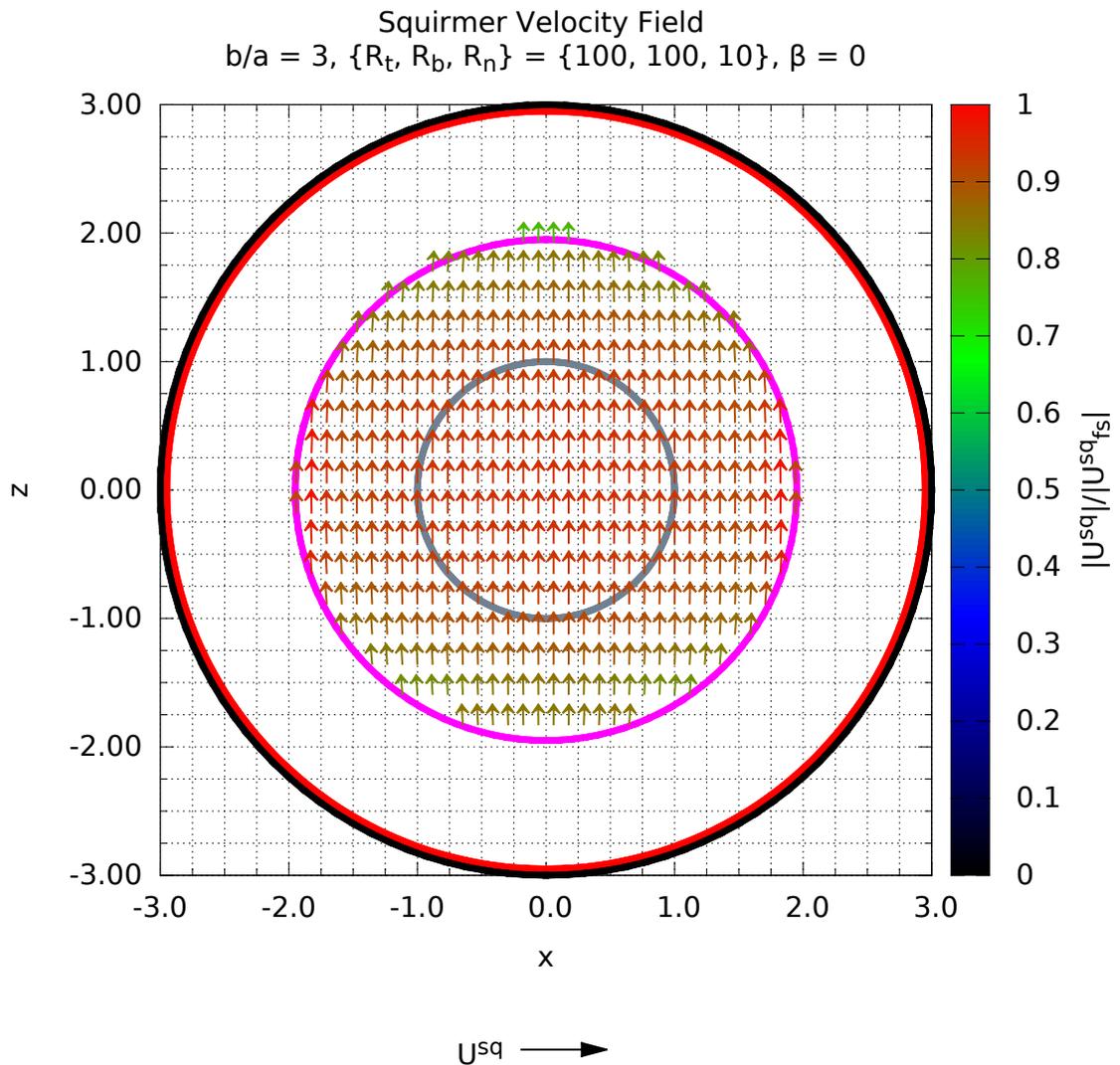


Fig. 5.46 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

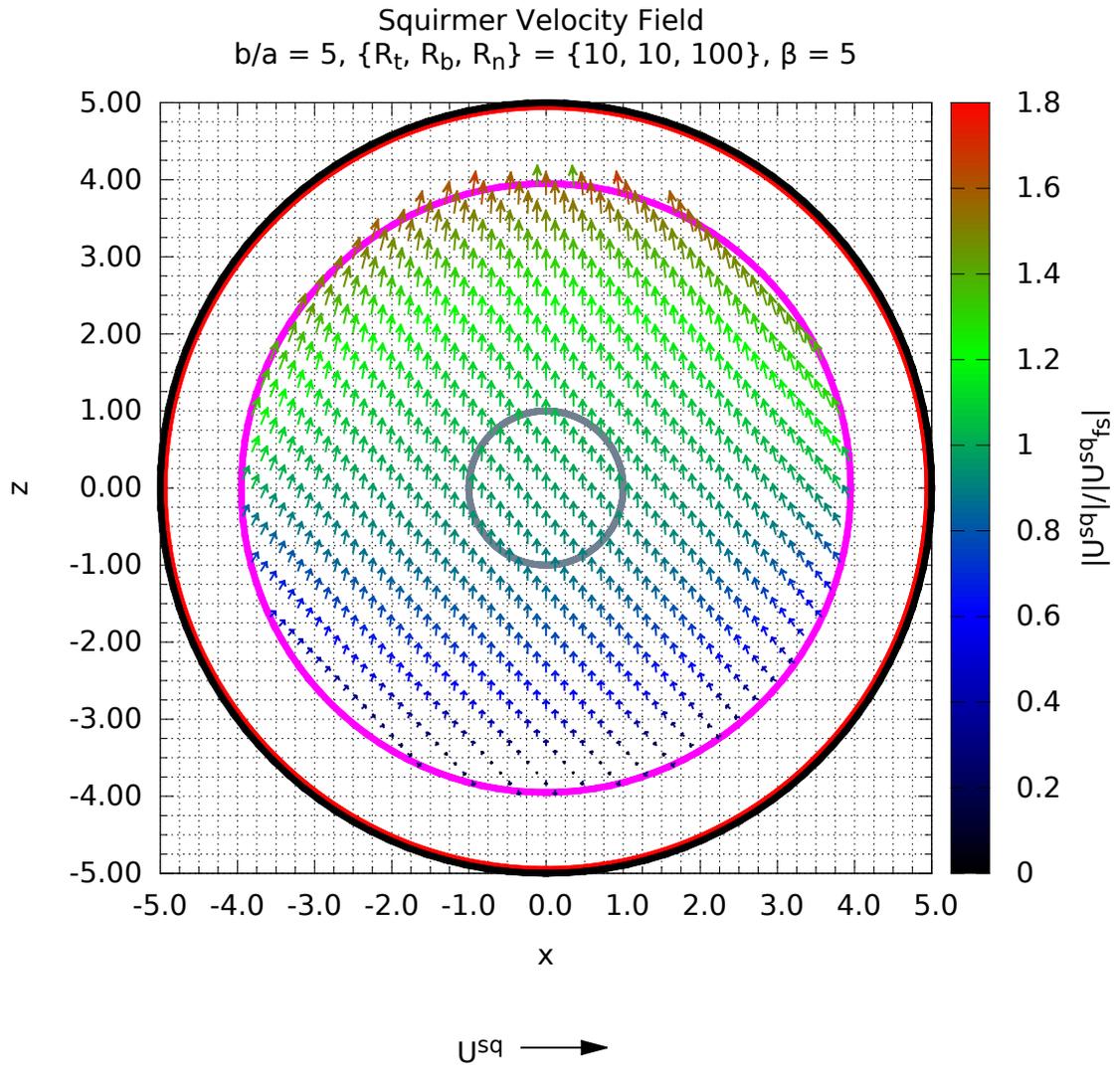


Fig. 5.47 This plot shows the squirmer velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

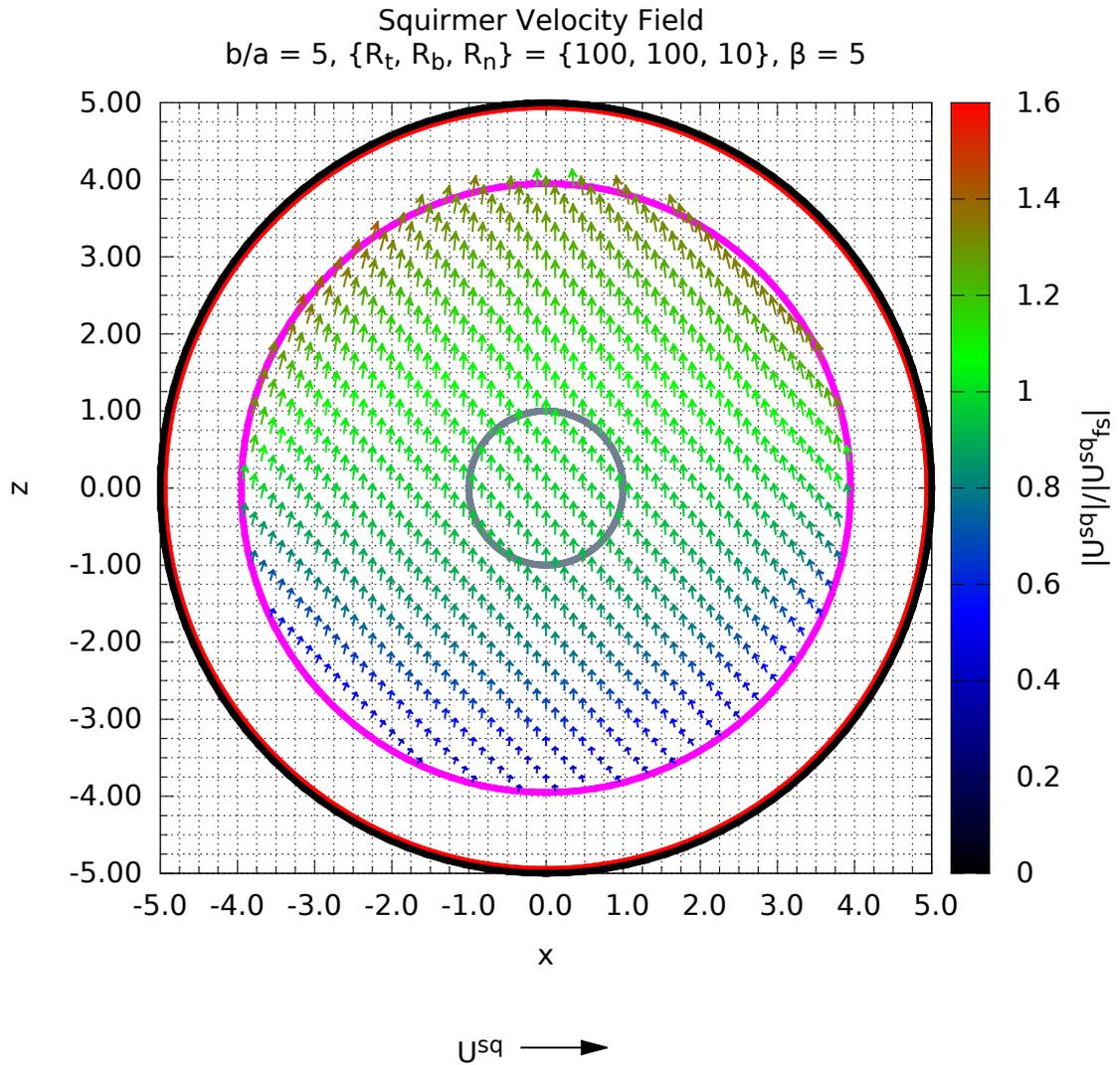


Fig. 5.48 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

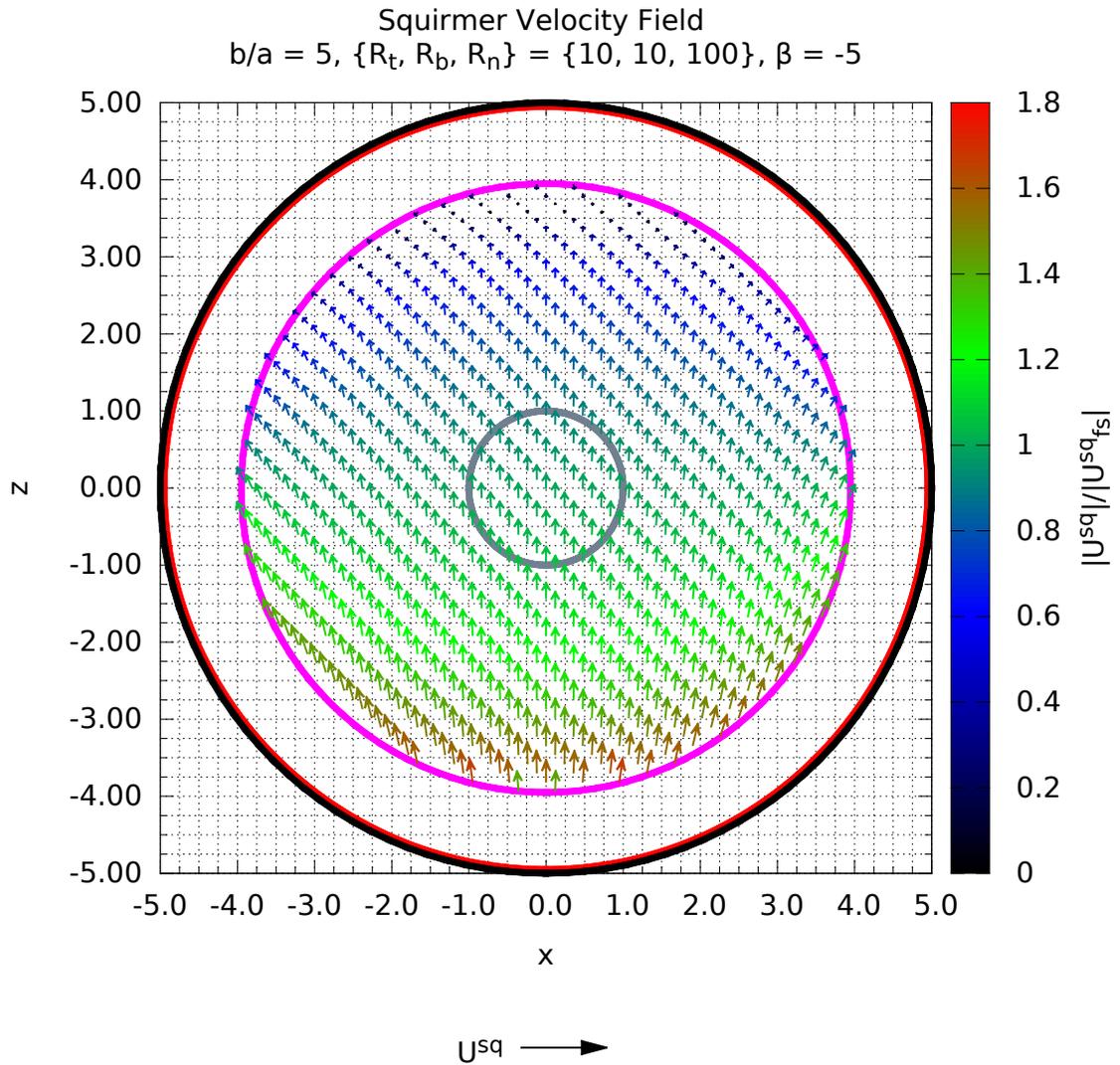


Fig. 5.49 This plot shows the squirmer velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

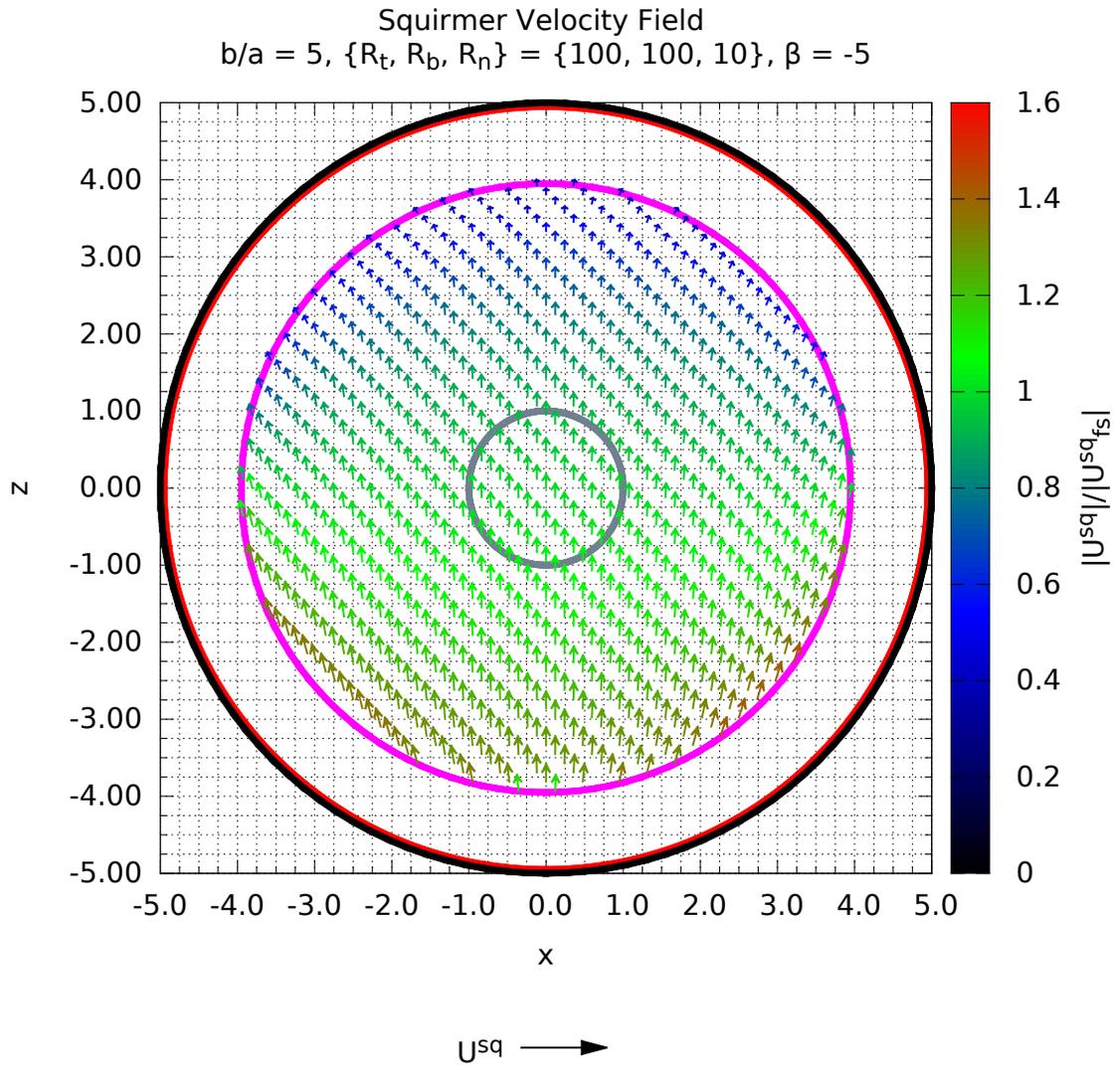


Fig. 5.50 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the tangential direction, $R_t = R_b < R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

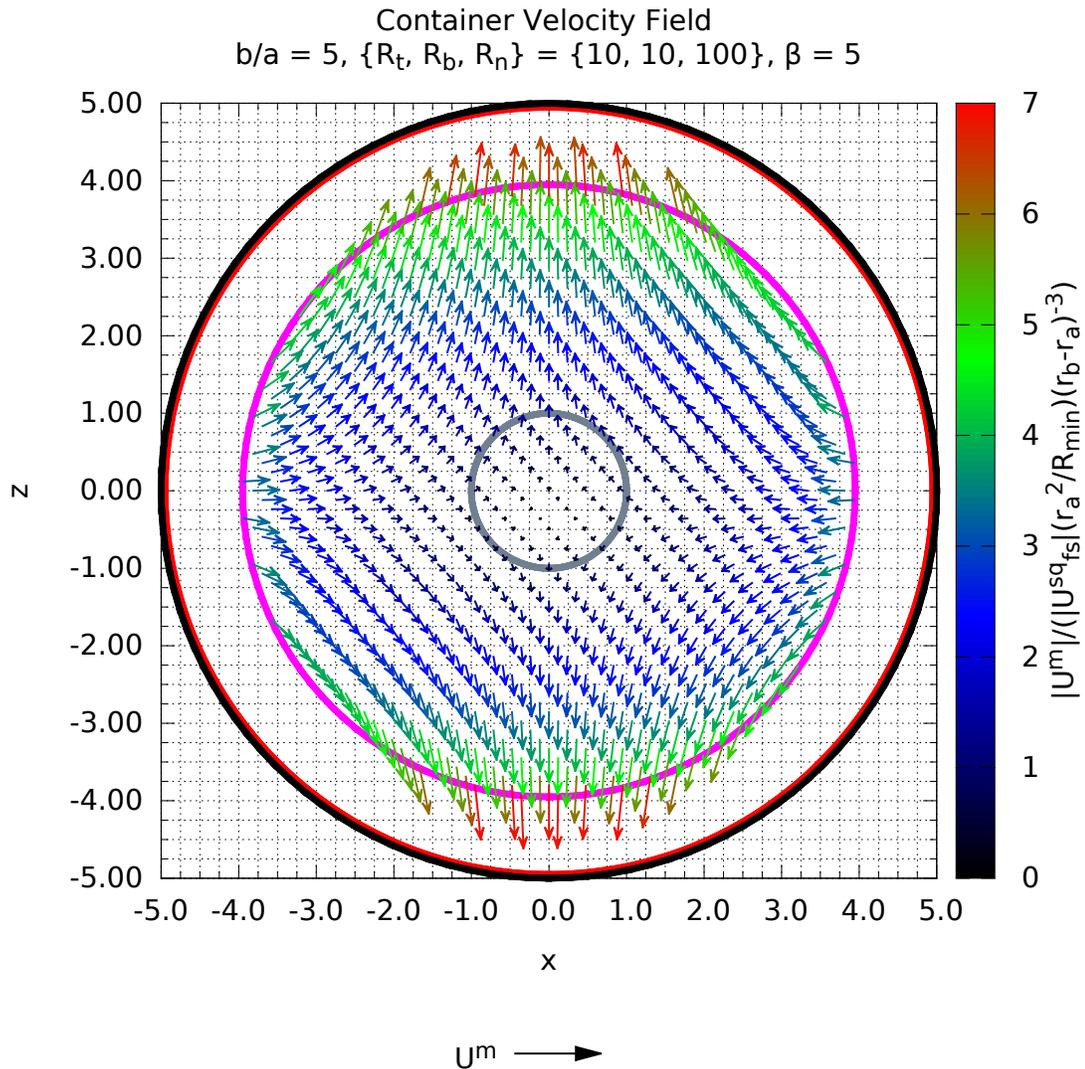


Fig. 5.51 This plot shows the container velocities that would result if a $\beta = +5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

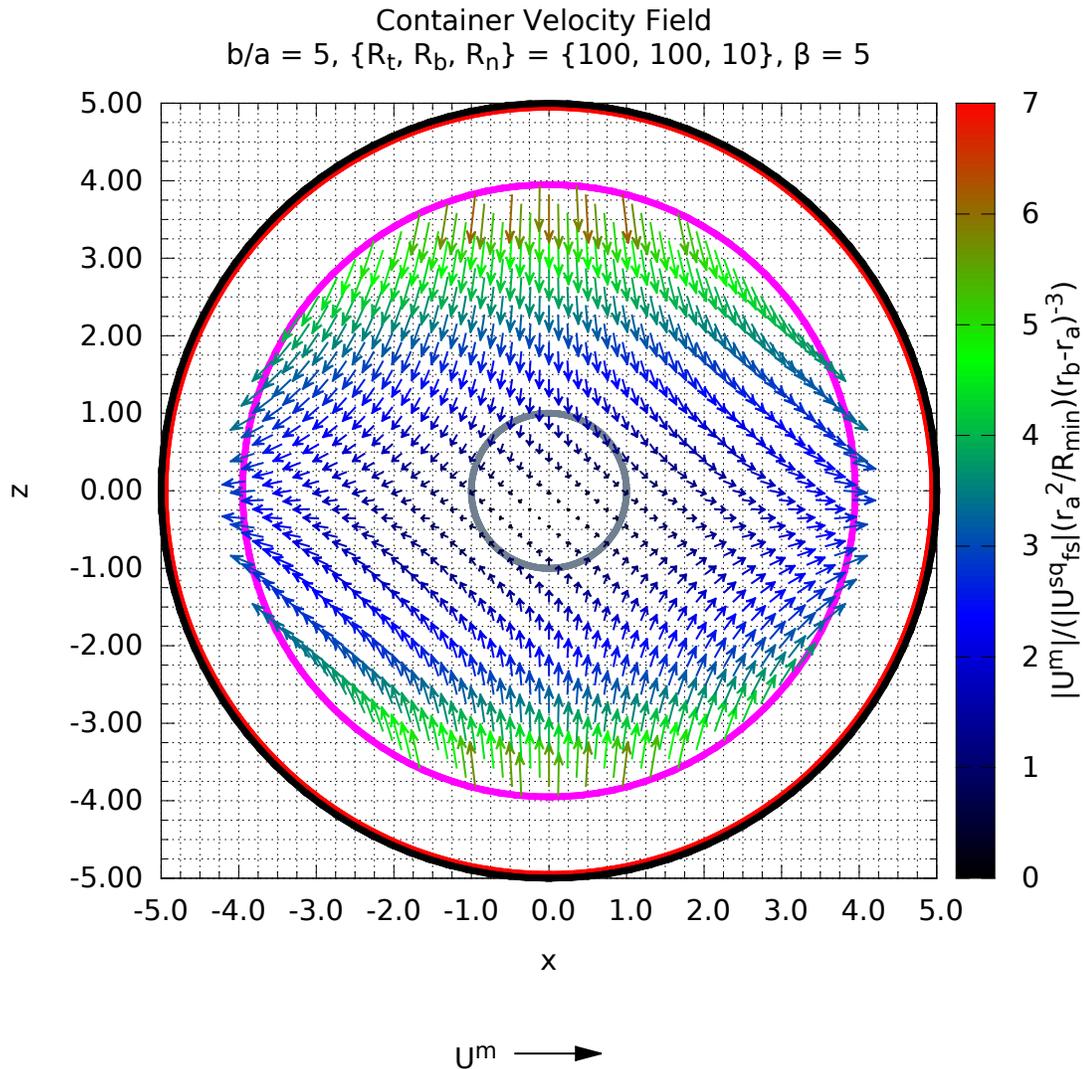


Fig. 5.52 This plot shows the container velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

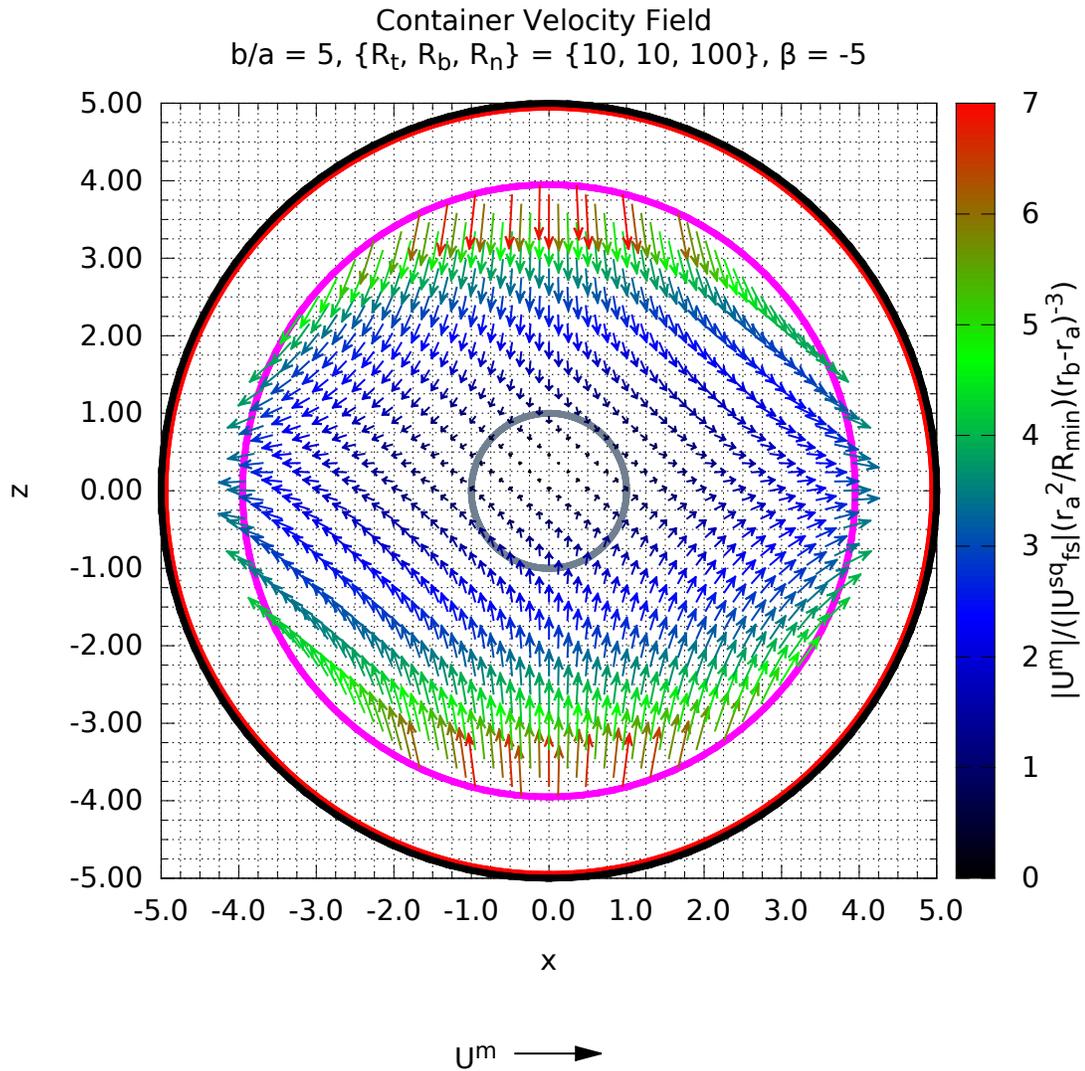


Fig. 5.53 This plot shows the container velocities that would result if a $\beta = -5$ squirmer were placed at various grid points inside of a container with size ratio $b/a = 5$ and weak resistance in the normal direction, $R_t = R_b > R_n$. The black, red, pink, and gray circles give notions of regions where $r = b$, $r = b - a\delta$, $r = r_{cp}^{max}$, and $r = a$ respectively.

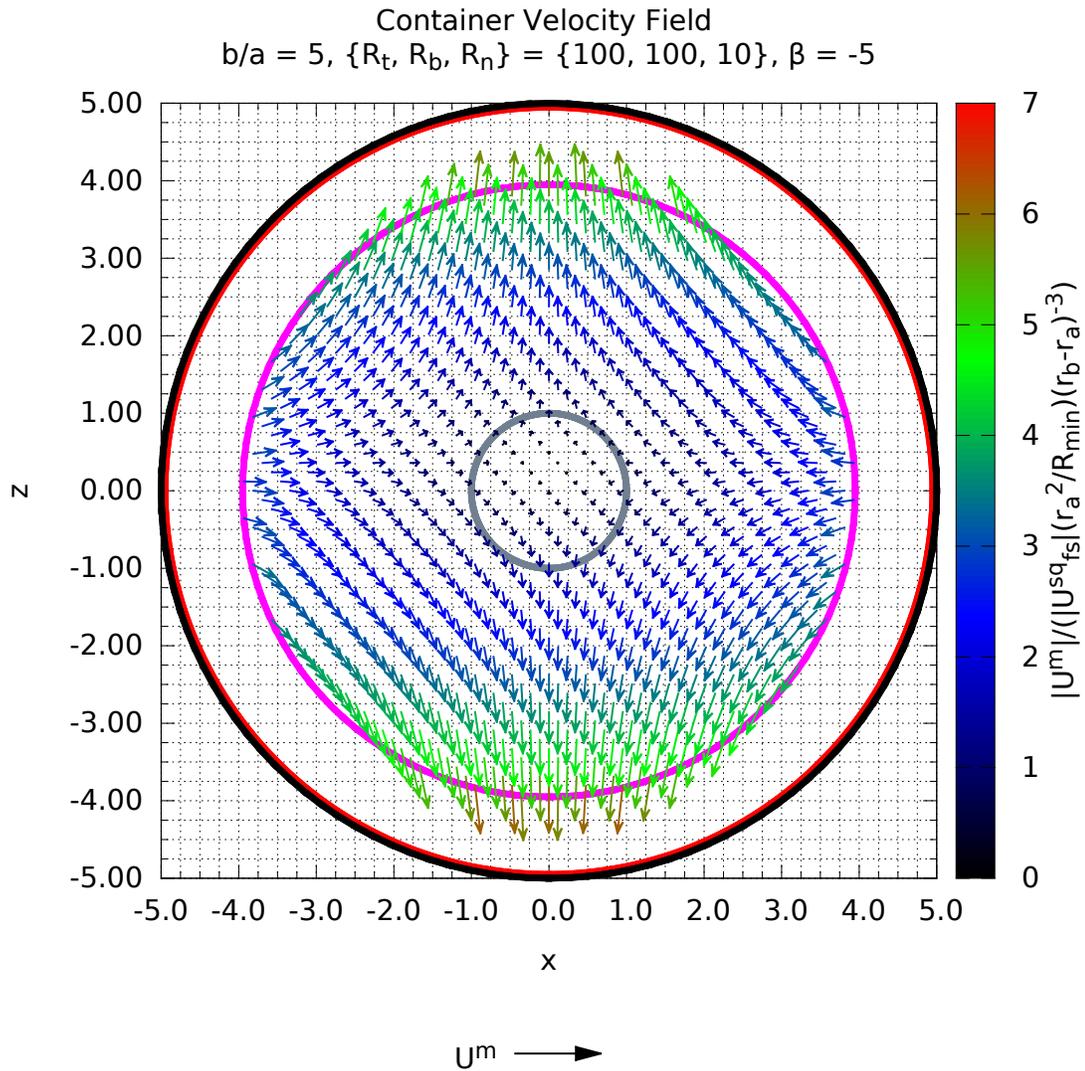


Fig. 5.54 Several squirmer trajectories are shown for $\beta = -5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$.

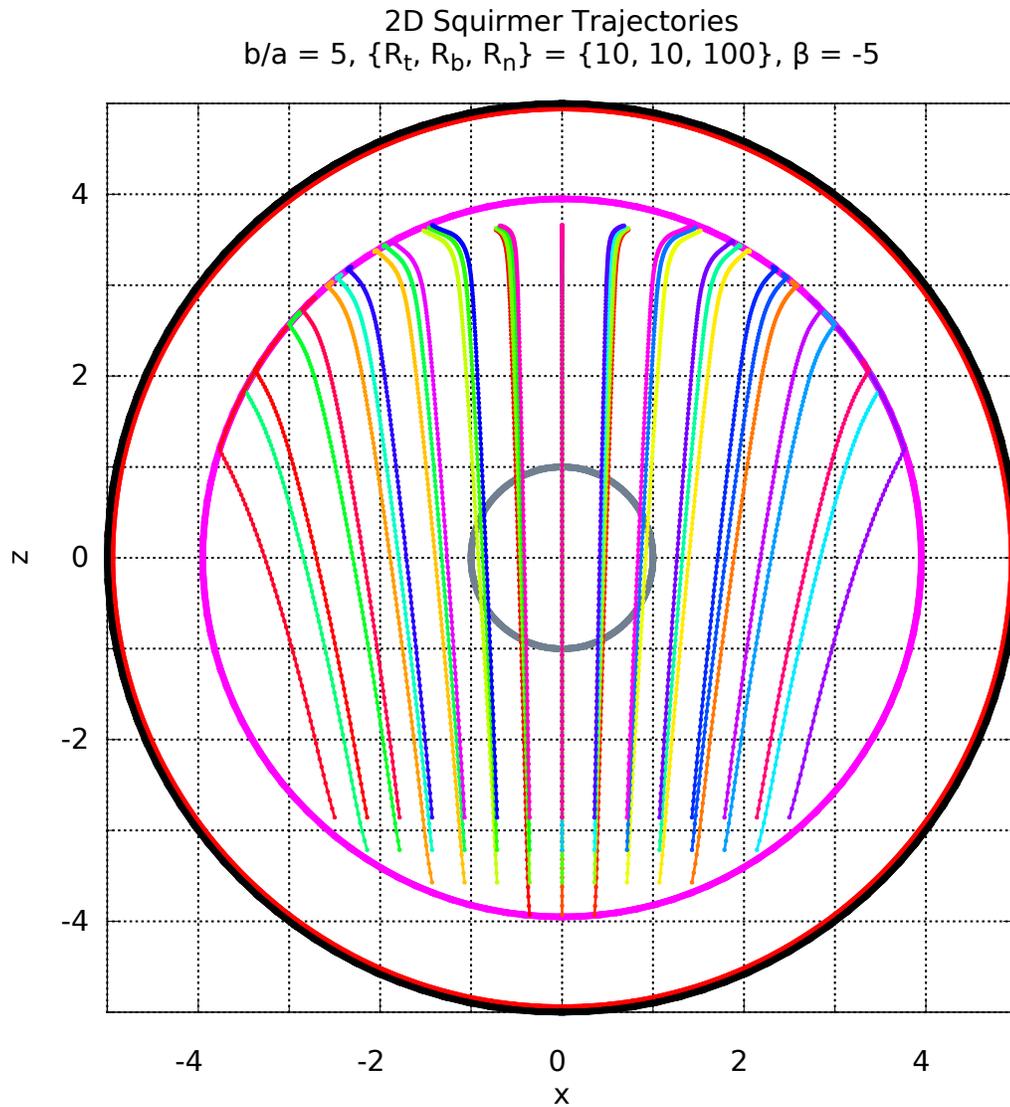


Fig. 5.55 Several squirmer trajectories are shown for $\beta = 0$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$.

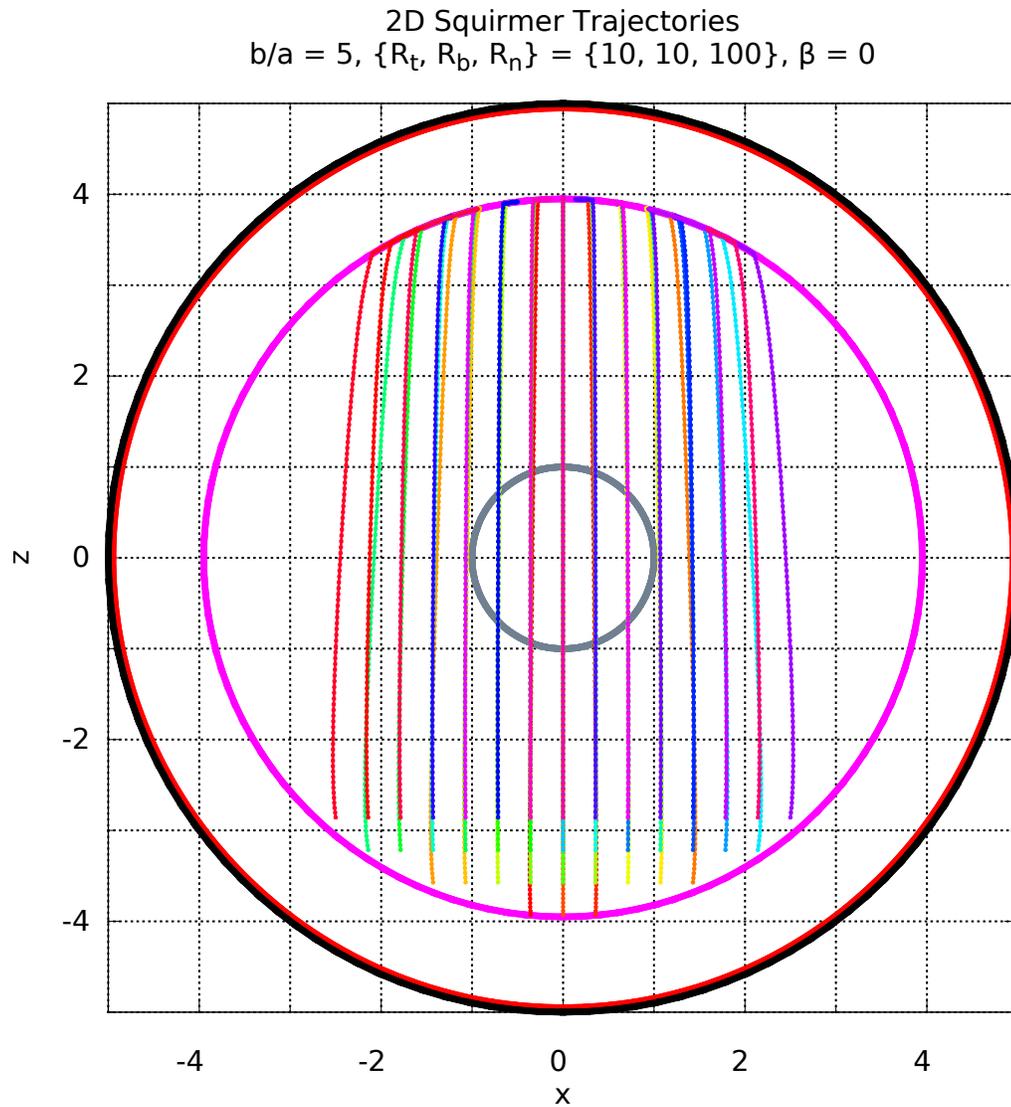


Fig. 5.56 Several squirmer trajectories are shown for $\beta = 5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$.

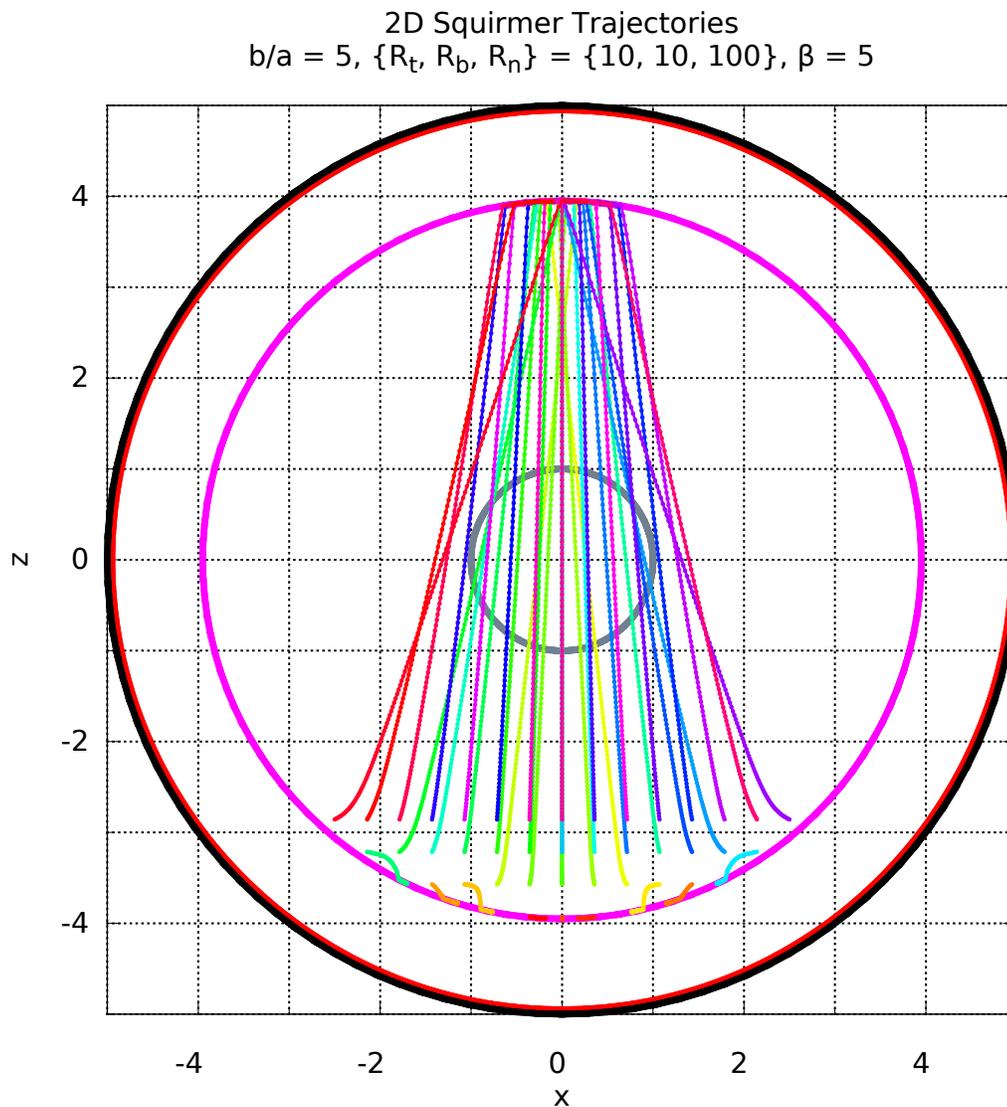


Fig. 5.57 Several squirmer trajectories are shown for $\beta = -5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$.

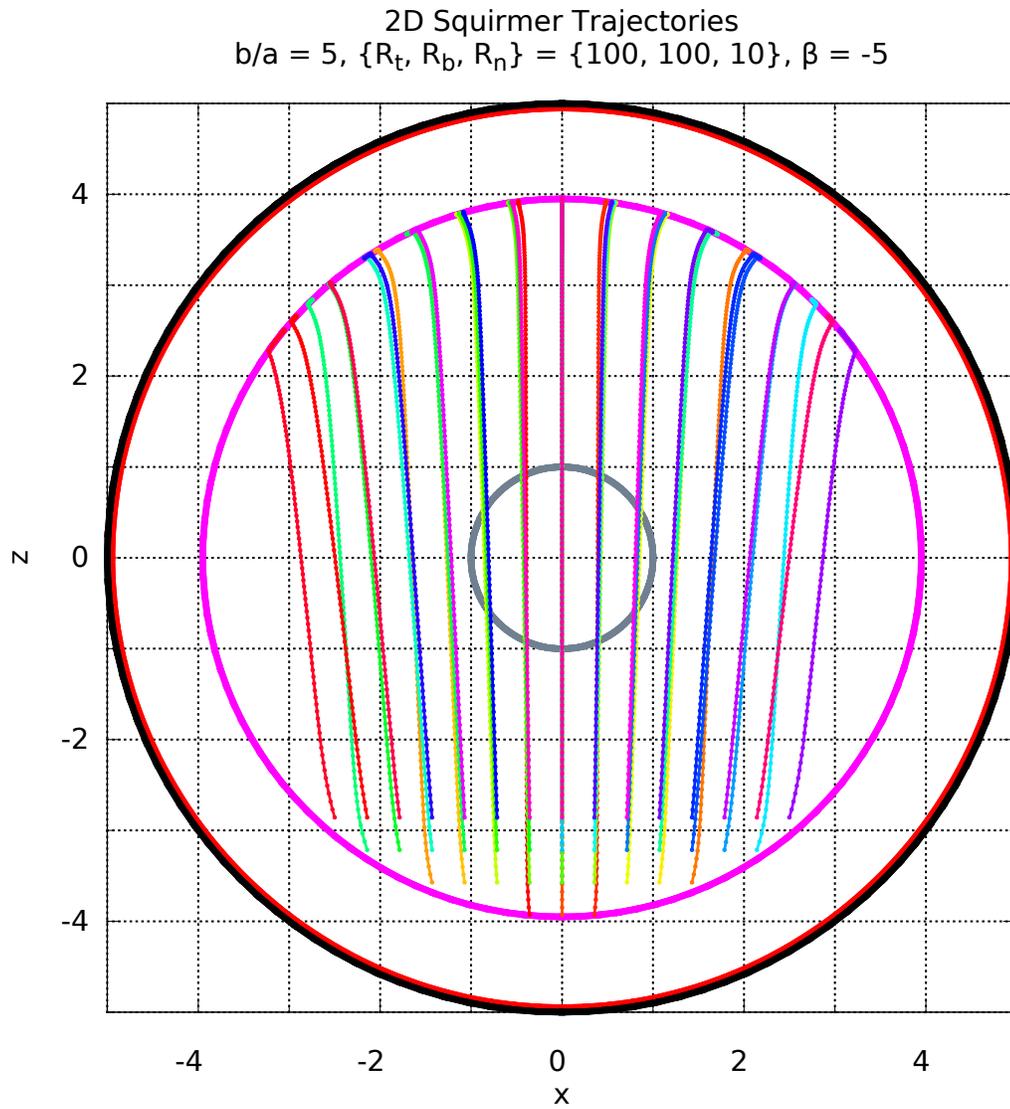


Fig. 5.58 Several squirmer trajectories are shown for $\beta = 0$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$.

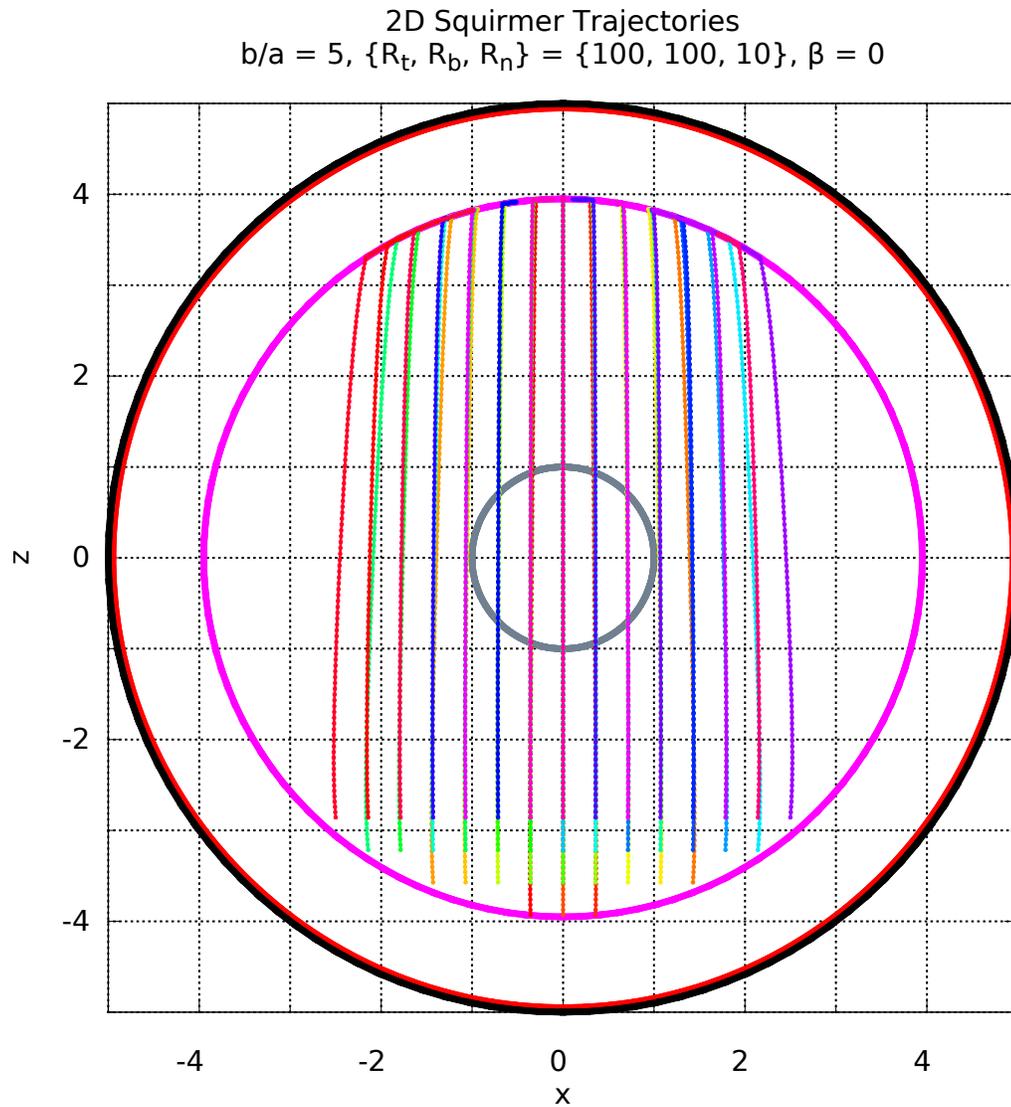


Fig. 5.59 Several squirmer trajectories are shown for $\beta = 5$ squirmers. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$.

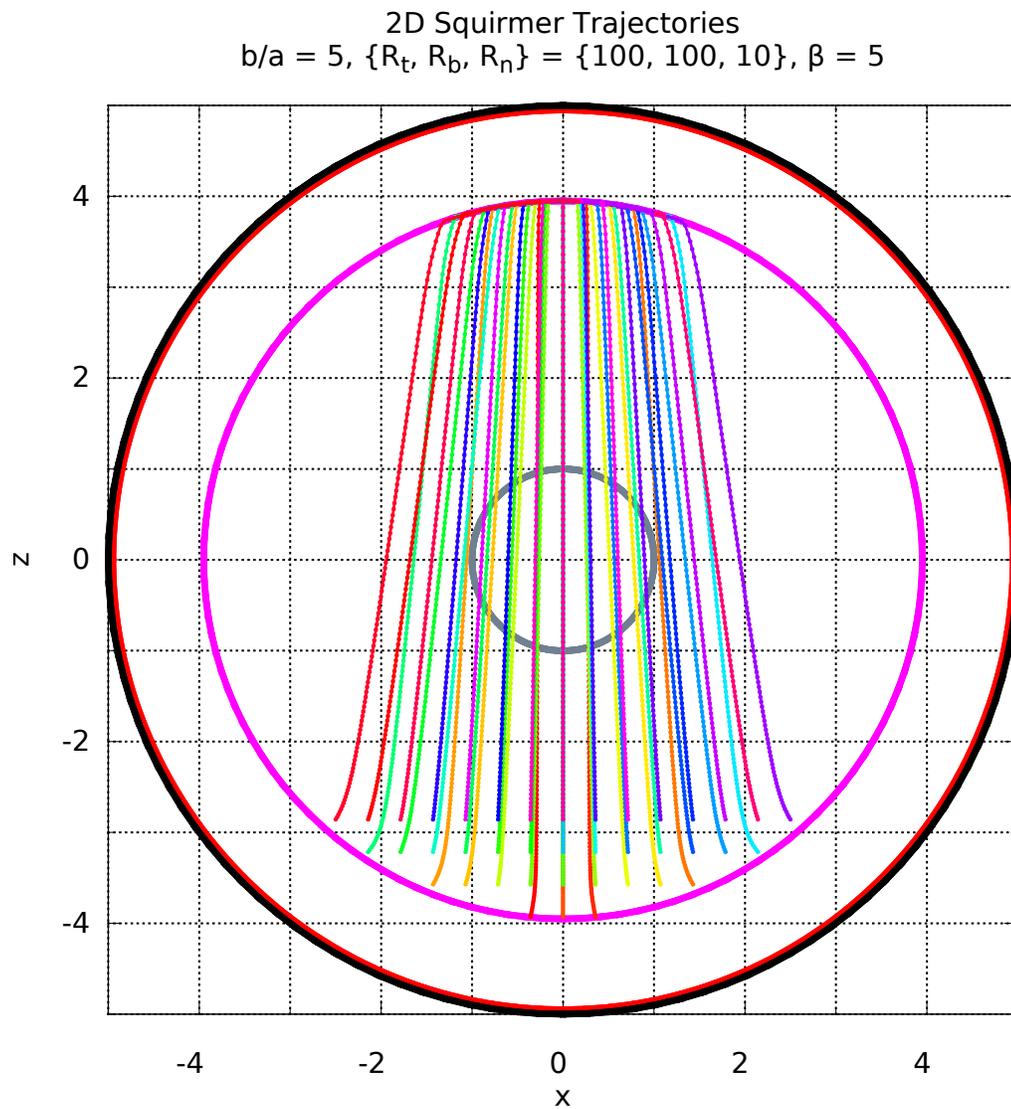


Fig. 5.60 Trajectories for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially outward (different from the $\beta = +5$ type squirmer) as they swim upward in z .

3D Squirmer Trajectories, U^{sq}
 $b/a = 5, \{R_t, R_b, R_n\} = \{10, 10, 100\}, \beta = -5$

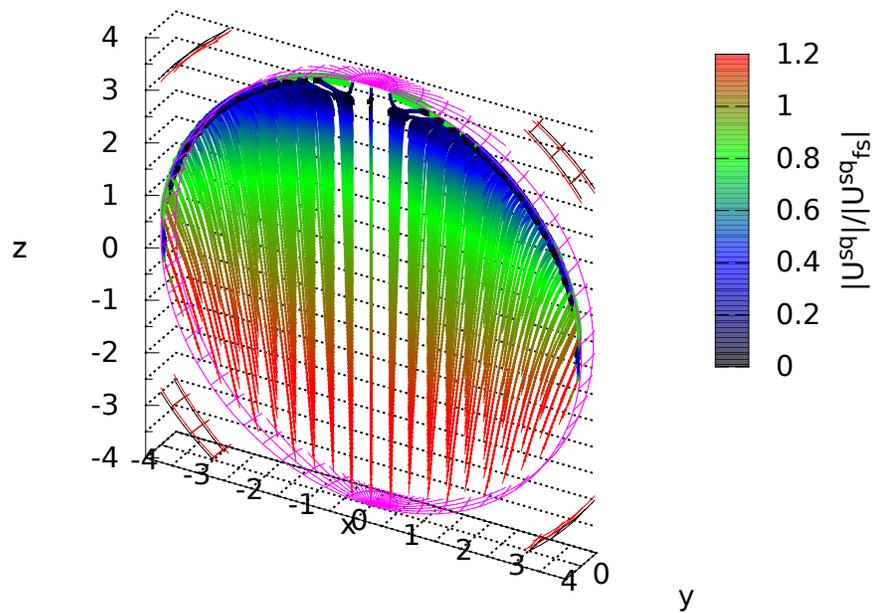


Fig. 5.61 Trajectories for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to swim along z showing only small radial migrations as they near the container wall (different from the $\beta = \pm 5$ type of squirmers).

3D Squirmer Trajectories, U^{sq}
 $b/a = 5$, $\{R_t, R_b, R_n\} = \{10, 10, 100\}$, $\beta = 0$

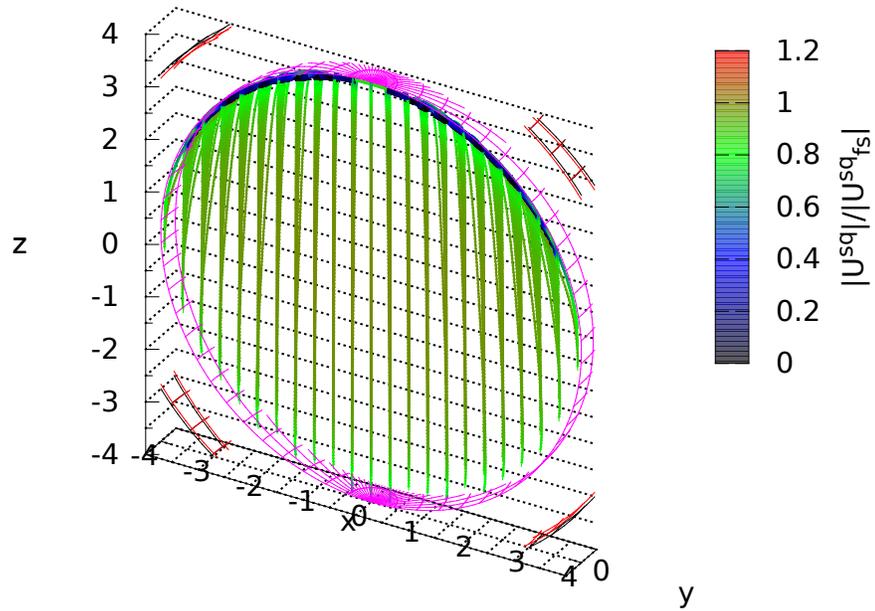


Fig. 5.62 Trajectories for $\beta = +5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially inward as they swim upward in z .

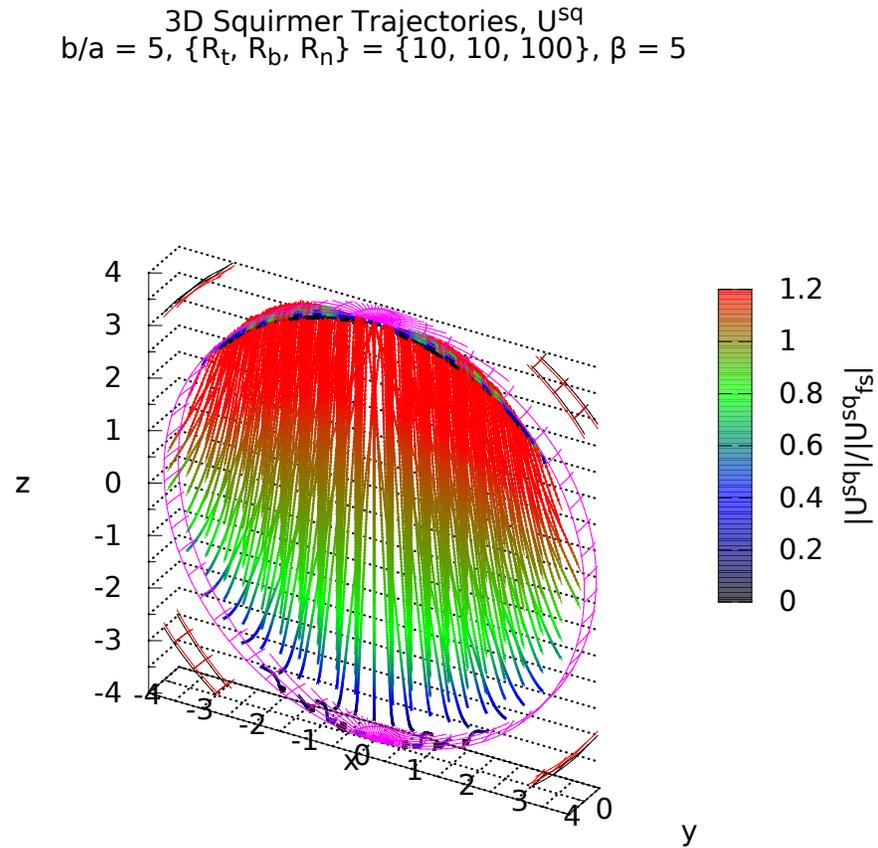


Fig. 5.63 Trajectories for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to migrate radially outward (different from the $\beta = +5$ type squirmer) as they swim upward in z . This effect is much weak than was seen for the weak tangential resistance case.

3D Squirmer Trajectories, U^{sq}
 $b/a = 5, \{R_t, R_b, R_n\} = \{100, 100, 10\}, \beta = -5$

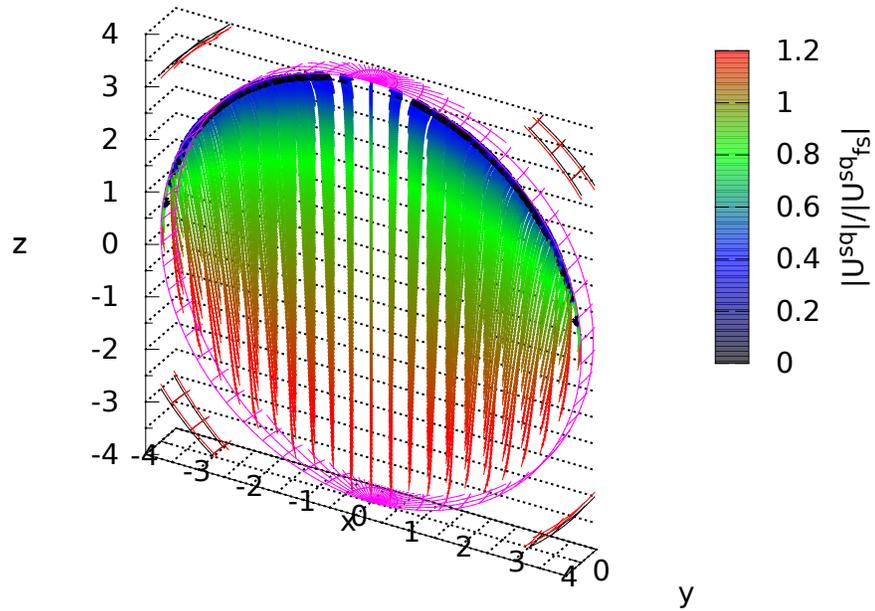


Fig. 5.64 Trajectories for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The squirming particles tend to swim along z showing only small radial migrations as they near the container wall (different from the $\beta = \pm 5$ type of squirmers).

3D Squirmer Trajectories, U^{sq}
 $b/a = 5$, $\{R_t, R_b, R_n\} = \{100, 100, 10\}$, $\beta = 0$

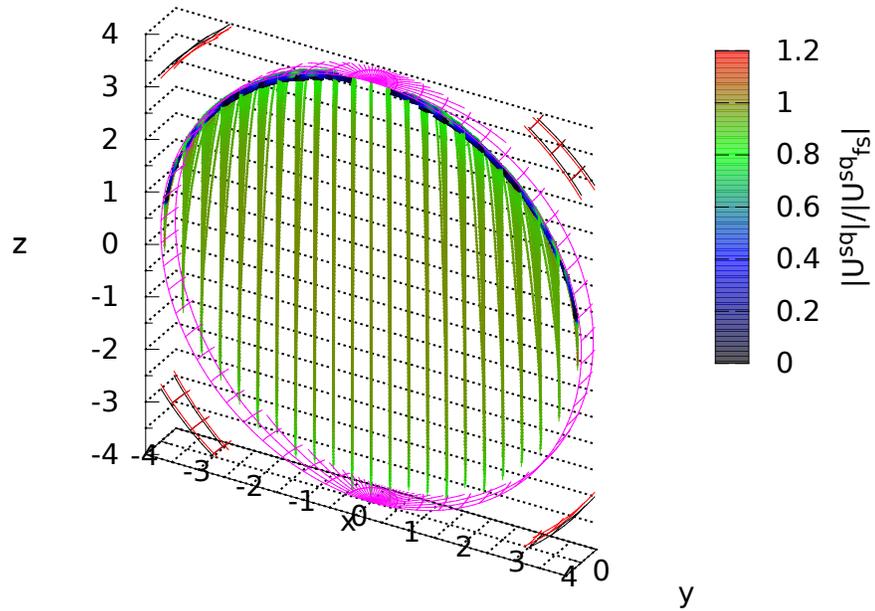


Fig. 5.65 Trajectories for $\beta = +5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 3$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$. The squirming particles tend to migrate radially inward as they swim upward in z . This effect is much weaker than was seen for the weak tangential resistance case.

3D Squirmer Trajectories, U^{sq}
 $b/a = 5$, $\{R_t, R_b, R_n\} = \{100, 100, 10\}$, $\beta = 5$

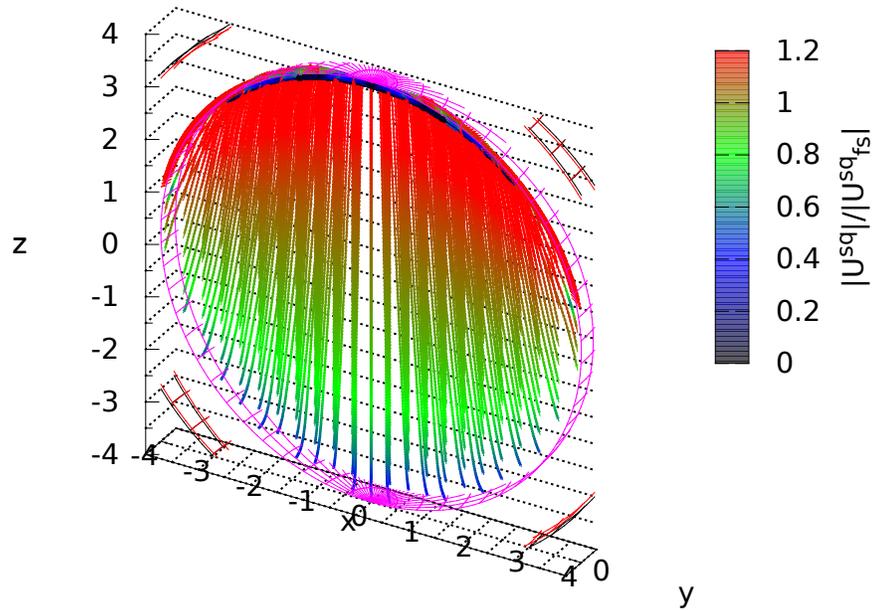


Fig. 5.66 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane.

3D Squirmer Trajectories, Ω^{sq}
 $b/a = 5, \{R_t, R_b, R_n\} = \{10, 10, 100\}, \beta = -5$

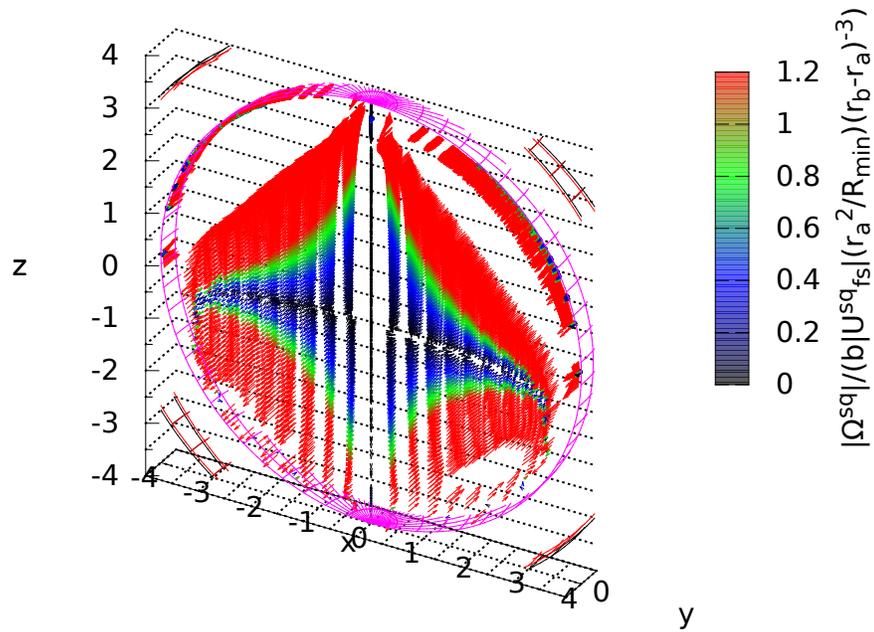


Fig. 5.67 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane.

3D Squirmer Trajectories, Ω^{sq}
 $b/a = 5$, $\{R_t, R_b, R_n\} = \{10, 10, 100\}$, $\beta = 0$

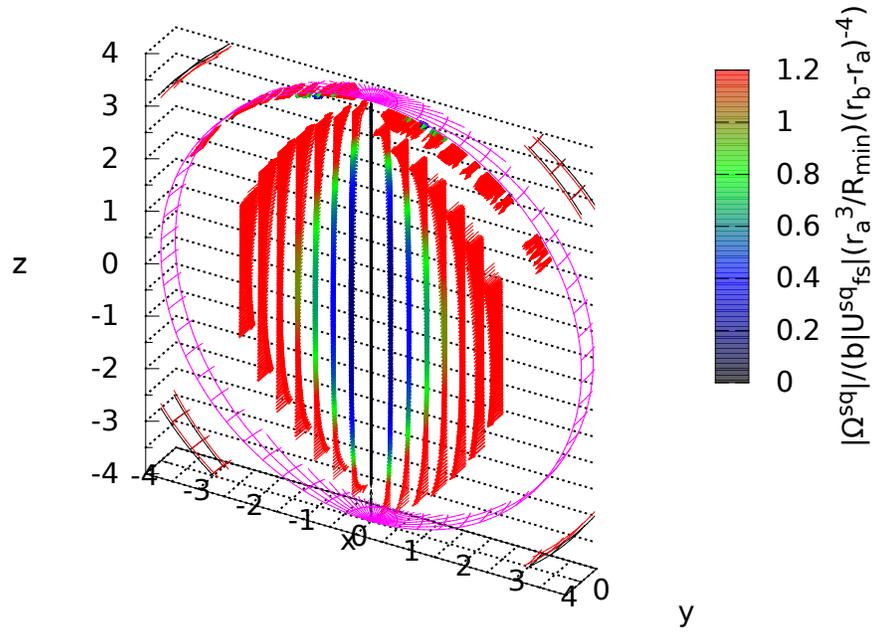


Fig. 5.68 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane.

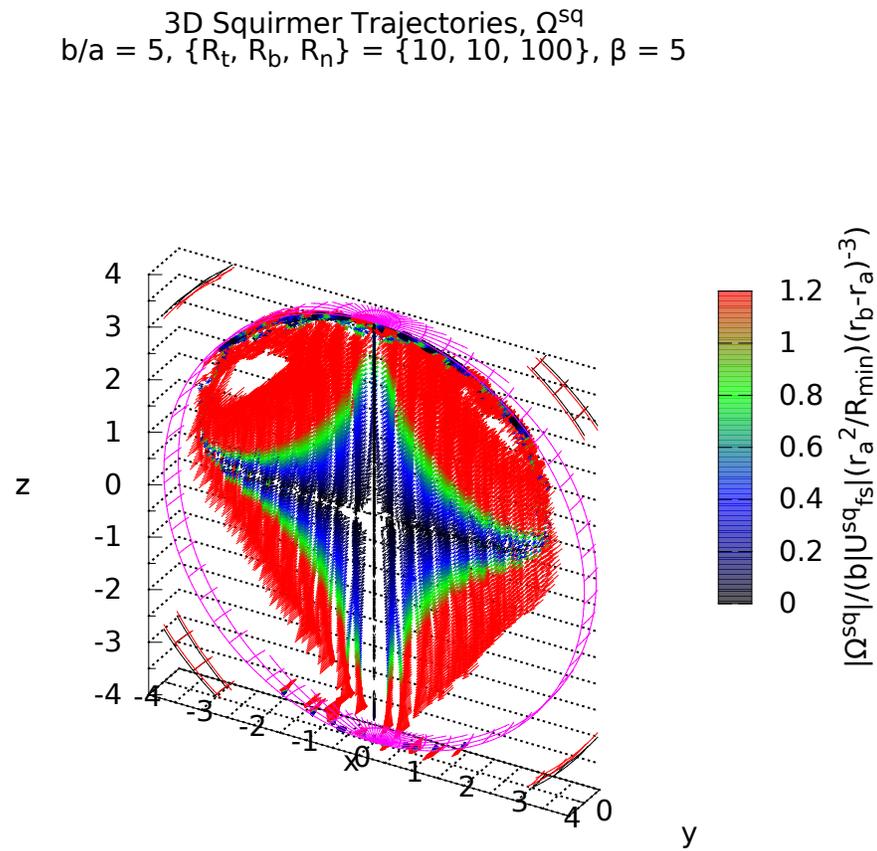


Fig. 5.69 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane.

3D Squirmer Trajectories, Ω^{sq}
 $b/a = 5$, $\{R_t, R_b, R_n\} = \{100, 100, 10\}$, $\beta = -5$

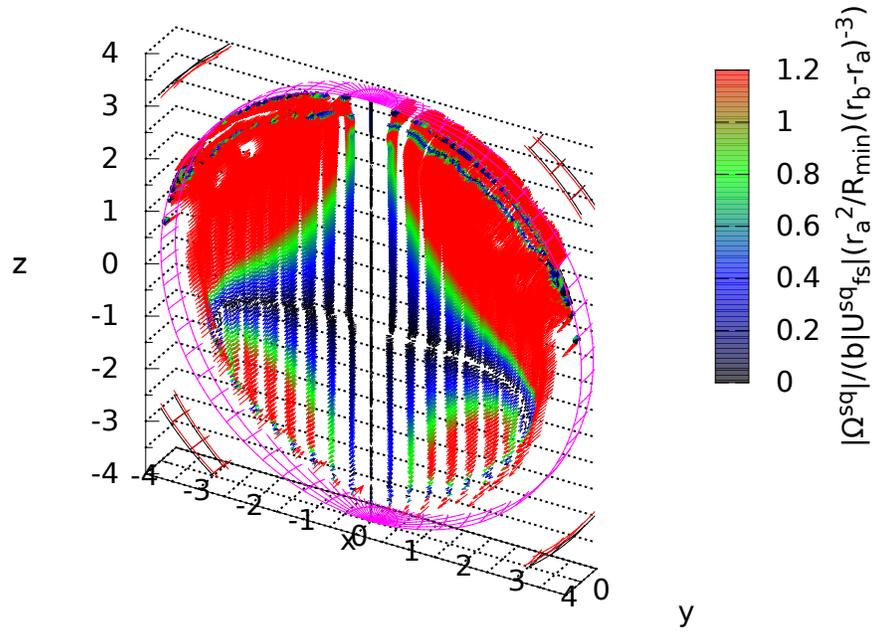


Fig. 5.70 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane.

3D Squirmer Trajectories, Ω^{sq}
 $b/a = 5, \{R_t, R_b, R_n\} = \{100, 100, 10\}, \beta = 0$

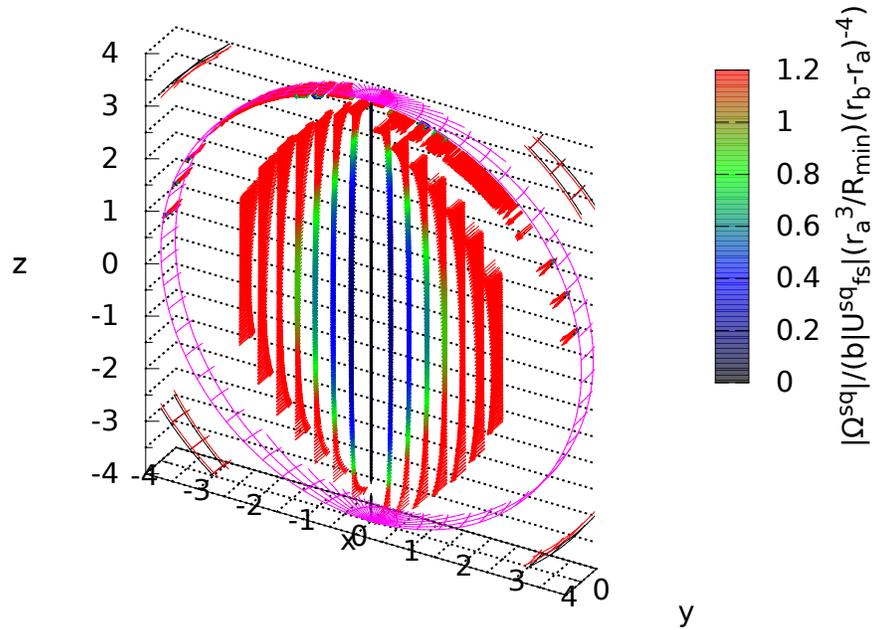


Fig. 5.71 Trajectories supplemented with a scaled squirmer angular velocity Ω^{sq} for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid) are shown for the case where $b/a = 5$, $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$. The angular velocity is distributed with anti-parallel direction but symmetric magnitude across the yz lane.

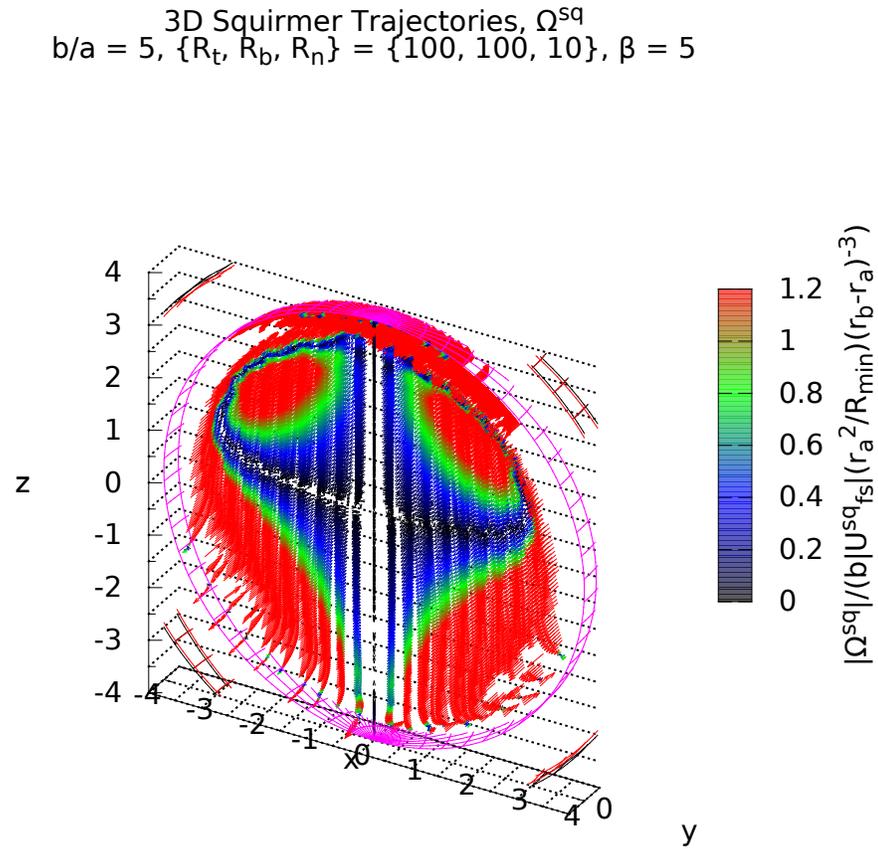


Fig. 5.72 Container trajectories are shown for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$.

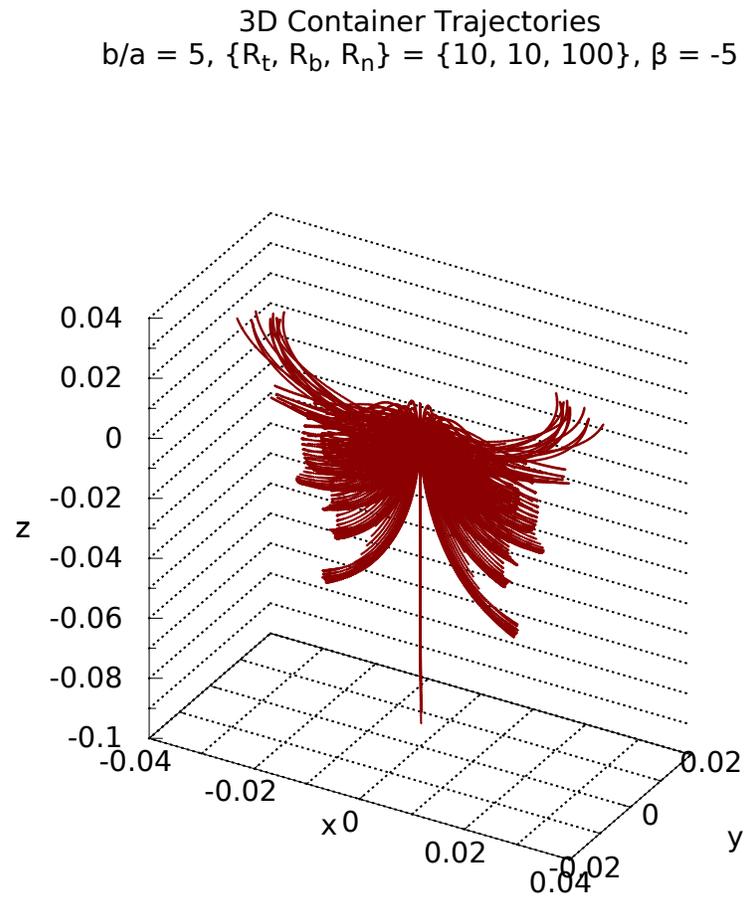


Fig. 5.73 Container trajectories are shown for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

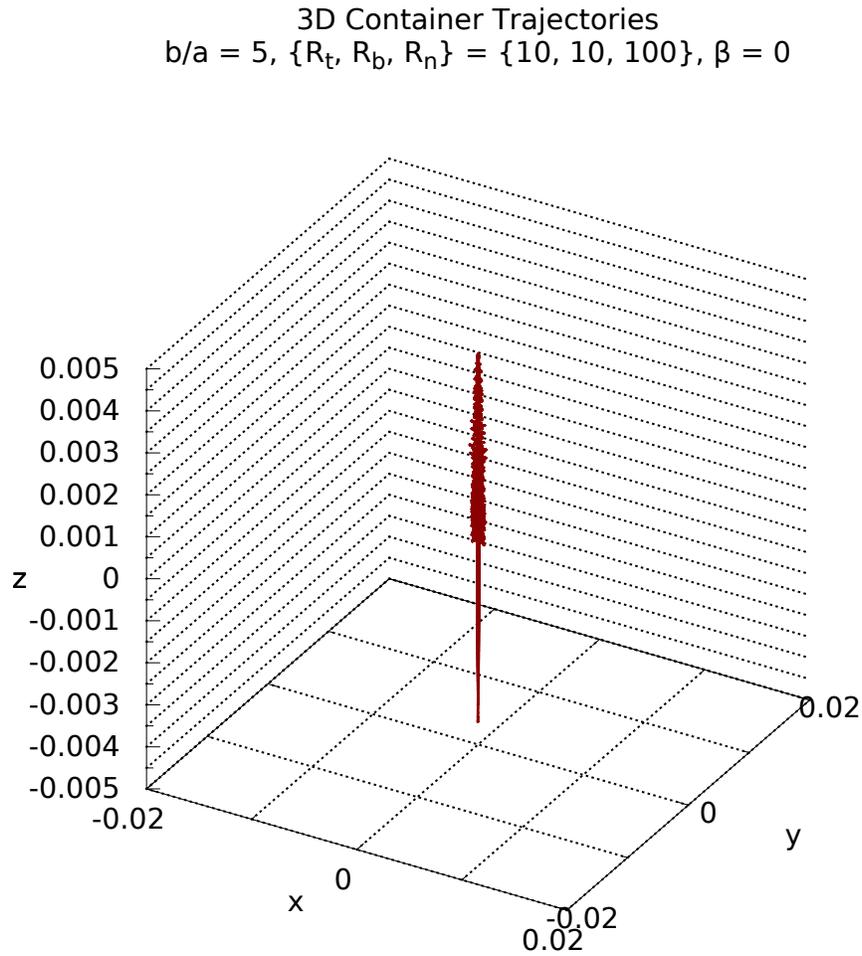


Fig. 5.74 Container trajectories are shown for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $e_s = e_z$.

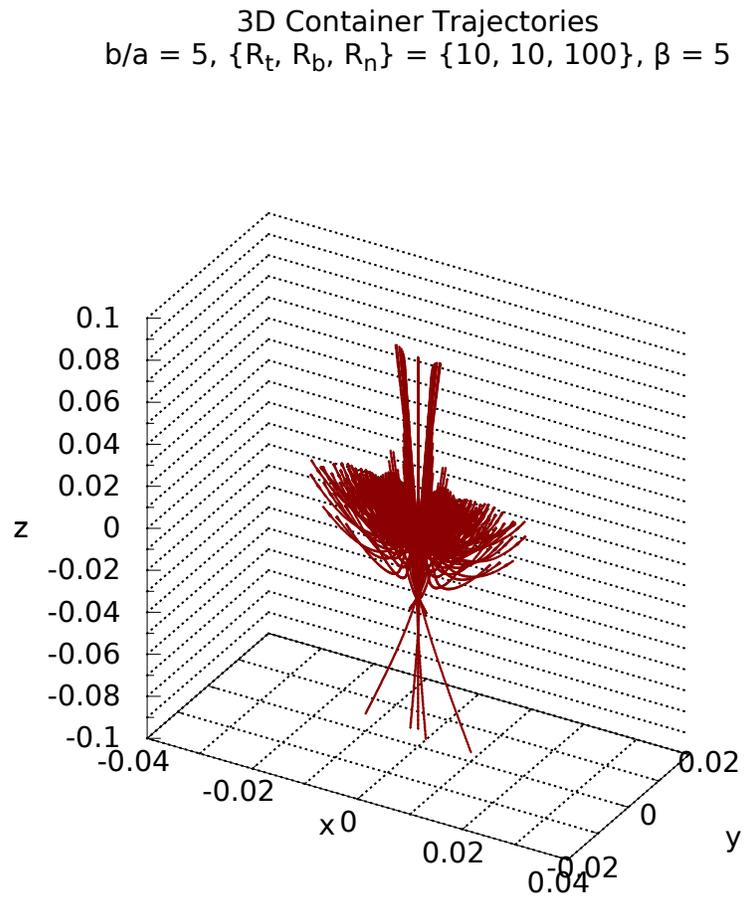


Fig. 5.75 Container trajectories are shown for $\beta = -5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$.

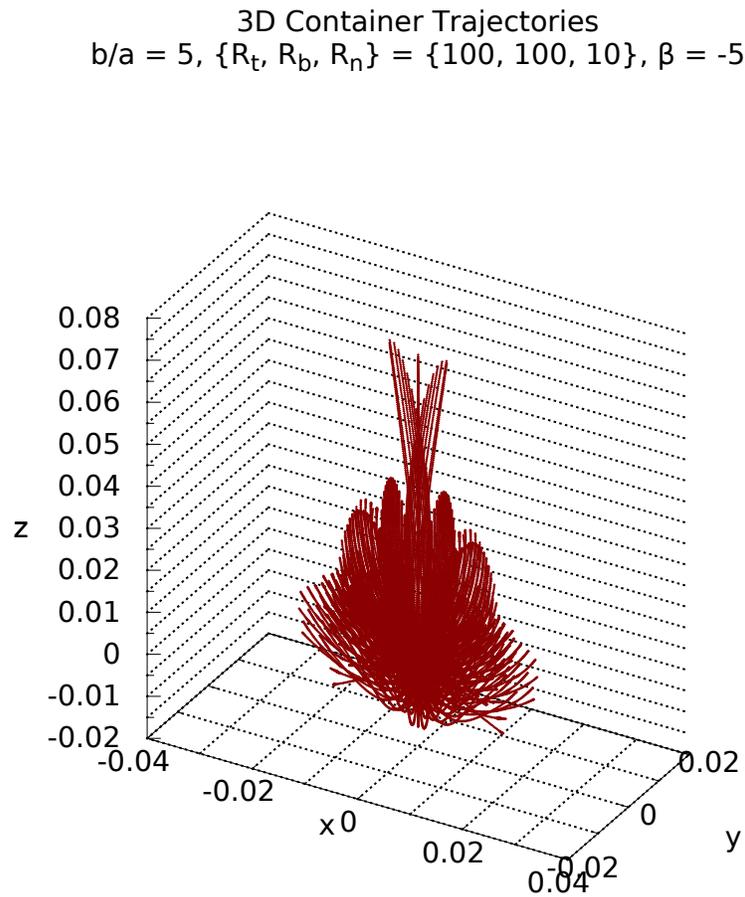


Fig. 5.76 Container trajectories are shown for $\beta = 0$ neutral squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

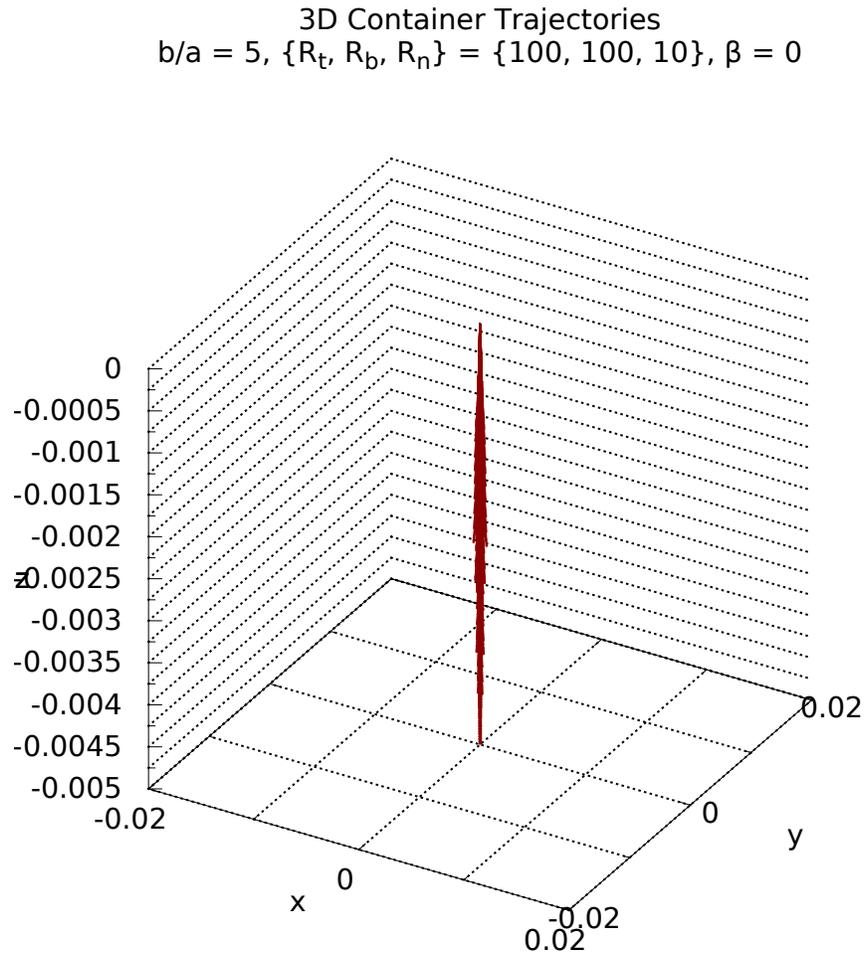


Fig. 5.77 Container trajectories are shown for $\beta = 5$ squirmers with various starting positions (defined by a 2d grid). Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$.

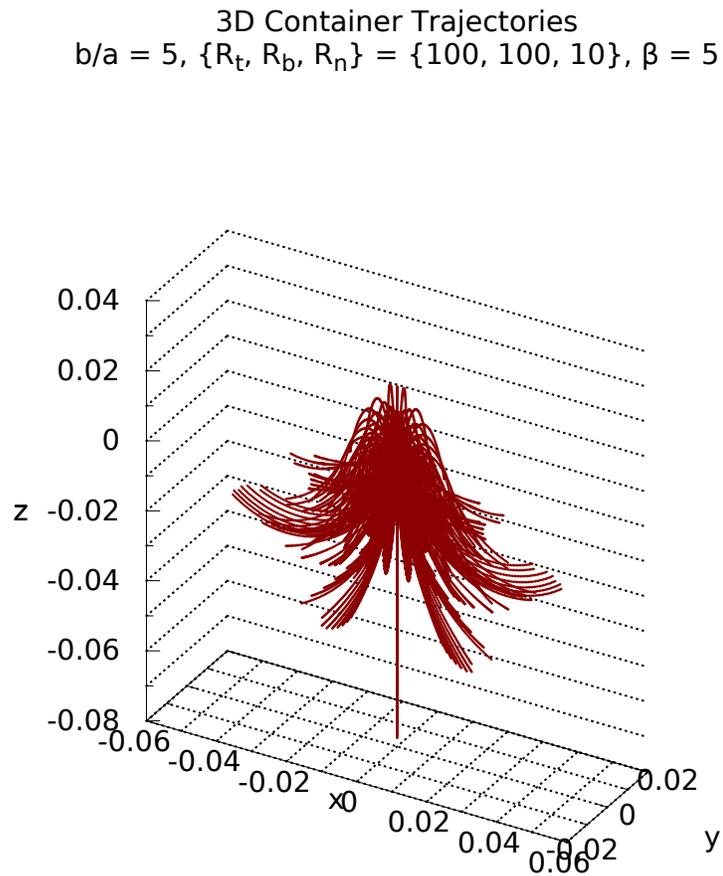


Fig. 5.78 The total and net distance that a $\beta = -5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

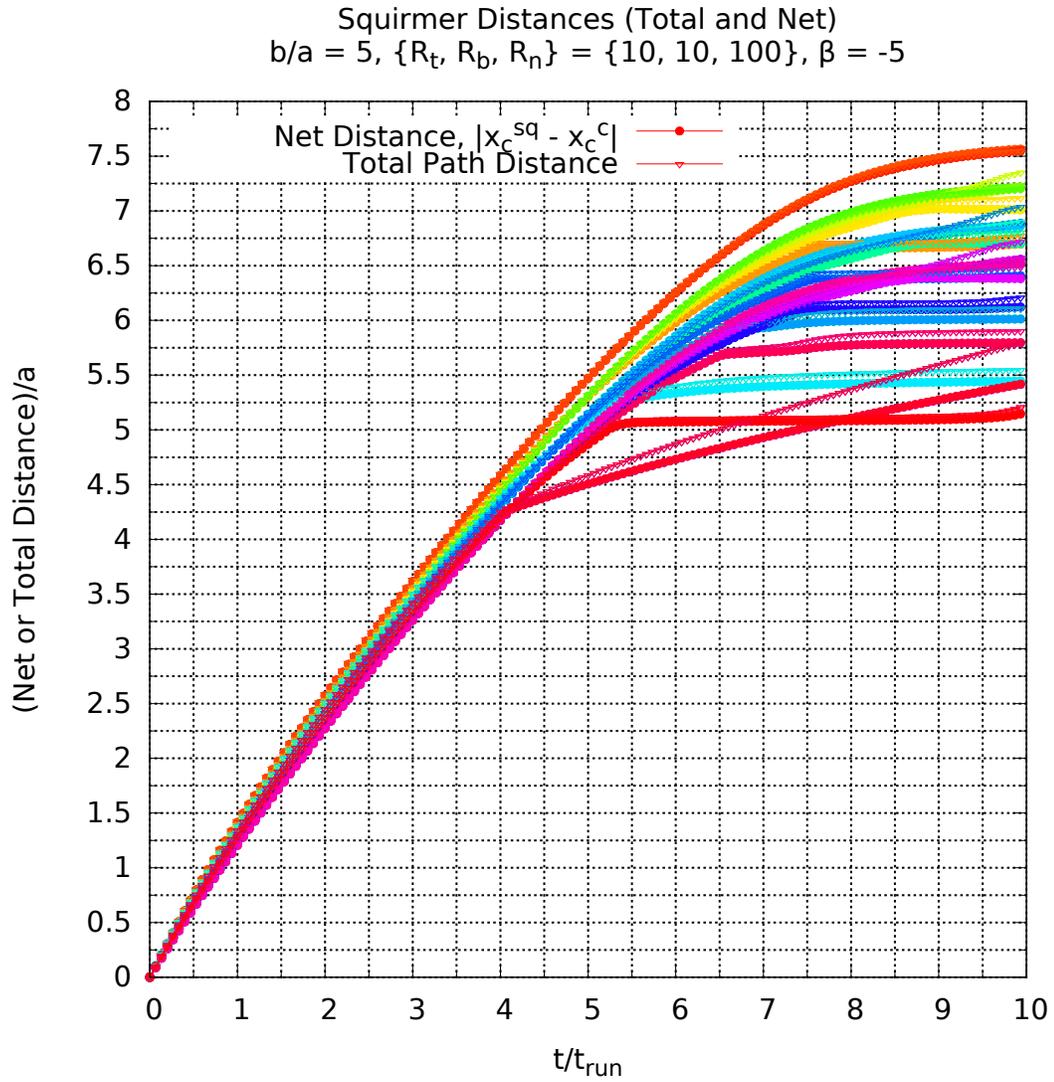


Fig. 5.79 The total and net distance that a $\beta = 0$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

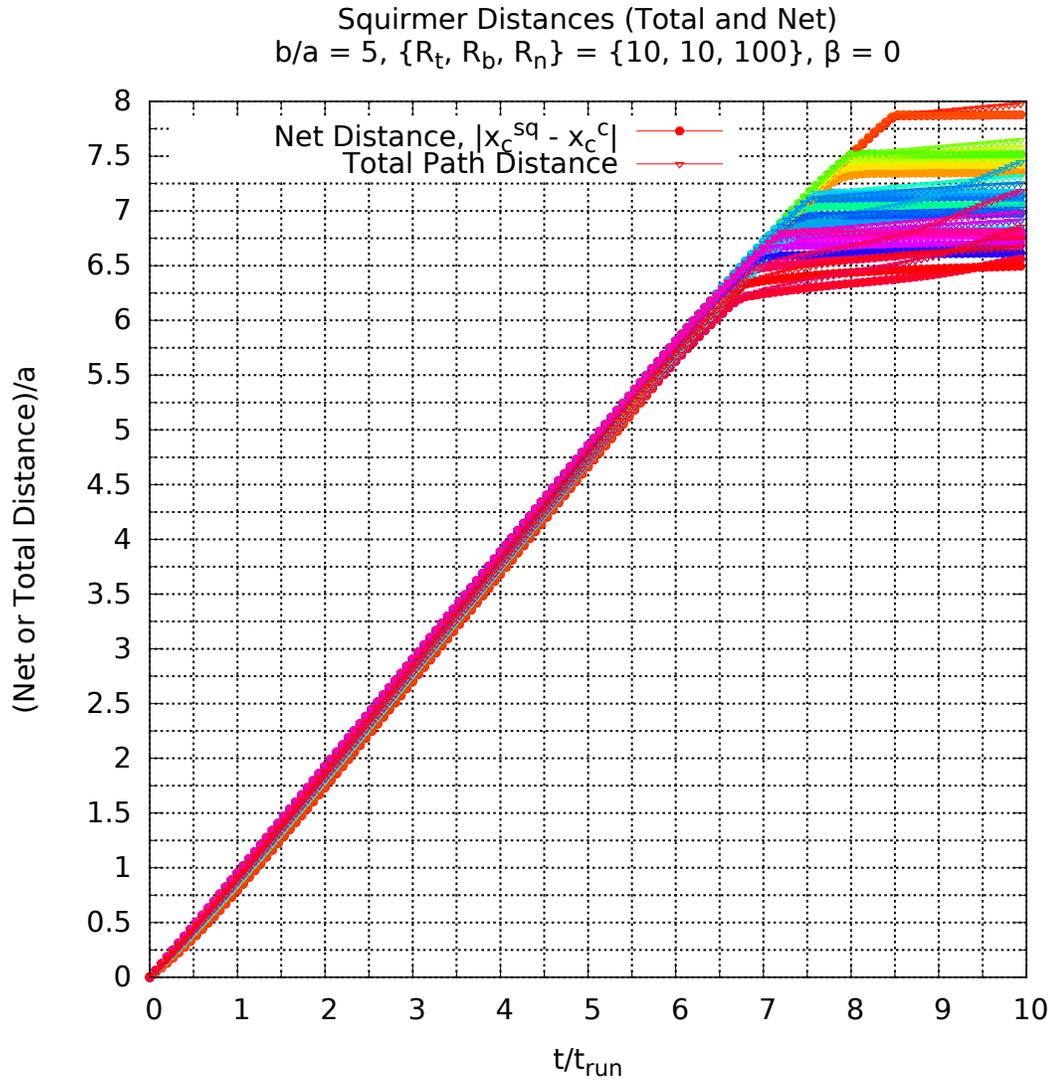


Fig. 5.80 The total and net distance that a $\beta = 5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{10, 10, 100\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

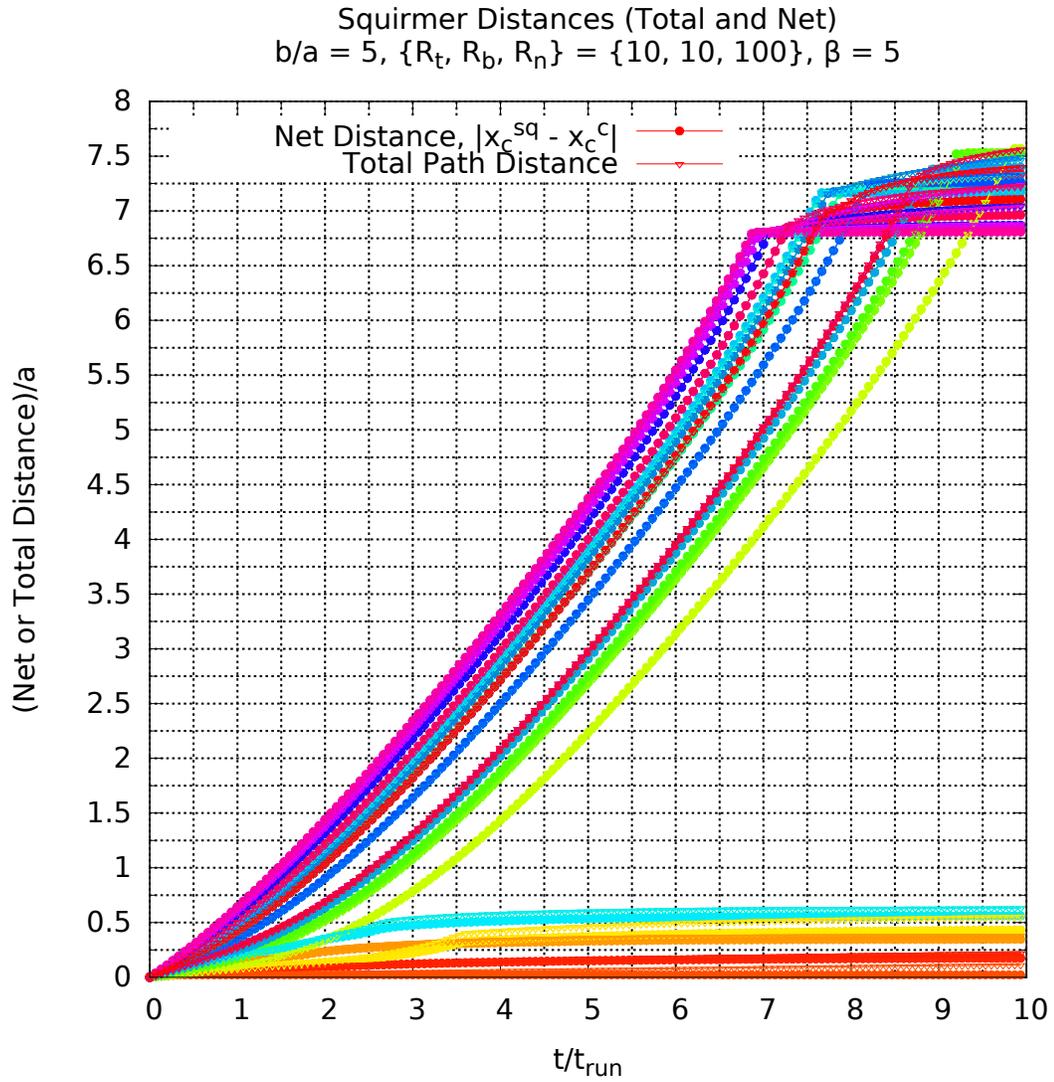


Fig. 5.81 The total and net distance that a $\beta = -5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

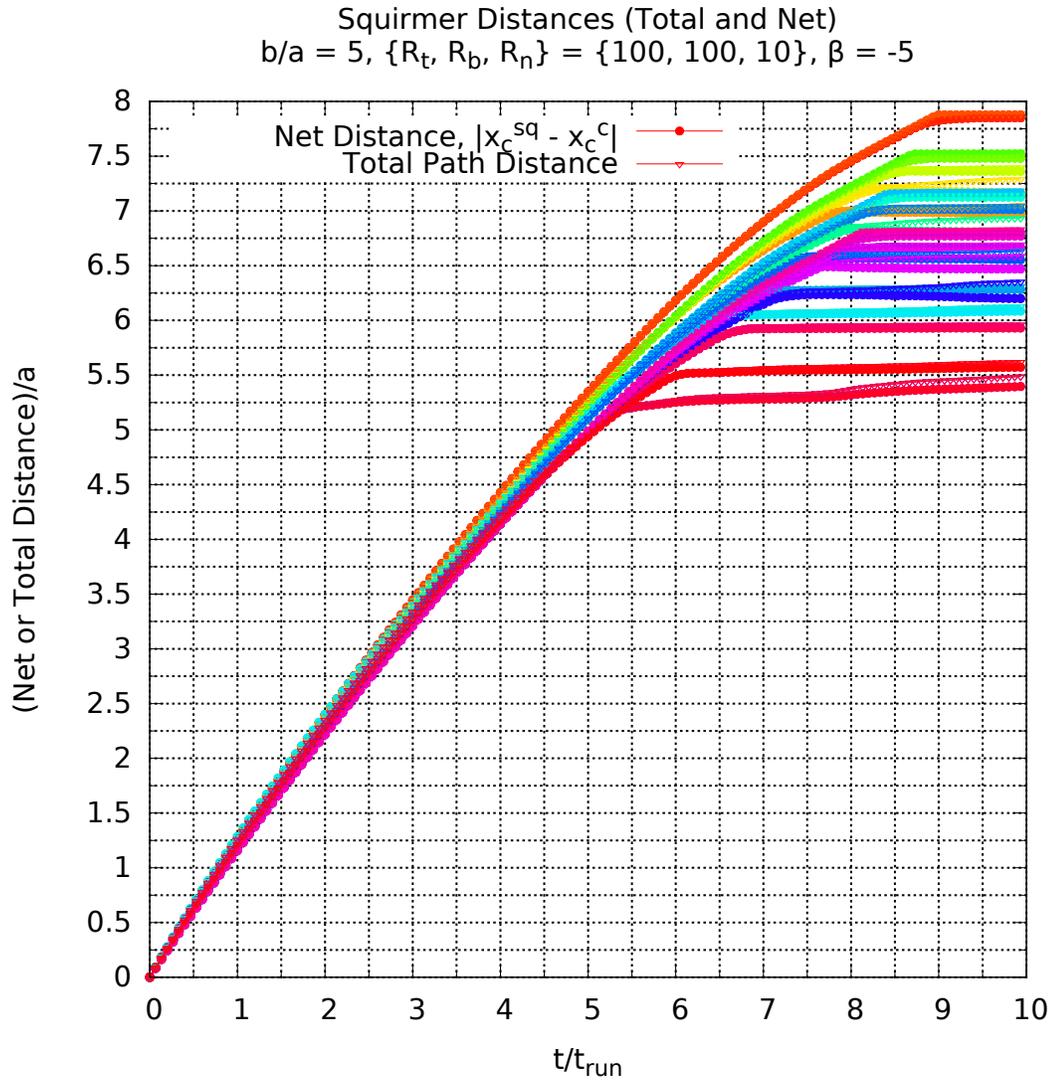


Fig. 5.82 The total and net distance that a $\beta = 0$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $e_s = e_z$.

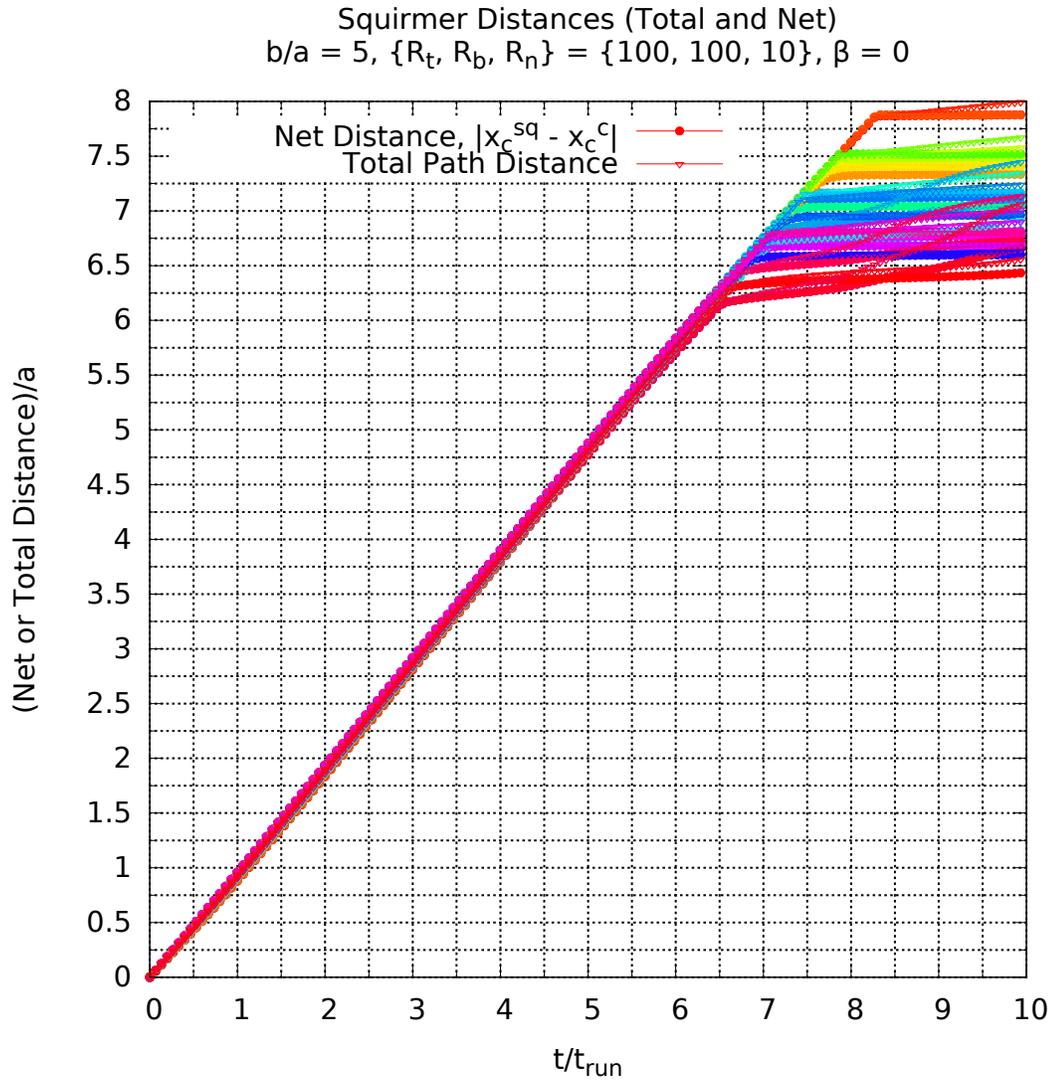


Fig. 5.83 The total and net distance that a $\beta = 5$ squirmer travels is shown for various trajectories. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$. Each squirmer is initialized with $\mathbf{e}_s = \mathbf{e}_z$.

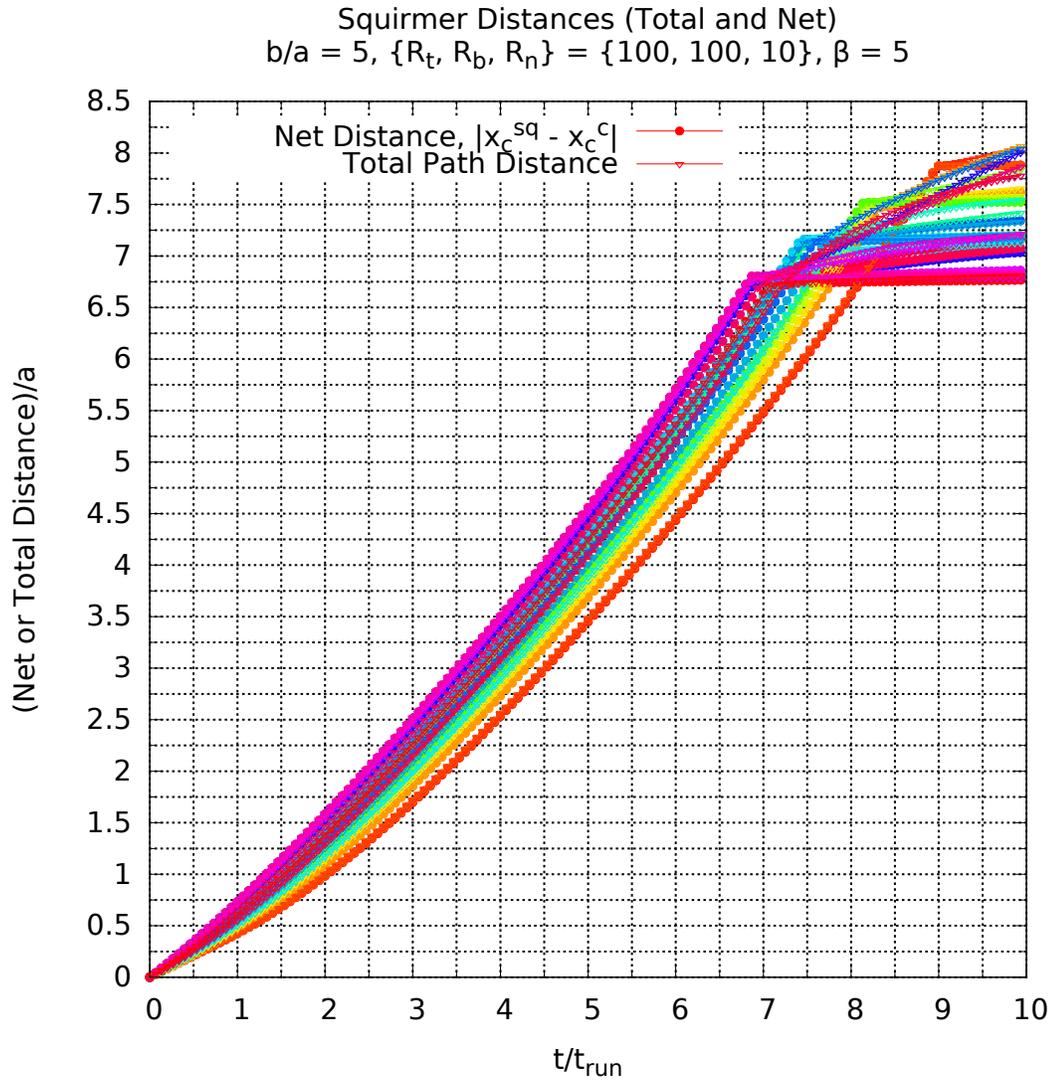


Fig. 5.84 The difference between the net distance that a pusher ($\beta = -5$) and puller ($\beta = +5$) swim is shown as a function of time. In this case the trajectory dynamics are shown under weak tangential resistance to permeable flow. Pushers are seen to move relative distances greater than pullers at short times. At long times, pullers move greater relative distances. Roughly speaking, this is due to the fact that pushers tend to be attracted towards the container walls. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$.

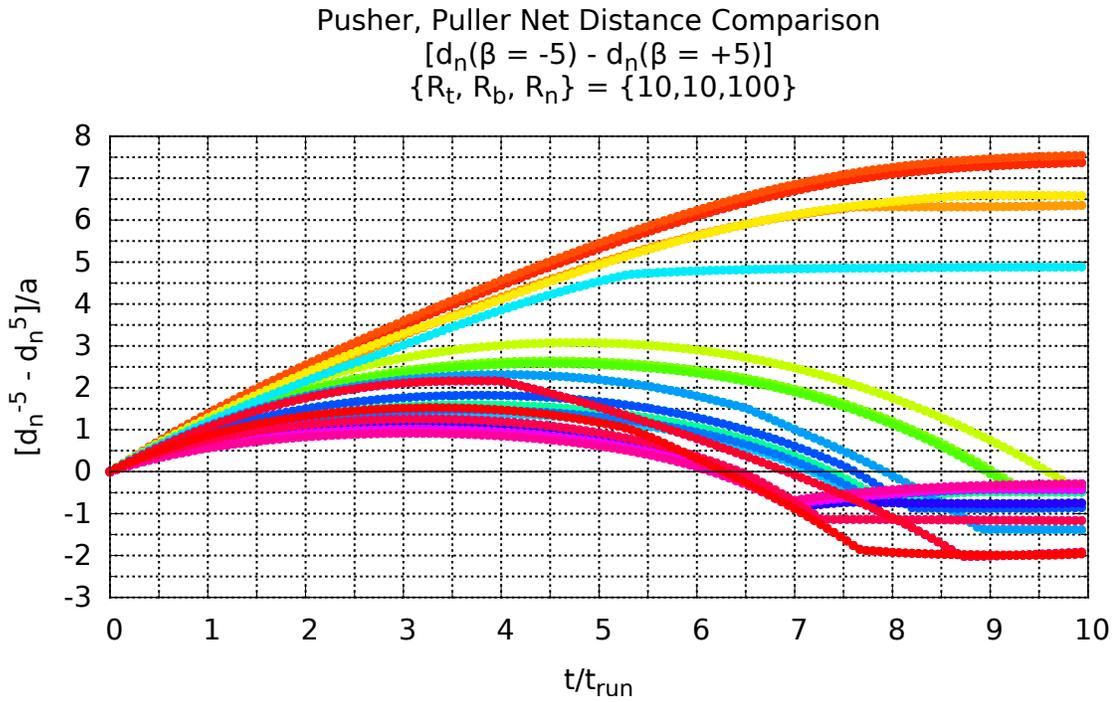


Fig. 5.85 The difference between the net distance that a pusher ($\beta = -5$) and puller ($\beta = +5$) swim is shown as a function of time. In this case the trajectory dynamics are shown under weak normal resistance to permeable flow. Pushers are seen to move relative distances greater than pullers at short times. At long times, pullers move greater relative distances. Roughly speaking, this is due to the fact that pushers tend to be attracted towards the container walls. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$.

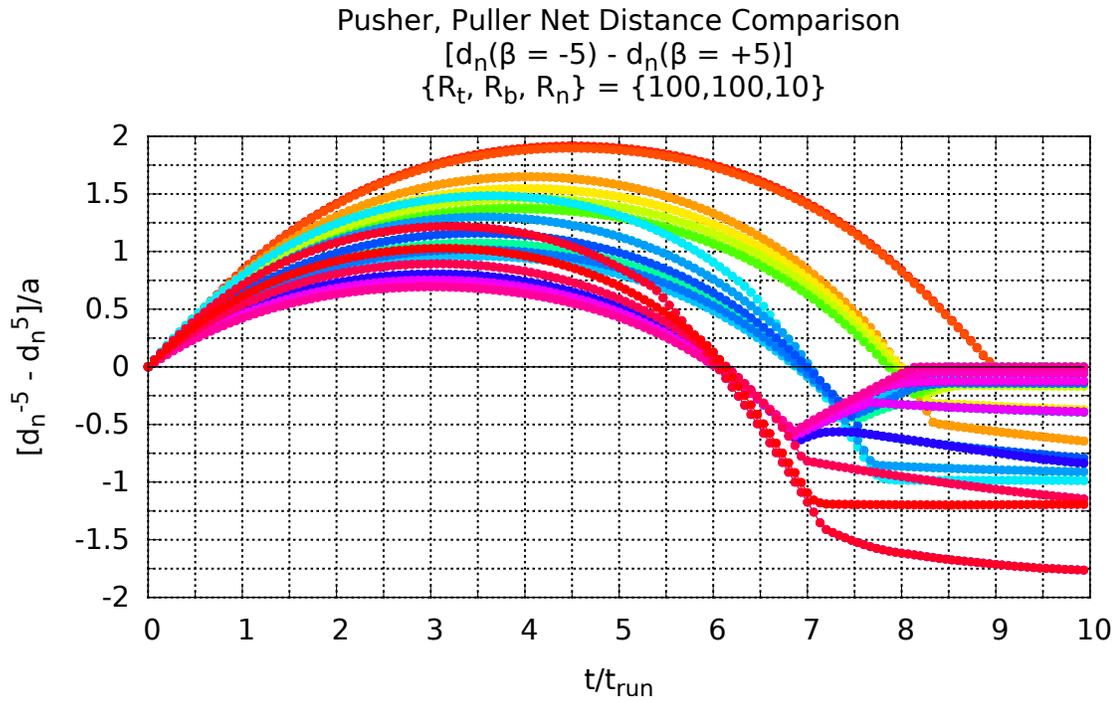


Fig. 5.86 The difference between the net distance that a pusher ($\beta = -5$) swims in the weak tangential and weak normal resistance regimes is plotted as a function of time. Pushers are seen to favor motion in weak tangential resistance at short times and weak normal resistance at long times. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$.

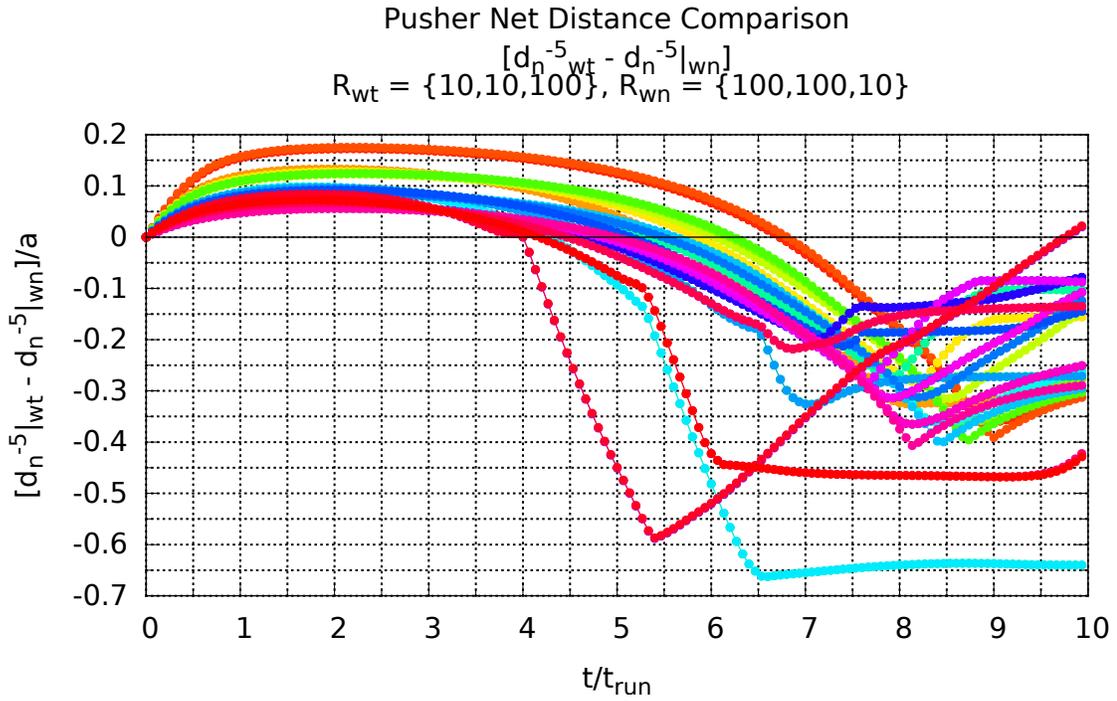


Fig. 5.87 The difference between the net distance that a puller ($\beta = +5$) swims in the weak tangential and weak normal resistance regimes is plotted as a function of time. Pullers are seen to favor motion in weak normal resistance for almost all times. Parameters for this plot are given by a size ratio $b/a = 5$, and permeability resistances $R = \{100, 100, 10\}$.

