An Adaptive Constraint-Based Immersed Body Method for Multiphase Fluid-Structure Interaction

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Engineering Sciences and Applied Mathematics

By

Nishant Nangia

EVANSTON, ILLINOIS

March 2019
An Adaptive Constraint-Based Immersed Body Method for Multiphase Fluid-Structure Interaction

Nishant Nangia

Many industrial fluid flow problems involve the interaction between heavy, rigid objects and one or more fluid phases. For several decades, there has been a vested interest in simulating these fluid-structure interaction (FSI) problems in order to improve engineering design processes. However, numerical simulations of these problems can be challenging and computationally expensive due to the complexity and multiphysics nature of the governing partial differential equations. This thesis presents a robust, adaptive numerical technique for simulating high density ratio and high shear multiphase flows. The method is second-order accurate and stable for density and viscosity ratios of up to one million, and employs the level set approach to resolve topologically complex interfaces. This incompressible flow solver is coupled to a constraint-based immersed body method, enabling efficient simulation of rigid, multiphase fluid-structure interaction. Adaptive mesh refinement is used to deploy fine grid resolution in parts of the domain requiring more accuracy, such as fluid-solid or fluid-gas interfaces and regions of high vorticity. A variety of two-
and three-phase fluid flow problems are simulated to demonstrate the wide applicability of the technique. Additionally, this thesis presents an efficient post-processing approach for computing hydrodynamic forces and torques on immersed bodies and investigates optimality in aquatic locomotion for undulatory swimmers.
Acknowledgements

This dissertation is a collection of work that has been made possible with the support of multiple people over the past few years. I must first and foremost thank my advisor Neelesh A. Patankar, who gave me the opportunity to work on a variety of interesting problems. Prof. Patankar allowed me the freedom to develop my own research interests and projects, while still offering helpful advice during times of frustration. I am forever grateful for his guidance throughout my doctoral studies. I am also indebted to my co-advisor Amneet Pal Singh Bhalla, who was an invaluable mentor and collaborator. Prof. Bhalla seemingly always had the availability and patience to help me with nitty-gritty mathematical and implementation details, and I am grateful to him for hosting my visit to Lawrence Berkeley National Laboratory, which made Chapter 2 of this thesis possible. Through both Prof. Patankar’s and Prof. Bhalla’s guidance, I have acquired valuable technical skills that I will carry with me for the rest of my career. I hope to maintain a collaboration with the Patankar and Bhalla labs for many years to come.

I am also thankful to Boyce E. Griffith for developing the open-source library IBAMR, which has been used extensively in this work. Prof. Griffith imposed high software design standards and offered many helpful suggestions through extensive code reviews. I gained significant experience in object-oriented and C++ programming throughout this work and was able to make meaningful contributions to IBAMR because of his guidance. I would also like to thank Rahul Bale for his helpful discussions on aquatic locomotion,
which made Chapter 4 possible. I am grateful to David L. Chopp and Alvin Bayliss for serving on my thesis committee.

The work presented in this dissertation was made possible by the National Science Foundation Graduate Research Fellowship Program (NSF award DGE-1324585), which allowed me the freedom to pursue my own research direction. Additional support was provided by the National Science Foundation’s SI2 program (NSF awards OAC 1450327 and OAC 1450374). Chapter 4 was made possible by National Science Foundation grants CBET-0828749, CMMI-0941674 and CBET-1066575 (awarded to Neelesh A. Patankar) and Chapter 5 was made possible by the U.S. Department of Energy, Office of Science, ASCR (award number DE-AC02-05CH11231) and NIH HL117163 (awarded to Boyce E. Griffith). Computational resources were provided by Northwestern University’s Quest high performance computing service; additional resources were provided by the Extreme Science and Engineering Discovery Environment (XSEDE) Bridges, at the Pittsburgh Supercomputing Center, and Comet at the San Diego Supercomputer Center through allocation TG-ASC170023, which is supported by National Science Foundation grant number ACI-1548562.

Getting a PhD can be a lonely and discouraging endeavor. Therefore, I would like to thank my friends who have made my time at Northwestern enjoyable: Brennan Sprinkle, Nathan Sanford, Yuxin Chen, and many others. I am also thankful to my past and present labmates, who were always available to go on coffee runs and waste time with me on (mostly non-work related) conversations: Namu Patel, Paul Jones, Wenjun Kou, Shashank Acharya, Tom Zhao, and Sourav Halder. Finally, I would like to thank my family for their ongoing support. My best friend and partner Patricia Do has advised on
many figures and endured countless hours of practice conference presentations; she has rightfully earned her position as “loudest” author. I also thank my younger brother Udit Nangia, and of course Oskee Nangia for loving and believing in me for my entire life. Last but not least, I am forever thankful to my mother Uma Nangia and my father Narinder Nangia who have made many sacrifices to ensure my continual success in life. I could not have made it this far without their support.
Dedication

To my family
# Table of Contents

Abstract 3

Acknowledgements 5

Dedication 8

List of Tables 12

List of Figures 14

Chapter 1. Introduction 44

Chapter 2. A robust incompressible Navier-Stokes solver for high density ratio multiphase flows 48

2.1. Introduction 49

2.2. The continuous equations of motion 55

2.3. The discretized equations of motion 57

2.4. Solution methodology 86

2.5. Software implementation 95

2.6. Solver accuracy and performance 96

2.7. Numerical examples 124

2.8. Conclusions 149
<table>
<thead>
<tr>
<th>Chapter 3. A DLM immersed boundary method based wave-structure interaction</th>
<th>155</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1. Introduction</td>
<td>156</td>
</tr>
<tr>
<td>3.2. The continuous equations of motion</td>
<td>163</td>
</tr>
<tr>
<td>3.3. Spatial discretization</td>
<td>167</td>
</tr>
<tr>
<td>3.4. Solution methodology</td>
<td>174</td>
</tr>
<tr>
<td>3.5. Prescription of solid density, viscosity and a well-balanced gravitational force</td>
<td>187</td>
</tr>
<tr>
<td>3.6. Software implementation</td>
<td>188</td>
</tr>
<tr>
<td>3.7. Wave-structure interaction</td>
<td>188</td>
</tr>
<tr>
<td>3.8. Free-surface piercing and floating structure examples</td>
<td>199</td>
</tr>
<tr>
<td>3.9. Conclusions</td>
<td>227</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 4. Optimal specific wavelength for maximum thrust production in undulatory propulsion</th>
<th>231</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1. Introduction</td>
<td>231</td>
</tr>
<tr>
<td>4.2. Results</td>
<td>233</td>
</tr>
<tr>
<td>4.3. Discussion</td>
<td>249</td>
</tr>
<tr>
<td>4.4. Materials and methods</td>
<td>256</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5. A moving control volume approach to computing hydrodynamic forces and torques on immersed bodies</th>
<th>274</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1. Introduction</td>
<td>275</td>
</tr>
<tr>
<td>5.2. Equations of motion</td>
<td>280</td>
</tr>
<tr>
<td>5.3. Hydrodynamic force and torque</td>
<td>285</td>
</tr>
</tbody>
</table>
5.4. Software implementation 301
5.5. Results 301
Conclusions 331

Chapter 6. Ongoing work and future scope 333
6.1. Unified landscape of optimal aquatic locomotion 333
6.2. Simulation of industrial fluid-structure interaction problems 347

References 359
## List of Tables

2.1 Errors in velocity and pressure after a single time step for the two-dimensional static and inviscid bubble. The exact curvature $\kappa = 1/R$ is used for the surface tension force calculation. 127

2.2 Errors in velocity and pressure after a 50 time steps for the two-dimensional static and inviscid bubble. Exact curvature $\kappa = 1/R$ is used for the surface tension force calculation. 127

2.3 Errors in velocity and pressure after a single time step for the two-dimensional static and inviscid fluid column. 132

2.4 Errors in velocity and pressure after a 50 time steps for the two-dimensional static and inviscid fluid column. 132

3.1 Parameter specification for the three second-order Stokes wave cases considered in Sec. [3.7.2]. 195

3.2 Locations of the pressure and water height probes for the 3D water column impacting a stationary obstacle. Pressure probe locations are given by 3D coordinates while water height probe locations are given by 2D coordinates in the $xy$-plane 227
4.1 Mean specific wavelength, lateral Reynolds number, and aspect ratio for the organisms studied in this work. *Anguilliform swimmers. †Swimmer has a distinct caudal fin. All *Clupea harengus* and *Danio rerio* specimens were larval. Aspect ratio data are measured from side–view images or schematics of swimmer and not related to the listed $N_i$ value.

5.1 Comparison of force measurements for Stokes flow around two concentric spheres.

5.2 Comparison of torque measurements $\mathbf{M} \cdot \mathbf{e}_y$ for Stokes flow around single rotating sphere.
List of Figures

2.1  (a) A staggered-grid discretization of the computational domain $\Omega$. (b) A single Cartesian grid cell on which the components of the velocity field $\mathbf{u}$ are approximated on the cell faces ($\rightarrow$, black); the pressure $p$, and level set $\phi$ are approximated on the cell center (●, black); and the interpolated viscosity is approximated on cell nodes (■, green).

2.2  The shifted control volume (---, green) over which an approximation to $(\nabla \cdot \mathbf{u} \psi)_i - \frac{1}{2},j$ is computed. (a) The staggered grid velocities used to compute the advection velocity $\mathbf{u}_{\text{adv}}$. (b) The location of the advection velocity on faces (×, green) of the shifted control volume.

2.3  The shifted control volume (---, green) over which an approximation to $(\nabla \cdot \mathbf{u} \psi)_i - \frac{1}{2},j$ is computed. In each diagram, the control volume is surrounding the same face-centered degree of freedom located at $x_{i-\frac{1}{2},j}$ and the upwind $\psi_C$, far upwind $\psi_U$, and downwind $\psi_D$ are labeled according to their usage in Eq. (2.23). (a) The required degrees of freedom to compute the cubic upwind interpolation of $\psi_e$ when $u_e \geq 0$. (b) The required degrees of freedom to compute the cubic upwind interpolation of $\psi_e$ when $u_e < 0$. (c) The required degrees of freedom to compute the cubic upwind interpolation of $\psi_n$ when $v_n \geq 0$. (d) The
required degrees of freedom to compute the cubic upwind interpolation
of $\psi_n$ when $v_n < 0$.

2.4 Two adjacent 3D staggered grid cells on which the viscous component
$(L_\mu u)^x_{i-\frac{1}{2},j,k}$ is computed on a particular face ($\times$, red) using velocity
derivatives and viscosity on cell centers ($\bullet$, black), $xy$-edges ($\triangledown$, orange),
and $xz$-edges ($\triangledown$, green). The $yz$-edges ($\diamondsuit$, blue) are not used in
the computation of $(L_\mu u)^z_{i-\frac{1}{2},j,k}$, but would be required to compute
$(L_\mu u)^y_{i,j-\frac{1}{2},k}$ and $(L_\mu u)^z_{i,j,k-\frac{1}{2}}$.

2.5 The velocity, pressure, and viscosity degrees of freedom in the vicinity of
cell $(N-1,j)$ required to enforce physical boundary conditions. Interior
and boundary values are colored in black, and ghost values are colored
in grey.

2.6 The [a] density and [b] viscosity fields, along with the initial velocity
vectors for the manufactured solution described in Sec. 2.6.1.

2.7 $L^1$ ($\blacksquare$, blue) and $L^\infty$ ($\bullet$, red) errors as a function of grid size $N$ for
the non-conservative manufactured solution with periodic boundary
conditions: (a) convergence rate for $u$; (b) convergence rate for $p$.

2.8 $L^1$ ($\blacksquare$, blue) and $L^\infty$ ($\bullet$, red) errors as a function of grid size $N$ for
the non-conservative manufactured solution with specified normal and
tangential velocity (vel-vel) boundary conditions: (a) convergence rate
for $u$; (b) convergence rate for $p$. 
2.9 \( L^1 (■, \text{blue}) \) and \( L^\infty (●, \text{red}) \) errors as a function of grid size \( N \) for the non-conservative manufactured solution with specified normal velocity and tangential traction (vel-tra) boundary conditions: (a) convergence rate for \( u \); (b) convergence rate for \( p \).

2.10 \( L^1 (■, \text{blue}) \) and \( L^\infty (●, \text{red}) \) errors as a function of grid size \( N \) for the non-conservative manufactured solution with specified normal traction and tangential velocity (tra-vel) boundary conditions: (a) convergence rate for \( u \); (b) convergence rate for \( p \).

2.11 \( L^1 (■, \text{blue}) \) and \( L^\infty (●, \text{red}) \) errors as a function of grid size \( N \) for the non-conservative manufactured solution with specified normal and tangential traction (tra-tra) boundary conditions: (a) convergence rate for \( u \); (b) convergence rate for \( p \).

2.12 The (a) density and (b) viscosity fields, along with the initial velocity vectors for the manufactured solution described in Sec. 2.6.2.

2.13 Errors as a function of grid size \( N \) for varying density ratios \( \rho_R = 10^1 - 10^6 \) with \( \rho_0 = 10^0 \) and constant viscosity ratio \( \mu_R = 10^1 \) with \( \mu_0 = 10^{-4} \) for a non-conservative manufactured solution. Normal velocity and tangential traction boundary conditions are specified at all physical boundaries in every case. (a) \( L^1 \) convergence rate for \( u \); (b) \( L^\infty \) convergence rate for \( u \); (c) \( L^1 \) convergence rate for \( p \); (d) \( L^\infty \) convergence rate for \( p \).
2.14 Errors as a function of grid size $N$ for varying viscosity ratios

$\mu_R = 10^4 - 10^6$ with $\mu_0 = 10^{-4}$ and constant density ratio $\rho_R = 10^4$

with $\rho_0 = 10^9$ for a non-conservative manufactured solution. Normal velocity and tangential traction boundary conditions are specified at all physical boundaries in every case. (a) $L^1$ convergence rate for $u$; (b) $L^\infty$ convergence rate for $u$; (c) $L^1$ convergence rate for $p$; (d) $L^\infty$ convergence rate for $p$. 107

2.15 The (a) initial density and (b) viscosity fields, along with the initial velocity vectors for the manufactured solution described in Sec. 2.6.3. 108

2.16 $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the conservative manufactured solution with specified normal velocity and tangential traction boundary conditions on all boundaries. In these cases, the density $\rho$ is not reset between time steps. (a) convergence rate for $u$; (b) convergence rate for $p$; (c) convergence rate for $\rho$. 109

2.17 $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the conservative manufactured solution with specified normal velocity and tangential traction boundary conditions on all boundaries. In these cases, the density $\rho$ is reset between time steps. (a) convergence rate for $u$; (b) convergence rate for $p$; (c) convergence rate for $\rho$. 112

2.18 Velocity vectors and pressure field plotted on the grid hierarchy used for the locally refined mesh convergence study. In both figures, the purple box indicates a locally refined grid with grid spacing $\Delta x_{\text{min}} = \Delta x_0/4$. 113
2.19 \( L^1 \) (■, blue) and \( L^\infty \) (●, red) errors as a function of coarsest grid spacing \( \Delta x_0 \) for the non-conservative manufactured solution described in Sec. 2.6.1 on a locally refined mesh. The specified boundary conditions are normal velocity and tangential traction (vel-tra). (a) convergence rate for \( u \); (b) convergence rate for \( p \).

2.20 \( L^1 \) (■, blue) and \( L^\infty \) (●, red) errors as a function of coarsest grid spacing \( \Delta x_0 \) for the conservative manufactured solution with specified normal velocity and tangential traction boundary conditions on all boundaries. In these cases, the density \( \rho \) is not reset between time steps. (a) convergence rate for \( u \); (b) convergence rate for \( p \); (c) convergence rate for \( \rho \).

2.21 The initial location of the air-suspended water droplet, and the initial velocity vectors for the problem set up described in Sec. 2.6.5.

2.22 Relative residual as a function of iteration count for the final solves of the water bubble problem described in Sec. 2.6.5, for varying grid sizes \( N = 32 \rightarrow 1024 \). MKS units are used to specify the material properties of both water and air, yielding a Reynolds number of \( Re = 2.71 \times 10^4 \).

(a) Velocity subdomain solver relative residual \( \| A \hat{x}_u - b_u \|_2 / \| b_u \|_2 \) vs. number of iterations. (b) Pressure subdomain solver relative residual \( \| -L_v \theta + \frac{1}{\Delta t} (b_p + D \cdot \hat{x}_u) \|_2 / \| -\frac{1}{\Delta t} (b_p + D \cdot \hat{x}_u) \|_2 \) vs. number of iterations. (c) Staggered Stokes solver relative residual \( \| Bx - b \|_2 / \| b \|_2 \) vs. number of iterations.
2.23 Relative residual as a function of iteration count for the final solves of
the water bubble problem described in Sec. 2.6.5, for varying grid sizes
$N = 32 - 1024$. CGS units are used to specify the material properties
of both water and air, yielding a Reynolds number of $Re = 2.71 \times 10^3$.

(a) Velocity subdomain solver relative residual $\|A\hat{x}_u - b_u\|_2 / \|b_u\|_2$ vs.
number of iterations. (b) Pressure subdomain solver relative residual
$\| - L_v \theta + \frac{1}{\Delta t} (b_p + D \cdot \hat{x}_u) \|_2 / \| - \frac{1}{\Delta t} (b_p + D \cdot \hat{x}_u) \|_2$ vs. number of
iterations. (c) Staggered Stokes solver relative residual $\|Bx - b\|_2 / \|b\|_2$
vs. number of iterations.

2.24 $L^1 (■, blue)$ and $L^\infty (●, red)$ errors as a function of grid size $N$ for
the conservative manufactured solution with specified normal velocity
and normal velocity boundary conditions on all boundaries, and smooth
density profile given by Eq. (2.113). In these cases, the density $\rho$ is not
reset between time steps. (a) convergence rate for $u$; (b) convergence
rate for $p$; (c) convergence rate for $\rho$.

2.25 $L^1 (■, blue)$ and $L^\infty (●, red)$ errors as a function of grid size $N$ for
the conservative manufactured solution with specified normal velocity
and normal velocity boundary conditions on all boundaries, and smooth
density profile given by Eq. (2.113). In these cases, the density $\rho$ is not
reset between time steps. Forward Euler timestepping is used to update
$\rho$, while an upwind scheme is used to approximate $\rho^{lim}$ on the faces of
the shifted control volumes. (a) convergence rate for $u$; (b) convergence
rate for $p$; (c) convergence rate for $\rho$. 

2.26 $L^1$ (□, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the 3D non-conservative manufactured solution with specified normal velocity and tangential traction (vel-tra) boundary conditions: (a) convergence rate for $\mathbf{u}$; (b) convergence rate for $p$.

2.27 Temporal evolution of errors for a viscous bubble with $\rho_i/\rho_o = 10^3$ and $La = 12000$, for grid sizes $N = 40$ – 320. Here, curvature is computed numerically from the level set function. (a) Pressure difference relative error $E(\Delta p)$ vs. nondimensional time. (b) $L^1$ norm of velocity vs. nondimensional time. (c) $L^\infty$ norm of velocity vs. nondimensional time.

2.28 Velocity vectors and pressure of the static bubble with $\rho_i/\rho_o = 10^3$ and $La = 12000$, for grid size $N = 320$ at time $t/t_\sigma = 60.41$. Curvature is computed numerically from the level set function. 130

2.29 Temporal evolution of a tank being filled with heavy liquid for $\rho_h/\rho_l = 10^3$ and $N = 100$. Velocity vectors are denoted by arrows ($\rightarrow$, black).

2.30 Initial condition/exact solution (––––, red) and numerical solution (—, black) for convection of a high density droplet with density ratio $\rho_i/\rho_o = 10^6$ and initial velocity $\mathbf{u}_p(x,0)$. Inconsistent transport of mass and momentum occurs while employing a non-conservative momentum integrator. PPM is used for advection of the level set and CUI limiter is used for the convective operator. (a) Grid size $N = 32$ at time $t = 1$; (b) Grid size $N = 64$ at time $t = 1$; (c) Grid size $N = 128$ at time $t = 0.5$;
the simulation becomes unstable shortly after the shown time (d) Grid size $N = 256$ at time $t = 0.5$; the simulation becomes unstable shortly after the shown time

2.31 Initial condition/exact solution (---, red) and numerical solution (—, black) for convection of a high density droplet with density ratio $\rho_i/\rho_o = 10^6$ and initial velocity $\mathbf{u}_o(x,0)$. Consistent transport of mass and momentum is used for these cases. PPM is used for advection of the level set and CUI limiters are used for mass and convective fluxes. (a) Grid size $N = 32$ at time $t = 1$. (b) Grid size $N = 64$ at time $t = 1$. (c) Grid size $N = 128$ at time $t = 1$. (d) Grid size $N = 256$ at time $t = 1$. 

2.32 Initial condition/exact solution (---, red) and numerical solution (—, black) for convection of a high density droplet with density ratio $\rho_i/\rho_o = 10^6$ and initial velocity $\mathbf{u}_o(x,0)$. Consistent transport of mass and momentum is used for these cases. PPM is used for advection of the level set and various limiters are used for mass and convective fluxes. The grid size is $N = 256$ for all cases. (a) CUI limiter at time $t = 1$. (b) M-Gamma limiter at time $t = 1$. (c) FBICS limiter at time $t = 1$. (d) Minimum density in the domain during the first eight simulation time steps.

2.33 Three-dimensional convection of a high density ratio droplet (blue) with density ratio $\rho_i/\rho_o = 10^6$ and initial velocity $\mathbf{u}_o(x,0)$. The coarse grid has grid spacing $\Delta x_0 = \Delta y_0 = \Delta y_0 = 1/64$, with each simulation using $\ell = 2$ grid levels and refinement ratio $n_{ref} = 2$. The coarse grid level
is bounded by the red boxes, while the fine grid level is bounded by green boxes. (a)-(c) Three snapshots in time for inconsistent transport of mass and momentum, which becomes unstable shortly after $t = 0.6$.

(d)-(f) Three snapshots in time for inconsistent transport of mass and momentum, which remains stable throughout the simulation time.

2.34 Two dimensional evolution of the spreading water column with density ratio $\rho_l/\rho_g = 815.66$ and $\mu_l/\mu_g = 63.88$ at four different time instances: (left) Inconsistent and (right) consistent transport of mass and momentum is used with grid size $512 \times 128$.

2.35 Three dimensional evolution of the spreading water column with density ratio $\rho_l/\rho_g = 815.66$ and $\mu_l/\mu_g = 63.88$ at four different time instances. Consistent transport of mass and momentum is used for this case with grid size $256 \times 36 \times 64$.

2.36 Temporal evolution of (a) dimensionless front position and (b) dimensionless height for the present 2D (—, black) and 3D (×, red) consistent transport simulations, along with experimental data (●, blue) from Martin and Moyce [83], 3D simulation data (■, green) from Gu et al. [82], 2D simulation data (♦, grey) from Rezende et al. [81], and 2D simulation data (▼, yellow) from Patel and Natarajan [44].

2.37 Temporal evolution of the droplet splashing on a thin liquid film with density ratio $\rho_l/\rho_g = 815$ and $\mu_l/\mu_g = 55$ at five different time instances with grid size $1600 \times 400$: (left) $Re = 6.6$ and (right) $Re = 66$. 
2.38 (a) Location of the jet base $x_j$ (●, red), which is the midpoint of the two neck points (∗, red) for the liquid sheet ($T = 0.2, \text{Re} = 66$); (b) Temporal evolution of the dimensionless jet based location.

2.39 (a)-(c) Temporal evolution of the 3D droplet splashing on a thin liquid film with density ratio $\rho_l/\rho_g = 1000$ and $\mu_l/\mu_g = 40$ at three different time instances. Inconsistent transport of mass and momentum occurs while employing a non-conservative momentum integrator. Contrast these results to those shown in Fig. 2.40, in which consistent transport is used.

2.40 (a)-(e) Temporal evolution of the 3D droplet splashing on a thin liquid film with density ratio $\rho_l/\rho_g = 1000$ and $\mu_l/\mu_g = 40$ at five different time instances. (f) Locations of the different refined mesh levels from coarsest to finest: red, green, pink, orange. Consistent transport of mass and momentum is used for these cases. Unlike the results shown in Fig. 2.39, here we observe a physical solution.

2.41 Temporal evolution of (a) dimensionless bottom radius and (b) dimensionless lamella height for the present 3D (●, red) simulations, along with 3D simulation data (■, blue) from Nikolopoulos et al. 2005 [92], 3D simulation data (▲, green) from Nikolopoulos et al. 2007 [92], and 3D simulation data (♦, grey) from Manik et al. [94].

3.1 (a) Sketch of the immersed structure contained within a domain containing liquid and gas phases. (b) Numerical discretization of the
domain $\Omega$ into Eulerian grid cells (■, purple) and Lagrangian markers (■, orange). (c) A single Cartesian grid cell on which the components of the velocity field $\mathbf{u}$ are approximated on the cell faces (→, black); the pressure $p$ and level sets $\phi$ and $\psi$ are approximated on the cell center (●, black); and the Lagrangian quantities are approximated on the marker point (■, orange), which can be arbitrarily placed on the Eulerian grid. 170

3.2 Sketch of the two-stage process for setting the density and viscosity in the computational domain. (a) Material properties are first prescribed in the “flowing” phase based on the liquid-gas level set function $\phi$ (—, black) and ignoring the structure level set function $\psi$ (---, orange). (b) Material properties are then corrected in the phase occupied by the immersed body. 176

3.3 Convergence study for the temporal evolution of wave elevation at $x = 2.87\lambda$ for a second-order Stokes wave; (—, black) analytical expression given by Eq. 3.52; (..., blue) present simulation for a 442 × 66 grid; (···, green) present simulation for a 884 × 132 grid; (---, red) present simulation for a 1768 × 264 grid. 194

3.4 (a) Locations of (●, red) Case A, (■, yellow) Case B, and (▲, green) Case C on a phase diagram denoting the applicability of wave theories described by Le Méhauté [129]; figure adapted from Holthuijsen [130]. Long-time temporal evolution of wave elevation for second-order Stokes waves with parameters described by (b) Case A, (c) Case B, and
3.5 **A:** Temporal evolution of unobstructed second-order Stokes waves at six different time instances. **B:** Temporal evolution of second-order Stokes waves interacting with a stationary trapezoidal obstacle at six different time instances.

3.6 Sketch of the problem set up for an incoming second-order Stokes wave interacting with a trapezoidal structure; (blue) initially quiescent water of depth $d$; (orange) stationary trapezoidal obstacle; (grey) region over which velocities and level set values are damped; (red) locations of the water elevation measurement stations. Diagram is not to scale.

3.7 Temporal evolution of wave elevation measured at stations (a) S2, (b) S3, (c) S4, (d) S5, (e) S6, and (f) S7, for a 2D second-order Stokes wave interacting with a stationary trapezoidal obstacle (see Fig. 3.6); (•, yellow) experimental data from Beji and Battjes [131, 132]; (---, red) simulation data from Kasem and Sasaki [126]; (—, black) present simulation data.

3.8 Temporal evolution of a cylinder free-falling into a column containing two fluids at six different time instances with 64 CPD.

3.9 Temporal evolution of (a) vertical position and (b) vertical velocity for a 2D cylinder free-falling into a column containing two fluids. (•, yellow) simulation data from Ghasemi et al. [50]; (—, red) present simulation data.
data with 32 grid cells per diameter; (---, black) present simulation data for 64 grid cells per diameter.

3.10 Temporal evolution of a cylinder heaving on an air-water interface at four different time instances with 46 CPR: (left) density and (right) vorticity generated in the range of $-25$ to $25$.

3.11 Temporal evolution of dimensionless vertical position for a 2D cylinder heaving on an air-water interface. (●, yellow) experimental data from Itô [133]; (···, blue) present simulation data with 11 grid cells per radius; (---, green) present simulation data for 23 grid cells per radius; (——, black) present simulation data with 46 grid cells per radius.

3.12 Temporal evolution of a sphere heaving on an air-water interface at two different time instances with 40 CPR: (top) density and (bottom) locations of the different refined mesh levels from coarsest to finest: red, green, pink, orange.

3.13 Temporal evolution of dimensionless vertical position for a 3D sphere heaving on an air-water interface. (●, yellow) experimental data from Beck and Liapis [134]; (---, red) present simulation data with 20 grid cells per radius; (——, black) present simulation data for 40 grid cells per radius.

3.14 (a) Initial problem set up for a stationary cylinder (orange) placed on a quiescent air-water interface. (b) Temporal evolution of $||u||_\infty$ with the (●, red) full gravitational forcing, and the ( ■, blue) flow gravitational
forcing. Velocity vectors and gravitational forcing for (c) $\rho g$, and (d) $\rho^{\text{flow}} g$ at $t = 1.0$. The scale for velocity vectors is identical in both figures.

3.15 Temporal evolution of a cylinder entering an air-water interface at four different time instances. Inconsistent transport of mass and momentum is used for these cases. The simulation becomes unstable shortly after $t = 0.25$. The plotted vorticity is in the range $-50$ to $50$. Locations of the different refined mesh levels from coarsest to finest are shown in purple and green.

3.16 Temporal evolution of a cylinder entering an air-water interface at four different time instances. Consistent transport of mass and momentum is used for these cases. The plotted vorticity is in the range $-50$ to $50$. Locations of the different refined mesh levels from coarsest to finest are shown in purple and green.

3.17 Dimensionless slamming coefficient as a function of penetration depth for a 2D cylinder entering an air-water interface at constant velocity. (●, yellow) experimental data from Campbell and Weynberg [136]; (—, black) analytical formula from von Kàrmàn [135]; (---, green) simulation data from Patel and Natarajan [119]; (--) simulation data from Zhang et al. [137] (..., purple) simulation data from Kleefsman et al. [138]; (-●-, red) present simulation data.
3.18 Temporal evolution of a rectangular barge floating an air-water interface at six different time instances with 40 CPW. The plotted vorticity is in the range $-50$ to $50$.

3.19 Temporal evolution of the inclination angle for a 2D barge floating on an air-water interface. (●, yellow) simulation data from Patel and Natarajan [44]; (—, black) present simulation data for 40 grid cells per width.

3.20 Temporal evolution of a 2D wedge free-falling into an air-water interface at four different time instances. Inconsistent transport of mass and momentum is used for these cases. The simulation becomes unstable shortly after $t = 0.785$. The plotted vorticity is in the range $-300$ to $300$.

3.21 Temporal evolution of a 2D wedge free-falling into an air-water interface at four different time instances. Consistent transport of mass and momentum is used for these cases. The plotted vorticity is in the range $-300$ to $300$.

3.22 Locations of the different refined mesh levels from coarsest to finest for the 3D free-falling wedge simulation. The coarse mesh is outlined by red boxes while the fine mesh is outlined by green boxes.

3.23 Temporal evolution of a 3D wedge free-falling into an air-water interface at four different time instances.
3.24 Temporal evolution of (a) vertical position and (b) vertical velocity for a 3D wedge free-falling into an air-water interface. (●, yellow) experimental data from Yettou et al. [139]; (■, green) simulation data from Pathak and Raessi [51]; (—, red) present simulation data.

3.25 Sketches of the initial problem set up for a water column impacting a stationary rectangular object; (a) 3D view specifying the size of the domain and the locations of the pressure sensors P1, P3, P5, and P7 (×, black) on the surface of the body; (b) top view and (c) side view indicating the dimensions and locations of the water column (blue) and the obstacle (orange), along with the locations of the water height probes H2 and H4.

3.26 Temporal evolution of a water column impacting a rectangular object at eight different time instances.

3.27 Temporal evolution of pressure measured at probes (a) P1, (b) P3, (c) P5, and (d) P7, for a 3D water column impacting a stationary obstacle (see Fig. 3.25 and Table 3.2); (●, yellow) experimental data from MARIN; (---, red) simulation data from Kleefsman et al. [138]; (--, blue) simulation data from Pathak and Raessi [51]; (—, black) present simulation data.

3.28 Temporal evolution of water height measured at probes (a) H2, and (b) H4 for a 3D water column impacting a stationary obstacle (see Fig. 3.25 and Table 3.2); (●, yellow) experimental data from MARIN; (---, red)
simulation data from Kleefsman et al. [138]; (—,, blue) simulation data from Pathak and Raessi [51]; (—, black) present simulation data.

4.1 Observed St and SW ranges from meta–analysis data of body/caudal fin swimmers. The Strouhal number (top) and specific wavelength (bottom) for the species considered in this study. Data points represent average values, $\bar{x}_i$, where $x \in \{St, SW\}$, of individual species and error bars indicate $\pm$ one s.d. from the species mean. Dashed lines represent $St = 0.3$ and $SW = 10$. Distinction is made between anguilliform and non–anguilliform swimmers. Data labeled as *Ambystoma mexicanum* and *Ambystoma mexicanum young* are from adult and larval axolotl, respectively. All observations of *Clupea harengus* and *Danio rerio* came from anguilliform larvae. For some species, the error bars are not visible at this scale, while for others, only one observation is recorded and no error bars are available; see Table 4.1 for more details.

4.2 Variability in St as a function of $Re_{lat}$ and AR. Intraspecies mean Strouhal number vs. (a) lateral Reynolds number, and (b) aspect ratio for observed non-anguilliform(●) and anguilliform(▲) swimmers. Of these, orange (green) points represent swimmers with (without) well–defined caudal fins. Error bars indicate $\pm$ one s.d. from the species mean. For some species, the error bars are not visible at this scale, while for others, only one observation is recorded and no error bars are available; see Table 4.1 for more details.
4.3 Variability in SW and OSW as a function of $Re_{\text{lat}}$ and AR. Intraspecies mean specific wavelength vs. (a) lateral Reynolds number, (b) aspect ratio for observed non-anguilliform (●) and anguilliform (▲) swimmers. Of these, orange (green) points represent swimmers with (without) well–defined caudal fins. Red crosses (X) represent the optimal specific wavelength for simulations done in the present study. Blue asterisks (∗) represent the optimal specific wavelengths for robotic undulating sheets reported in [155, 156]. Error bars indicate ± one s.d. from the species mean. For some species, the error bars are not visible at this scale, while for others, only one observation is recorded and no error bars are available; see Table 4.1 for more details.

4.4 Measured swimming speed and force from undulating sheet simulations. (a) Axial swimming speed and propulsive force computed from free–swimming (green) and translation–locked (black) simulations of rectangular sheets plotted against the specific wavelength. In free–swimming simulations, the forward swimming speed of the undulating plate was an output parameter of the simulation. Simulations were carried out at a lateral Reynolds number $Re_{\text{lat}} = 4.49$, with corresponding swimming–speed Reynolds number range $1.8 \times 10^4 < Re < 1.51 \times 10^2$. (b) Evolution of axial $F_x$ and heave forces $F_z$ over time for a translation–locked, undulating sheet simulation with $SW = 13.33$. The oscillation in $F_z$ is a signature of the linear recoil effect on the swimmer. The sway force $F_y$ is not shown because the kinematics of the swimmer’s
undulation lead to large $F_y$ values, although it also oscillates about a mean value. (c) Evolution of $\mathbf{U}$ in each coordinate direction over time for a self–propelled, undulating sheet simulation with SW = 13.33. The oscillation of the heave velocity $U_z$ about 0 is not easily visible at this scale.

4.5 Measured propulsive force from low $\text{Re}_{\text{lat}}$ undulating sheet simulations. The axial propulsive force generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate span was varied. These data represent cases where $\text{Re}_{\text{lat}} < 1 \times 10^2$.

4.6 Measured propulsive force from high $\text{Re}_{\text{lat}}$ undulating sheet simulations. The axial propulsive force generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate length was varied. These represent cases where $\text{Re}_{\text{lat}} > 1 \times 10^2$.

4.7 Measured propulsive force from low $\text{Re}_{\text{lat}}$ sheets with varying frequency and amplitude. Axial propulsive force vs. specific wavelength for additional small–sheet simulations in which (a) frequency and (b) amplitude were varied.

4.8 Measured propulsive force from low $\text{Re}_{\text{lat}}$ sheets with varying span and length. Axial propulsive force vs. specific wavelength for additional small–sheet simulations with varying aspect ratios. In case (a), plate span was varied while in case (b), plate length was varied.
4.9 Measured propulsive force from anguilliform and carangiform sheet simulations. Axial propulsive force vs. specific wavelength for additional small-sheet simulations with prescribed anguilliform (▲) and carangiform (●) amplitude profiles.

4.10 Measured swimming speed and force from realistic eel and mackerel simulations. Axial swimming speed and propulsive force computed from free-swimming (green) and translation-locked (black) simulations of undulating (a) eel bodies, and (b) mackerel bodies, plotted against the specific wavelength. In free-swimming simulations, the forward swimming speed of the undulating body was an output parameter of the simulation. These simulations were carried out at $Re_{lat} = 1.12 \times 10^2$, with corresponding swimming-speed Reynolds number range $1.378 \times 10^3 < Re < 5.95 \times 10^3$.

4.11 Vortical structures shed from free-swimming eel and mackerel. Three-dimensional vortical structures visualized for free-swimming simulations of an eel (a–c) and mackerel (d–f) at three different SW values. The wakes are visualized using isosurfaces of q-criterion, where $q = \frac{1}{2} (\|A\|^2 - \|S\|^2)$, where $A$ and $S$ are the antisymmetric and symmetric parts of the fluid velocity gradient tensor $\nabla u$, respectively.

4.12 Wake and body visualizations from free-swimming simulations. Top figures show the midline kinematics (black) over time for the three different types of undulating bodies considered in the present numerical study. Dashed red line denote the amplitude function $\pm a(x)$. Middle
and bottom figures show contours of vorticity magnitude for the three bodies. Middle figures show the top–view with undulations present in the lateral direction, while bottom figures show the side–view of each body. (a), (b), & (c) An undulating flat plate with SW = 10 and Re_{lat} = 4.49; the low Reynolds number causes the wake to remain large and mostly attached. (d), (e) & (f) An undulating eel body with SW = 10 and Re_{lat} = 1.12 \times 10^2. (g), (h), & (i) An undulating mackerel body with SW = 10 and Re_{lat} = 1.12 \times 10^2.

4.13 Computational setup and grid refinement validation case for an undulating sheet. (a) Computational setup for a translation–locked simulation of an undulating sheet with L = 20 cm, h = 2 cm, f = 3 Hz, a = 1 cm and λ = 20 cm, which corresponds to Re_{lat} = 3.37 \times 10^2. Three adaptive mesh levels are shown and the immersed structure is always placed on the finest mesh level. (b) Time evolution of axial force generated by the sheet for two different grid spacing values: \Delta X_0 = (3.125, 3.125, 3.125) \times 10^{-3}L \text{ (orange)} and \Delta X_1 = (1.5625, 1.5625, 1.5625) \times 10^{-3}L \text{ (blue)}.

4.14 Correlation between different Reynolds numbers. The swimming–speed Reynolds number Re vs. Re_{lat} (●; blue), Re_{lat,L} (♦; purple), and Re_{wave} (■; red) for the swimmers considered in [148], along with the lines of best fit. The best fit lines are given by log Re =

1.803 + 1.340 \log \text{Re}_{lat}, \log \text{Re} = -0.4058 + 1.2306 \log \text{Re}_{lat,L}, \text{ and } \log \text{Re} = -1.450 + 1.083 \log \text{Re}_{wave}.


4.15 Dimensionless axial force data from Fig 4.5. Dimensionless axial propulsive force \( \hat{F} \) generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate span was varied. These data represent cases where (a) \( \text{Re}_{\text{lat}} = 8.43 \times 10^{-1} \) and (b) \( \text{Re}_{\text{lat}} = 4.49 \).

4.16 Dimensionless axial force data from Fig 4.6. Dimensionless axial propulsive force \( \hat{F} \) generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate length was varied. These data represent cases where \( \text{Re}_{\text{lat}} = 3.37 \times 10^{2} \).

4.17 Dimensionless axial force data from Fig 4.7. Dimensionless axial propulsive force \( \hat{F} \) vs. specific wavelength for additional small-sheet simulations in which (a) frequency and (b) amplitude were varied. These data represent cases where (a) \( 2.53 \leq \text{Re}_{\text{lat}} \leq 10.11 \) and (b) \( 0.28 \leq \text{Re}_{\text{lat}} \leq 10.16 \).

4.18 Dimensionless axial force data from Fig 4.8. Dimensionless axial propulsive force \( \hat{F} \) vs. specific wavelength for additional small-sheet simulations with varying aspect ratios. In case (a), plate span was varied while in case (b), plate length was varied. These data represent cases where (a) \( \text{Re}_{\text{lat}} = 8.43 \times 10^{-2} \) and (b) \( \text{Re}_{\text{lat}} = 4.49 \).

4.19 Dimensionless axial force data from Fig 4.9. Dimensionless axial propulsive force \( \hat{F} \) generated by a stationary undulating sheet with prescribed anguilliform (▲) and carangiform (●) amplitude profiles.
plotted against specific wavelength. These data represent cases where \( \text{Re}_{\text{lat}} = 4.49 \).

4.20 Momentum transfer by an undulatory swimmer. At the front of the swimmer, the body sucks stationary fluid with negligible momentum \( m_{\text{in}} \) and accelerates it downstream. This ejected fluid is often manifested as a wake with momentum \( m_{\text{wake}} \sim M_{\text{wave}}(\lambda f) \). Finally, this wake eventually dissipates further downstream as it loses momentum.

4.21 Correlation between maximum axial fluid velocity and traveling wave velocity. The maximum fluid axial velocity \( u_{\text{max}} \) generated by the sheet over \( \Omega \) and over one swimming cycle \([t_0, t_0 + T]\) vs. the traveling wave speed \( \lambda f \).

4.22 Quantitative analysis of the velocity and friction mechanisms. (a) Net axial force generated by simulations of an undulating plate with \( L = 1.0 \) cm, \( h = 0.1 \) cm, \( f = 3 \) Hz, \( a = 0.05 \) with (red) periodic boundary conditions, and (black) wall boundary conditions. Average axial fluid momentum multiplied by undulation frequency over one swimming cycle with periodic boundary conditions (green) as functions of SW. (b) velocity of the traveling wave \( \lambda f \) (pink) and change in weighted mass (blue).

4.23 Qualitative analysis of the velocity mechanism. Mid-sheet \( (z = -0.05 \) cm) contours of axial fluid velocity for a stationary sheet simulation for various snapshots in time. Black dots represent the Lagrangian points of the undulating body. The plate has \( L = 1.0 \) cm, \( h = 0.1 \) cm, \( f = 3 \).
Hz, \( a = 0.05 \) cm with (a) \& (b) SW = 13.33, (c) \& (d) SW = 10, (e) \& (f) SW = 5.

4.24 Qualitative analysis of the friction mechanism. Mid–sheet \((z = -0.05)\) contours of normalized axial fluid velocity \((u_a/u_{max})\) for a stationary sheet simulation for various snapshots in time. Black dots represent the Lagrangian points of the undulating body. The plate has \( L = 1.0 \) cm, \( h = 0.1 \) cm, \( f = 3 \) Hz, \( a = 0.05 \) cm with (a) \& (b) SW = 13.33, (c) \& (d) SW = 10, (e) \& (f) SW = 5. For each difference SW, \( u_{max} \) is taken to be the maximum axial fluid velocity over \( \Omega \) and over one swimming cycle.

5.1 Sketch of the immersed structure (solid line) surrounded by an arbitrary control volume (dashed line)

5.2 A staggered grid spatial discretization. The \( x \)-velocity component \( u \) is solved for at locations depicted with horizontal arrows. The \( y \)-velocity component \( v \) is solved for at locations depicted with vertical arrows. The pressure \( p \) is solved for at locations depicted with a solid black dots.

The corners \((x_L, y_L)\) and \((x_U, y_U)\) define the control volume \( V_{CV}(t) \), which is shaded in red. The dashed line represents \( S_{CV}(t) = \partial V_{CV}(t) = \mathcal{L} \cup \mathcal{T} \cup \mathcal{R} \cup \mathcal{B} \).

5.3 The velocity and pressure locations on the staggered–grid discretization required to evaluate the surface integral terms in the hydrodynamic force calculation on the bottom face of \( S_{CV} \).
5.4 Comparison of the drag coefficient for flow past a cylinder at \( Re = 550 \) measured in two different ways. (—, red): Control volume using Eq. (5.16); (---, black): Lagrange multiplier using Eq. (5.45).

5.5 Comparison of the drag coefficient a translating cylinder at \( Re = 550 \) measured in three different ways. ●: Control volume using Eq. (5.15); —: Control volume using Eq. (5.16); ---: Lagrange multiplier. Note that LM and present moving CV curves are on top of each other.

5.6 Vorticity generated by a circular cylinder at \( t = 5 \) for \( Re = 550 \): (a) two disjoint control volumes; (b) two slightly overlapping control volumes; (c) each control volume contains one and a half cylinders; (d) each control volume contains both cylinders. All figures are plotted for vorticity between −20 and 20.

5.7 Temporal evolution of drag coefficient for \( Re = 550 \): (a) two disjoint control volumes; (b) two slightly overlapping control volumes; (c) one and a half cylinders; (d) two full cylinders. —: Bottom CV; --- (green): Top CV.

5.8 Temporal evolution of the drag coefficient of an oscillating cylinder at \( Re = 100 \) measured by the (a) present control volume approach and (b) Lagrange multiplier approach. Here, (—, black): \( \Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.04D \); (---, red): \( \Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.02D \); (−−−, blue): \( \Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.01D \).
5.9 Vorticity generated by an oscillating cylinder in a cross-flow at $tU_\infty/D = 100$ for Re = 185. The plotted vorticity is between −5 and 5.

5.10 Temporal evolution of drag coefficient measured by the (a) present control volume approach and (b) Lagrange multiplier approach. Here, 

(—, black): $\Delta x = \Delta y = 0.08D$; (---, red): $\Delta x = \Delta y = 0.04D$; (---, blue): $\Delta x = \Delta y = 0.02D$.

5.11 Comparison of the torque coefficient for a cylinder undergoing rotational oscillation at Re = 300 measured in two different ways. Here, (—, red):

Control volume using Eq. (5.19); (---, black): Lagrange multiplier using Eq. (5.46).

5.12 Comparison of the drag coefficient for a translating plate at Re = 20 measured by the present moving control volume approach (—, blue); (---: asymptotic value from Dennis et al. [222].

5.13 (a) Vorticity generated by a moving flat plate at $t = 7.5$ for Re = 20. The moving control volume spanning both the coarsest and finest grid level is shown in blue. The plotted vorticity is between −7.5 and 7.5.

(b) Temporal evolution of drag coefficient measured by the present moving control volume approach (—, blue); (---: asymptotic value from Dennis et al. [222].

5.14 (a) Vorticity generated by a moving flat plate at $t = 7.5$ for Re = 20. The stationary control volume spanning both the coarsest and finest grid level is shown in red. The plotted vorticity is between −7.5 and
7.5. (b) Temporal evolution of drag coefficient measured by the present control volume approach (—, red); ---: asymptotic value from Dennis et al. [222].

5.15 (a) Vorticity generated by a translating flat plate at $t = 7.5$ for $Re = 20$. The moving control volume spanning both the coarsest and finest grid level is shown in green. The plotted vorticity is between $-7.5$ and $7.5$.

(b) Temporal evolution of drag coefficient measured by the present moving control volume approach (—, green); ---: asymptotic value from Dennis et al. [222].

5.16 Vorticity generated by the two-dimensional eel model at $Re = 5609$, along with the moving CV location at four different time instances on an adaptive mesh hierarchy. All figures are plotted for vorticity between $-10$ and $10$.

5.17 (a) Temporal evolution of (—, black) axial $U_r$ & (---, blue) lateral $V_r$ swimming velocity, and (---, red) rotational $W_r^z$ velocity of the eel calculated by Eqs. (5.51) and (5.52). (b) Temporal evolution of net (—, black) axial $F_x$ & (---, blue) lateral $F_y$ forces, and (---, red) net torque $M_z$ on the body of the eel measured by the present moving control volume approach. (c) Temporal evolution of (—, black) axial $P_x$ and (---, blue) lateral $P_y$ momentum of the entire fluid domain. Temporal evolution of change in linear momentum within the control volume: (—, gray) $L_x$ and (---, red) $L_y$. 

321
5.18 Vorticity generated by two-dimensional drafting, kissing, and tumbling of two cylindrical particles, along with the moving CV location at four different time instances on an adaptive mesh hierarchy. All figures are plotted for vorticity between $-30$ and $30$.

5.19 (a) Temporal evolution of the $x$ coordinate of center of mass $X_p$ for (—, black) particle 1 & (—, green) particle 2. (b) Temporal evolution of the $y$ coordinate of center of mass $Y_p$ for (—, black) particle 1 & (—, green) particle 2. Center of mass data from Jafari et al. [224] for (×, black) particle 1 & (■, green) particle 2. (c) Temporal evolution of net $F_x$ via the moving control volume approach on (—, black) particle 1 & (—, green) particle 2. Temporal evolution of net $F_x$ via the Lagrange multiplier approach approach on (—, black) particle 1 & (—, green) particle 2. (d) Temporal evolution of net $F_y$ via the moving control volume approach on (—, black) particle 1 & (—, green) particle 2. Temporal evolution of net $F_y$ via the Lagrange multiplier approach approach on (—, black) particle 1 & (—, green) particle 2. The time period between the two red lines indicates inaccurate CV force measurements since each rectangular CV contains multiple bodies.

5.20 Surface nodes on the inner shell (red) and outer shell (gray) for Stokes flow between concentric spheres. Location of a control volume containing only the inner shell (red) and containing both the inner and outer shell (black).
6.1 The inverse of the Strouhal number \((1/\text{St})\) vs. specific wavelength (SW) for robotic experimental and simulation studies of BCF and MPF swimmers analyzed in Chapter 4 and \([151, 156]\).  

6.2 The hydrodynamic power \(P_{U_s}\) spent and axial swimming speed vs. lateral Reynolds number \(\text{Re}_{\text{lat}}\) for the simulated eel data presented in Chapter 4.  

6.3 The hydrodynamic power \(P_{U_s}\) spent and axial swimming speed vs. scaled height at three different aspect ratios for carangiform sheets.  

6.4 The (a) axial swimming speed, and (b) Strouhal number generated from simulations of an undulating eel body with fixed length \(L = 10\) cm and tail amplitude \(a = 1\) cm; (●, black) the \(\text{Re}_{\text{lat}}\) and SW values used for each simulation in the parametric study; (■, white) the \(\text{Re}_{\text{lat}}\) and SW values for observed BCF swimmers \([148]\).  

6.5 The cost of transport expended by an simulated, undulating eel body with fixed length \(L = 10\) cm and tail amplitude \(a = 1\) cm; (●, black) the \(\text{Re}_{\text{lat}}\) and SW values used for each simulation in the parametric study; (■, white) the \(\text{Re}_{\text{lat}}\) and SW values for observed BCF swimmers \([148]\).  

6.6 (a) (♦, red) Empirical scaling between the Strouhal number and wave-speed Reynolds number, and (b) (♦, green) between Reynolds number and lateral Reynolds number for meta-analysis data of BCF swimmers \([148]\), along with simulation data of undulatory (●) eel, (■) sheet, and (♦) mackerel bodies; (▲) robotic parametric study data.
of robotic, undulatory sheets \cite{155,156}. Parametric study data are colored from blue to orange to represent the variation from lower to higher swimming speeds within each set.

6.7 Schematic of the turbulent energy cascade, and trade-offs between various turbulence models.

6.8 Vorticity generated for flow past an airfoil at Re = 10^6 for pre- and post-stall angles of attack; (left) RANS model activated and (right) DES model activated.

6.9 (a) Visualization of a translating Ahmed body with unlocked degrees of freedom in the z-direction. (b) Temporal evolution of the z-coordinate of the body’s center of mass.

6.10 (a)-(d) Visualization of a self-propelled Ahmed body accelerating from rest at four instances in time. (e) Temporal evolution of the body’s forward velocity. The model has an approximate weight of 661 pounds.

6.11 Numerically computed signed distance function to a finite element representation of the (a) Ahmed body, and (c) SAE Notchback model. Zero contour of the numerically computed surface level set function $\psi$ for the (b) Ahmed body, and (d) SAE Notchback model.
CHAPTER 1

Introduction

Over the past few decades, computational fluid dynamics (CFD) has become a standard analysis tool for optimizing engineering design processes. CFD methods are now ubiquitous in academia and industry and have been proven useful for analyzing complex automotive [1], aerospace [2], biological [3], and many other fluid flow problems, for which theoretical analyses can be intractable. Fluid-structure interaction (FSI) problems are of particular interest to practitioners because many engineering phenomena involve the interplay between heavy objects and one (more multiple) fluid phases. FSI modeling has traditionally been carried out using body-fitted mesh approaches, in which the Navier-Stokes equations are discretized and solved for on unstructured grids that conform around some immersed body. Although these methods sharply resolve the fluid-structure interface, they require complex mesh management infrastructure and incur high computational costs for solving linear systems of equations. These costs are exacerbated for problems in which the immersed body undergoes significant deformation or motion.

Another class of methods for simulating these complicated FSI problems is the fictitious domain approach, which has been gaining popularity in recent years. In these approaches, the fluid domain is extended and discretized inside the region occupied by the structure and additional forces are computed on the right-hand side of the momentum equations to account for the fluid-solid coupling. One such technique is the immersed boundary (IB) method, which was introduced in 1972 by Peskin [4, 5] in the context of cardiovascular
flows. The key concept of the IB method is the split representation of the Eulerian fluid mesh and the Lagrangian solid mesh. The immersed structure is discretized by a set of connected, Lagrangian markers, which is tracked in a reference frame attached to the body, while the fluid domain is discretized by a simple Cartesian mesh. The structure is allowed to freely cut the background fluid grid, thereby eliminating the use of a body-conforming mesh. A constitutive model is used to compute elastic forces between markers, and energy-conserving grid transfer operations are used to spread forces to the Eulerian mesh and interpolate velocities onto the Lagrangian mesh. Since its introduction, variants of the IB method have been widely popular and applied in many fields.

However, conventional IB methods impose severe time-step restrictions for rigid body simulations due to the high stiffness coefficients required for the Lagrangian markers. These deficiencies were overcome by another fictitious domain approach called the distributed Lagrange multiplier (DLM) method, which was introduced in 1999 [6, 7, 8, 9]. In this formulation, an additional rigidity constraint is imposed inside the volumetric structure domain in terms of the deformation rate tensor. An efficient numerical method to enforce this constraint was described by Patankar et al. [7]; one can simply solve for the fluid velocity everywhere in the domain without taking into account the effect of the body, and then project that velocity field onto one that satisfies the rigidity constraint. This solution methodology is akin to Chorin’s projection method that enforces the divergence-free condition for incompressible flows [10, 11]. Although the original formulation was restricted to particulate flows, Shirgaonkar et al. extended the method to allow for prescribed body deformation, which enabled fast simulation of undulatory swimming and aquatic locomotion [12]. In 2013, Bhalla et al. unified the IB and DLM
methods by developing a constraint-based immersed body (cIB) framework for simulating neutrally-buoyant rigid and elastic structures, which has been widely used by the research community [13].

A major disadvantage of Cartesian grid based fictitious domain methods is poor solution accuracy at fluid-solid interface. To overcome this issue, Bhalla et al. implemented their method within the IBAMR library [14]. IBAMR is a simulation software focused on immersed boundary methods with adaptive mesh refinement (AMR) and distributed memory parallelism; it was created by Boyce E. Griffith and is publicly maintained at https://github.com/IBAMR/IBAMR. IBAMR relies on SAMRAI [15, 16] for Cartesian grid management and the AMR framework. Linear and nonlinear solver support in IBAMR is provided by the PETSc library [17, 18, 19]. Regions in the physical domain can be selectively refined, leading to better computational efficiencies than uniform meshes. The work presented here also made extensive use of IBAMR.

In this thesis, we present applications and extensions of Bhalla et al.’s method, which enable the simulation of even more complicated fluid flow problems. In Chapter 2 we describe a robust, incompressible Navier-Stokes solver for high density ratio and high shear multiphase flows, which is shown to be accurate and stable for material contrast ratios of up to six orders of magnitude. The solution methodology ensures consistent transport of mass and momentum, which is required for air-water density ratios. The momentum and continuity saddle-point system is solved in a monolithic fashion without any time splitting and the solver is well-balanced with respect to the gravitational, surface tension, and pressure gradient forces. In Chapter 3 we couple this fluid solver to the constraint-based
immersed body method, which enables fast simulations of wave-structure interaction. Various benchmark cases are presented to validate these numerical schemes.

In Chapter 4, we use the cIB method to investigate hydrodynamic optimality for body/caudal fin (BCF) swimmers. Through a meta-analysis, we find that swimming animals in nature undulate at a wavelength to tail amplitude ratio of around 10. We call this parameter the specific wavelength (SW). Through a parametric study, we find that an optimal specific wavelength (OSW) value of around 10 maximizes swimming speed or thrust in numerical simulations of realistic fish bodies, which indicates a possible reason for the natural emergence of this trait. In Chapter 5, we present a moving control volume approach to smoothly compute hydrodynamic forces and torques in immersed boundary simulations. This approach does not require evaluation of (possibly) discontinuous spatial velocity or pressure gradients within or on the surface of the immersed body, and can be used to obtain useful post-processed results. Finally in Chapter 6, we describe topics of ongoing and future research, which includes unification of the aquatic locomotion parameter space and simulation of industrial and turbulent fluid-structure interaction problems.
CHAPTER 2

A robust incompressible Navier-Stokes solver for high density ratio multiphase flows

This chapter presents a robust, adaptive numerical scheme for simulating high density ratio and high shear multiphase flows on locally refined staggered Cartesian grids that adapt to the evolving interfaces and track regions of high vorticity. The algorithm combines the interface capturing level set method with a variable-coefficient incompressible Navier-Stokes solver that is demonstrated to stably resolve material contrast ratios of up to six orders of magnitude. The discretization approach ensures second-order point-wise accuracy for both velocity and pressure with several physical boundary treatments, including velocity and traction boundary conditions. The chapter includes several test cases that demonstrate the order of accuracy and algorithmic scalability of the flow solver. To ensure the stability of the numerical scheme in the presence of high density and viscosity ratios, we employ a consistent treatment of mass and momentum transport in the conservative form of discrete equations. This consistency is achieved by solving an additional mass balance equation, which we approximate via a strong stability preserving Runga-Kutta time integrator and by employing the same mass flux (obtained from the mass equation) in the discrete momentum equation. The scheme uses higher-order total variation diminishing (TVD) and convection-boundedness criterion (CBC) satisfying limiter to avoid numerical fluctuations in the transported density field. The high-order
bounded convective transport is done on a dimension-by-dimension basis, which makes the scheme simple to implement. We also demonstrate through several test cases that the lack of consistent mass and momentum transport in non-conservative formulations, which are commonly used in practice, or the use of non-CBC satisfying limiters can yield very large numerical error and very poor accuracy for convection-dominant high density ratio flows. Our numerical scheme also uses well-balanced surface tension and gravity force discretizations. In the hydrostatic limit, we show that the well-balanced formulation mitigates spurious flow currents and achieves discrete force-balance between the pressure gradient and surface tension or gravity.

2.1. Introduction

Incompressible two-phase flows involving liquid and gas at ambient conditions are prevalent in engineering and natural processes. The high density and high viscosity contrasts in multiphase flows often result in significant shear at the fluid-fluid interface. This poses a significant challenge in the numerical modeling of multicomponent flows because of the stiff system of equations arising from the discretization of the incompressible Navier-Stokes equations. In addition, numerical error accumulation near the highly deforming interface often leads to catastrophic failure of the scheme. Several multiphase flows are driven by interfacial forces like surface tension and volumetric forces like gravity; avoiding spurious currents near such interfaces requires a consistent and well-balanced discretization of the forces. Despite these difficulties, multiphase flow modeling has been the subject of extensive research for the past two decades owing to its importance in both industrial and natural processes.
Several approaches to track the phases of a multiphase flow simulation exist in the literature, the most prominent being the volume-of-fluid (VOF) method of Hirt and Nichols [20] and the level set method of Osher and Sethian [21]. The VOF method tracks the volume fraction of each phase in each computational cell and reconstructs the phase interface in a piecewise fashion from the volume fraction data. The advection of the VOF-scalar is done though geometric means to ensure strict mass conservation and boundedness of volume fraction. The disadvantages of VOF method include its non-smooth interface representation and tedious geometry implementation requirements. On the other hand, the level set method captures the phase interface by computing a signed distance function. The zero level set of the distance field implicitly defines the position of the interface. The level set method formulates the geometric problem of distance finding as a nonlinear hyperbolic partial differential equation (PDE) for which established numerical techniques can be applied. Higher-order hyperbolic PDE discretization methods can enable smooth interface representation and accurate curvature calculation. The level set method can also lead to spurious mass changes resulting from the transport and reinitialization of the signed distance function [22]. However, this issue can be minimized by penalization techniques [23] or by hybridizing the level set method with VOF-like methods, as done in the mass-conserving level set method (MCLS) [24] and the coupled level set volume-of-fluid methods (CLSVOF [25], and VOSET [26]). In this work, we use the level set method with sign [27] and subcell [28] fixes to mitigate spurious changes in the mass of each phase.

A challenging problem that has not received much attention in the multiphase flow literature is the treatment of large density ratios. The multiphase community has only
recently begun to address the fundamental cause and resolution of numerical instabilities associated with flows with density ratios on the order of 100–1000 and greater. Over the past few years, many authors have proposed strategies for mitigating these instabilities within a wide range of numerical contexts; for example, Li et al. [29] for a moment-of-fluid method, Vaudor et al. [30] for a staggered grid CLSVOF method, Le Chenadec and Pitsch [31] and Owkes and Desjardins [32] for sharp interface VOF methods, and Jemison et al. [33] and Duret et al. [34] for compressible flow solvers, to name a few. Interestingly, some numerical schemes based on the VOF method, particularly that of Rudman [35] and Bussman [36], are not susceptible to flow instabilities with large density ratios because of their use of consistent transport schemes for the convective momentum and VOF-scalar in the discrete set of conservative equations. In the context of the level set framework, in which geometric information about interface transport is absent, it is difficult to maintain a discrete compatibility between mass and momentum transport. Raessi [37] and Raessi and Pitsch [38] introduced the first strategy for dealing with high density ratio flows within the level set framework by introducing geometric mass flux transport. This work has been limited to one- and two-dimensional problems because of the inherent difficulty of interface reconstruction and geometric transport with level sets. A more elegant solution to this problem that is easily extended to three spatial dimensions was proposed by Desjardins and Moureau [39] in the context of structured, staggered-grid Cartesian discretizations, and more recently by Ghods and Herrmann [40] for collocated unstructured grids. Instead of constructing piecewise linear interfaces from the level set field, they suggest solving an additional mass balance equation along with momentum and incompressibility constraint equations. The advantage of this approach
is that the same (algebraic) mass flux can be used in both mass and momentum transport for a given time step, yielding a stable scheme for high density ratio flows. The level set is transported independently and is only used to synchronize the density field with level set field at the beginning of each time step. To prevent numerical oscillations in the density field near the interface, both Desjardins and Moureau [39] and Ghods and Herrmann [40] used first-order upwinding for density and velocity transport. Doing so, however, produces diffusive flow features and smeared vortices. A key contribution of this work is to extend their approach to a second-order accurate, bounded scheme. The resulting method is empirically demonstrated to remain stable even at very high density ratios of $10^6$. This is achieved in this work by employing a modified version of the third-order accurate Koren’s limited Cubic Upwind Interpolation (CUI) scheme described by Patel and Natarjan [41]. Koren’s limited CUI is a total variation diminishing (TVD) and convection-boundedness criterion (CBC) satisfying convective limiter. The mass equation is integrated using a third-order strong-stability preserving Runge-Kutta time integrator (SSP-RK3) [42], and the mass flux from the penultimate (second) stage of the SSP-RK3 scheme is used in the convective operator to maintain discrete compatibility. We also compare the performance of the scheme using TVD but non-CBC convective limiters, such as a version of the piecewise parabolic method (PPM), to CBC limiters like CUI, the Modified Gamma scheme (M-Gamma) [41], and the Flux Blending Interface Capturing scheme (FCIBS) [43] for convection-dominant high density ratio flows.

More recently Patel and Natarajan [44] used a consistent convective scheme for momentum and algebraic (i.e., geometric reconstruction-free) VOF transport. Their consistent transport scheme implicitly maintains the discrete compatibility of the density flux
in the mass and momentum equations. Through several examples, they show that any mismatch between mass and momentum transport results in an inconsistent scheme and can result in extremely poor accuracy in certain flow scenarios. We remark that this is not the only way to achieve stabilization for high density ratio flows. For example, Desjardins and Moureau [39] and Sussman et al. [45] use the ghost fluid method to maintain sharp discontinuities across the interface within a hybridized level set framework. This approach also requires the use of some velocity extension algorithm across the phase interface to advect the level set field. Specifically, Sussman et al. [45] use the higher density fluid’s extrapolated velocity to advect the interface. We do not analyze the stability of sharp interface methods for high density ratio multiphase flows in this work.

A major challenge in incompressible flow simulation is the treatment of coupled velocity-pressure saddle-point system. In this work, we solve this saddle-point system in a monolithic fashion (without any time-splitting approach) using a preconditioned flexible GMRES scheme. We achieve high performance via a novel preconditioning strategy that combines a local-viscosity preconditioner [46] with a projection method-based preconditioner introduced by Griffith in the context of constant coefficient problems [47] and extended by Cai et al. [48] to treat variable coefficient problems. This work extends the uniform grid variable-coefficient solvers of Cai et al. to locally refined grids. Another difference is that Cai et al. [48] considers the non-conservative form of equations, i.e, they compute velocity and pressure fields for a given density field. In this work, because we employ consistent mass and momentum transport with conservative form of equations, we solve an additional mass balance equation for density evolution. We empirically demonstrate that our spatio-temporal discretization achieves pointwise second-order accuracy.
for the velocity, pressure, and density fields for a variety of boundary conditions. We note that most of the multiphase literature [36, 49, 50, 51, 52, 53, 54] treats the spatially varying viscous operator explicitly, although an implicit treatment has been used by some authors as well [45, 55, 56]. Because we use a solver strategy that allows us to treat the viscous operator implicitly, we are able to use a time step size based only on an advective CFL condition. In the context of a level set framework, the implicit treatment of the viscous terms reduces the frequency of level set reinitialization without degrading accuracy, which reduces numerical perturbations in the interface configuration [57].

The remainder of the chapter is organized as follows. We first introduce the continuous and discrete system of equations in Sections 2.2 and 2.3, respectively. Next we discuss the projection preconditioner in Section 2.4. Software implementation is described in Section 2.5. Accuracy and scalability results for the monolithic flow solver are presented thereafter for both conservative and non-conservative discretizations in Section 2.6. Finally, two-phase flow examples highlighting the importance of consistent and bounded mass and momentum transport are presented in Section 2.7. We also contrast the consistent results against the results obtained from an inconsistent and non-conservative flow solver. Wherever possible, simulation results from locally refined grids are presented. Examples demonstrating the well-balanced discretization of the interfacial surface tension and volumetric gravitational force are also presented. Well-balanced force formulations are particularly important in the hydrostatic limit, where it helps to mitigate spurious flow currents. This is achieved through a discrete force-balance between the pressure gradient and interfacial/external forces as shown by our test cases.
2.2. The continuous equations of motion

We follow the single fluid formulation for multiphase flows and consider a single viscous incompressible fluid with spatially and temporally varying density and viscosity occupying a fixed region of space $\Omega \subseteq \mathbb{R}^d$. The equations of motion for the fluid are the incompressible Navier-Stokes equations, which in conservative form are

\begin{align}
\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u} &= -\nabla p + \nabla \cdot \left[ \mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right] + \mathbf{f} + \mathbf{f}_s, \\
\nabla \cdot \mathbf{u} &= 0,
\end{align}

in which $\mathbf{x} = (x, y) \in \Omega$ are the fixed physical Eulerian coordinates, $\mathbf{u}(\mathbf{x}, t) = (u(x, t), v(x, t))$ is the fluid velocity, $p(\mathbf{x}, t)$ is the pressure, $\mathbf{f}(\mathbf{x}, t) = (f_1(x, t), f_2(x, t))$ is the momentum body force, $\mathbf{f}_s(\mathbf{x}, t)$ is the continuum surface tension force, and $\rho(\mathbf{x}, t)$ and $\mu(\mathbf{x}, t)$ are the spatially and temporally varying fluid density and dynamic viscosity, respectively. Note that the continuity constraint Eq. (2.2) follows directly from the conservation of mass equation over the entire domain,

\begin{align}
\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} &= \frac{D \rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0,
\end{align}

and the incompressible nature of the fluid, which in the Langrangian form can be expressed as $\frac{D \rho}{Dt} = 0$. The equations of motion can also be cast to non-conservative form by combining Eqs. (2.1), (2.2) and (2.3), so that

\begin{align}
\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u} \mathbf{u} \right) &= -\nabla p + \nabla \cdot \left[ \mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right] + \mathbf{f} + \mathbf{f}_s, \\
\nabla \cdot \mathbf{u} &= 0.
\end{align}
Although these forms of the equations are equivalent, direct discretizations of these equations lead to numerical schemes with different properties, as discussed in later sections. Note that the convective term $\nabla \cdot uu$ in Eq. (2.4) can also be expressed as $\mathbf{u} \cdot \nabla \mathbf{u}$ using the continuity constraint.

To complete the description of the equations of motion, it is necessary to specify initial conditions for $\mathbf{u}$, $\rho$, and $\mu$ and boundary conditions on $\partial \Omega$. As described by Griffith \[47\] for the constant-coefficient case, in this work we consider three types of boundary conditions: periodic, prescribed velocity, and prescribed traction.

Finally, suppose that two fluids with differing densities and viscosities occupy the regions in the computational domain $\Omega_0(t) \subset \Omega$ and $\Omega_1(t) \subset \Omega$, respectively. The codimension-1 interface between the fluids $\Gamma(t) = \Omega_0 \cap \Omega_1$ can be tracked as the zero contour of a scalar function $\phi(x, t)$, which is the so-called level set function \[21, 59, 49\],

\begin{equation}
\Gamma(t) = \{ x \in \Omega | \phi(x, t) = 0 \}.
\end{equation}

Level set methods are well-suited for tracking interfaces undergoing complex topological changes and are relatively easy to implement in both two and three spatial dimensions. The density and viscosity in the two phases are determined as a function of this scalar field by

\begin{equation}
\rho(x, t) = \rho(\phi(x, t)),
\end{equation}

\begin{equation}
\mu(x, t) = \mu(\phi(x, t)).
\end{equation}
The discretized form of Eqs. (2.7) and (2.8) are defined in Section 2.3.3 using a regularized Heaviside function. The signed distance function is passively advected by the incompressible fluid velocity, which in conservative form reads

\[
\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = 0.
\]

One common choice of \( \phi \) is the signed distance function; however, \( \phi \) generally will not remain a signed distance function under advection by Eq. (2.9) \[60\]. A reinitialization or redistancing procedure is used to maintain the signed distance property of \( \phi \) at every time step. When the fluid properties are determined from \( \phi \), we need initial conditions for \( \phi \) but not for \( \rho \) or \( \mu \).

### 2.3. The discretized equations of motion

This section details the discretizations of the non-conservative and conservative forms of the governing equations. For notational simplicity, we present the discretized equations in two spatial dimensions. An extension to three spatial dimensions is straightforward with the exception of the discretization of the viscous term; the treatment of this term in three dimensions is described in Sec. 2.3.7. The treatment of physical boundary conditions in this work follows an approach similar to that of Griffith \[47\] and is detailed in Sec. 2.3.8.

#### 2.3.1. Basic spatial discretization

This work uses a staggered-grid discretization of the equations of incompressible fluid flow on a rectangular domain \( \Omega \). A \( N_x \times N_y \) Cartesian grid covers the physical domain \( \Omega \) with mesh spacing \( \Delta x \) and \( \Delta y \) in each direction. Without loss of generality, the bottom left
corner of the domain is assumed to be situated at the origin \((0,0)\). The position of each grid cell center is then given by \(x_{i,j} = ((i + \frac{1}{2})\Delta x, (j + \frac{1}{2})\Delta y)\) for \(i = 0, \ldots, N_x - 1\) and \(j = 0, \ldots, N_y - 1\). For a given cell \((i,j)\), \(x_{i-\frac{1}{2},j} = (i\Delta x, (j + \frac{1}{2})\Delta y)\) is the physical location of the cell face that is half a grid space away from \(x_{i,j}\) in the \(x\)-direction, and \(x_{ij-\frac{1}{2}} = ((i + \frac{1}{2})\Delta x, j\Delta y)\) is the physical location of the cell face that is half a grid cell away from \(x_{i,j}\) in the \(y\)-direction. The pressure is approximated at cell centers and is denoted by \(p_{ij}^n \approx p(x_{ij}, t^n)\), in which \(t^n\) is the time at time step \(n\). Velocity components are defined at cell faces: \(u_{i-\frac{1}{2},j}^n \approx u(x_{i-\frac{1}{2},j}, t^n)\) and \(v_{ij-\frac{1}{2}}^n \approx v(x_{ij-\frac{1}{2}}, t^n)\). The components of the body force \(f = (f_1, f_2)\) are also approximated at \(x\)- and \(y\)-faces of the staggered grid cells, respectively. The density and viscosity are approximated at cell centers of the staggered grid and are denoted by \(\rho_{ij}^n \approx \rho(x_{ij}, t^n)\) and \(\mu_{ij}^n \approx \mu(x_{ij}, t^n)\). In our numerical scheme, these values are interpolated onto the required degrees of freedom, as needed. Similarly, the phase interface is tracked via the level set function, which is also defined at cell centers and denoted by \(\phi_{ij}^n \approx \phi(x_{ij}, t^n)\). See Fig. 2.1. Finally, we note that it is sometimes convenient to directly approximate the density field on faces of the staggered grid, i.e. \(\rho_{i-\frac{1}{2},j}^n \approx \rho(x_{i-\frac{1}{2},j}, t^n)\) and \(\rho_{ij-\frac{1}{2}}^n \approx \rho(x_{ij-\frac{1}{2}}, t^n)\), despite it being a scalar quantity; the reasoning behind this will be made apparent in Sec. 2.3.5.2.

The staggered-grid finite-difference approximations to the spatial differential operators have been described in various prior studies [47, 48, 61, 62]. We briefly summarize them here to complete the description of the spatial discretization. The divergence \(\mathbf{D} \cdot \mathbf{u}\) of the
velocity field $\mathbf{u} = (u, v)$ is approximated at cell centers by

\begin{align}
(\mathbf{D} \cdot \mathbf{u})_{i,j} &= (D^x u)_{i,j} + (D^y v)_{i,j}, \\
(D^x u)_{i,j} &= \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}, \\
(D^y v)_{i,j} &= \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}.
\end{align}

The components of the gradient $\mathbf{G}p = (G^x p, G^y p)$ of cell-centered pressure $p$ are approximated at cell faces by

\begin{align}
(G^x p)_{i-\frac{1}{2},j} &= \frac{p_{i,j} - p_{i-1,j}}{\Delta x}, \\
(G^y p)_{i,j-\frac{1}{2}} &= \frac{p_{i,j} - p_{i,j-1}}{\Delta y}.
\end{align}
The continuous strain rate tensor form of the viscous term is

$$\nabla \cdot \left[ \mu \left( \nabla u + \nabla u^T \right) \right] = \begin{bmatrix}
2 \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right) \\
2 \frac{\partial}{\partial y} \left( \mu \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial x} \left( \mu \frac{\partial v}{\partial x} + \mu \frac{\partial u}{\partial y} \right)
\end{bmatrix},$$

which leads to velocity coupling in the discretization for variable viscosity flows

$$\mathbf{L}_\mu \mathbf{u} = \begin{bmatrix}
(L_\mu u)_i^{x_{i-\frac{1}{2},j}} \\
(L_\mu u)_i^{y_{i,j-\frac{1}{2}}}
\end{bmatrix}.$$

The viscous operator is approximated using standard second-order, centered finite-differences

$$\begin{align*}
(L_\mu u)_i^{x_{i-\frac{1}{2},j}} &= \frac{2}{\Delta x} \left[ \mu_{i,j} \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} - \mu_{i-1,j} \frac{u_{i-1,\frac{1}{2},j} - u_{i-\frac{3}{2},j}}{\Delta x} \\
&+ \frac{1}{\Delta y} \left[ \mu_{i-\frac{1}{2},j+\frac{1}{2}} \frac{v_{i-\frac{1}{2},j+\frac{1}{2}} - v_{i-1,j+\frac{1}{2}}}{\Delta x} - \mu_{i-\frac{1}{2},j-\frac{1}{2}} \frac{v_{i-\frac{1}{2},j-\frac{1}{2}} - v_{i-1,j-\frac{1}{2}}}{\Delta x} \right]
\end{align*}$$

$$\begin{align*}
(L_\mu u)_i^{y_{i,j-\frac{1}{2}}} &= \frac{2}{\Delta y} \left[ \mu_{i,j} \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y} - \mu_{i,j-1} \frac{v_{i,j-1,\frac{1}{2}} - v_{i,j-1,\frac{1}{2}}}{\Delta y} \\
&+ \frac{1}{\Delta x} \left[ \mu_{i+\frac{1}{2},j-\frac{1}{2}} \frac{u_{i+\frac{1}{2},j-\frac{1}{2}} - u_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x} - \mu_{i+\frac{1}{2},j+\frac{1}{2}} \frac{u_{i+\frac{1}{2},j+\frac{1}{2}} - u_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x} \right]
\end{align*}$$

in which we require approximations to the viscosity at both the centers and nodes of the Cartesian grid cells. Node centered quantities are obtained via interpolation by either arithmetically averaging the neighboring cell centered quantities,

$$\mu_{i-\frac{1}{2},j-\frac{1}{2}}^A = \frac{\mu_{i,j} + \mu_{i-1,j} + \mu_{i,j-1} + \mu_{i-1,j-1}}{4}.$$
or harmonically averaging those quantities,

\[
\mu^H_{i-\frac{1}{2},j-\frac{1}{2}} = \left( \frac{1}{\mu_{i,j}} + \frac{1}{\mu_{i-1,j}} + \frac{1}{\mu_{i,j-1}} + \frac{1}{\mu_{i-1,j-1}} \right)^{-1}.
\]

We remark that in three spatial dimensions, the viscosity is required at the centers and edges of the Cartesian grid cells. The full three-dimensional discretization of the viscous term is detailed in Sec. 2.3.7. An additional approximation to the density-weighted, variable-coefficient Laplacian is required for the projection preconditioner described in Sec. 2.6.

\[
(L_p p)_{i,j} = \frac{1}{\Delta x} \left[ \frac{1}{\rho_{i+\frac{1}{2},j}} \frac{p_{i+1,j} - p_{i,j}}{\Delta x} - \frac{1}{\rho_{i-\frac{1}{2},j}} \frac{p_{i,j} - p_{i-1,j}}{\Delta x} \right] + \frac{1}{\Delta y} \left[ \frac{1}{\rho_{i,j+\frac{1}{2}}} \frac{p_{i,j+1} - p_{i,j}}{\Delta y} - \frac{1}{\rho_{i,j-\frac{1}{2}}} \frac{p_{i,j} - p_{i,j-1}}{\Delta y} \right],
\]

which requires density on faces of the Cartesian grid cells. These values also can be determined using either the arithmetic or harmonic averages of density from the two adjacent cell centers.

Evaluating finite-difference operators near boundaries of the computational domain and locally refined mesh boundaries requires specification of abutting “ghost” values, which will be described in Section 2.3.6 and Sec. 2.3.8.

### 2.3.2. Discretization of the convective derivative

In the present work, the nonlinear convective term, \( \nabla \cdot uu \) for non-conservative form and \( \nabla \cdot \rho uu \) for conservative form, is computed using the third-order accurate Koren’s limited cubic upwind interpolation (CUI) scheme, first proposed by Roe and Baines and
further investigated by Waterson and Deconinck [64] and by Patel and Natarajan [41] for multiphase flows. CUI satisfies both a convection-boundedness criterion (CBC) as well as the total variation diminishing (TVD) property, both of which are essential to ensure a monotonic and bounded convective scheme. This specific method belongs to a class of nonlinear upwind schemes (generally formulated in terms of flux limiters or normalized variables [64]), which attempt to achieve higher than first-order accuracy while maintaining the monotonicity of the convected variable. The nonlinear monotonic schemes overcome the consequences of Godunov’s order barrier theorem [65], which states that no linear convection scheme of second-order accuracy or higher can be monotonic. These schemes are generally described for cell-centered quantities, but can be formulated for face-centered quantities after appropriate shifting and averaging operations [47]. We will describe this next.

For simplicity, we describe the CUI discretization of \((\nabla \cdot \mathbf{u}\psi)_{i-\frac{1}{2},j}\), in which \(\psi\) is defined on \(x\)-faces and is advected by a staggered grid velocity \(\mathbf{u}\). Note that \(\psi\) in this case could be the \(x\)-component of velocity \(u_{i-\frac{1}{2},j}\) or the density on \(x\)-faces \(\rho_{i-\frac{1}{2},j}\). (The need for density advection will become apparent in Sec. 2.3.5.) As a first step, we construct control volumes centered about \(x_{i-\frac{1}{2},j}\) by shifting the computational grid by \(\frac{\Delta x}{2}\) in the \(x\)-direction; see Fig. 2.2. The advection velocity \(\mathbf{u}_{\text{adv}}\) on the faces of the shifted control volume is obtained by averaging the adjacent velocity components. As shown in Fig. 2.2

...
these velocity components are

\[
\begin{align*}
    u_w &= \frac{u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j}}{2}, \\
    u_e &= \frac{u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j}}{2}, \\
    v_s &= \frac{v_{i-1,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}}}{2}, \\
    v_n &= \frac{v_{i+1,j+\frac{1}{2}} + v_{i,j+\frac{1}{2}}}{2}.
\end{align*}
\]

(2.22)

Next, we use the CUI scheme to obtain \(\psi_w, \psi_e, \psi_s,\) and \(\psi_n\) on the faces of the shifted control volume. For a given shifted face \(f \in \{e, w, n, s\},\) the upwind \(\psi_C,\) far upwind \(\psi_U\) and downwind \(\psi_D\) are labeled depending on the direction of the advection velocity as
shown in Fig. 2.3. For instance, Fig. 2.3(a) depicts the case where \( u_e \geq 0 \) in which the face right of \( e \) is labeled as downstream (\( \psi_D \)) and the two faces counting leftward from \( e \) are labeled upwind (\( \psi_C \)) and far upwind (\( \psi_U \)). Analogous three-point stencils are used for the other shifted control volume faces. The upwinded (limited) approximation of \( \psi_f,\lim \) on a shifted face can be written in “normalized variable” form as

\[
\tilde{\psi}_f = \begin{cases} 
3\tilde{\psi}_C, & 0 < \tilde{\psi}_C \leq \frac{2}{13} \\
\frac{5}{6}\tilde{\psi}_C + \frac{1}{3}, & \frac{2}{13} < \tilde{\psi}_C \leq \frac{4}{5} \\
1, & \frac{4}{5} < \tilde{\psi}_C \leq 1 \\
\tilde{\psi}_C, & \text{otherwise},
\end{cases}
\]

in which the normalized value is defined by

\[
\tilde{\psi} = \frac{\psi - \psi_U}{\psi_D - \psi_U}.
\]

Finally, we compute the approximation to \((\nabla \cdot \mathbf{u}\psi)_{i-\frac{1}{2},j}\) via

\[
(\nabla \cdot \mathbf{u}_{\text{adv}\psi,\lim})_{i-\frac{1}{2},j} \approx \frac{u_e \psi_e - u_w \psi_w}{\Delta x} + \frac{v_n \psi_n - v_s \psi_s}{\Delta y}.
\]

A similar procedure can be followed to compute a discretization for advected quantities on the \( y \)-faces. It is also straightforward to extend this scheme to three spatial dimensions because all the computations are performed on a dimension-by-dimension basis. We remark that there are other CBC and TVD satisfying limiters that can be used in place of CUI, including M-Gamma \[41\] and FBICS \[43\]. We refer readers to Waterson and Deconinck \[64\] and to Patel and Natarajan \[41\] for further details. We also employ TVD
but non-CBC satisfying high resolution limiters like xsPPM7 \cite{47, 66}, which is a version of the piecewise parabolic method (PPM), to advect scalars in this work. This method,
however, can produce unphysical oscillations in the values of variables that have numerical bounds, as evidenced by numerical experiments reported in Sec. 2.7.3. Therefore, we use CUI (unless otherwise stated) for the convective discretization, and we use xsPPM7 to advect scalars such as the signed distance function only where noted. We remark that if density and viscosity are formulated in terms of a volume-fraction variable \( \alpha \), which has bounds \( 0 \leq \alpha \leq 1 \), a CBC satisfying limiter must be employed to advect and bound \( \alpha \). We also remark that in the cases for which a cell-centered approximation to an advective derivative is needed (e.g., \( (\nabla \cdot u_\phi)_{i,j} \) such as for evolving the level set function), the convective scheme remains the same (as described above), and the control volume is the unshifted grid cell. As a result, the advective velocity for the cell-centered case would be the original staggered grid velocity \( u \) which is discretely divergence-free by construction. Finally, it is a simple exercise to show that if the staggered grid velocity is discretely divergence-free, then its linear interpolation onto the staggered control volumes also satisfies the same discrete divergence-free property on those control volumes away from the boundaries of the computational domain. Further, if a discrete divergence-free condition is also enforced in the ghost cells abutting the physical domain, as done in this work (see Sec. 2.3.8), the interpolated advective velocity for the face-centered control volume is discretely divergence-free throughout the computational domain.

2.3.3. Interface tracking: the level set method

The interface between the two phases is represented by the zero level set of the scalar field \( \phi(x, t) \). It is convenient to initialize \( \phi \) to be the signed distance from the interface
\( \Gamma^0 = \Gamma(0) \) (Eq. (3.7)), i.e.

\[
\phi_{i,j}^0 = \begin{cases} 
\min_{y \in \Gamma_0} \|x_{i,j} - y\|, & x_{i,j} \in \Omega_0(0) \\
-\min_{y \in \Gamma_0} \|x_{i,j} - y\|, & x_{i,j} \in \Omega_1(0),
\end{cases}
\]

which can be computed analytically for the simple initial interfaces considered in the present work. Note that \( \phi \) is not guaranteed to retain the signed distance property under linear advection, Eq. (2.9). Let \( \tilde{\phi}^{n+1} \) denote the level set function following an advection procedure after time stepping through the interval \([t^n, t^{n+1}]\). We aim to reinitialize it to obtain a signed distance function \( \phi^{n+1} \). This can be achieved by computing a steady-state solution to the Hamilton-Jacobi equation

\[
\frac{\partial \phi}{\partial \tau} + \text{sgn}(\tilde{\phi}) (\|\nabla \phi\| - 1) = 0,
\]

\[
\phi(x, \tau = 0) = \tilde{\phi}(x),
\]

in which we have dropped the \( n + 1 \) superscript because this process is agnostic to the particular time step under consideration. At the end of a physical time step, Eq. (2.27) is evolved in pseudo-time \( \tau \), which, at steady state, produces a signed distance function satisfying the Eikonal equation \( \|\nabla \phi\| = 1 \). Here, sgn denotes the sign of \( \tilde{\phi} \), which is either 1, \(-1\), or 0. The discretization of Eq. (2.27) from the psuedo-time interval \([\tau^m, \tau^{m+1}]\) yields

\[
\frac{\phi^{m+1} - \phi^m}{\Delta \tau} + \text{sgn}(\tilde{\phi}_{i,j}) \left[ H_G \left( D_x^+ \phi_{i,j}, D_x^- \phi_{i,j}, D_y^+ \phi_{i,j}, D_y^- \phi_{i,j} \right) - 1 \right] = 0,
\]
in which $H_G$ denotes a discretization of $\|\nabla \phi\|$ using the Godunov-Hamiltonian, and $D_{x}^{\pm}$ and $D_{y}^{\pm}$ denote one-sided discretizations of $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$, respectively. These are typically discretized using high-order essentially non-oscillatory (ENO) or weighted ENO (WENO) schemes \[68\].

It is well known that continually applying Eq. (2.29) will cause the interface to shift as a function of $\tau$ \[69\], which will eventually shrink closed interfaces and lead to substantial spurious changes in the volume of each phase. To mitigate this numerical issue, we employ second-order ENO finite differences combined with a subcell-fix method described by Min \[28\]. Briefly, the subcell-fix method uses $\tilde{\phi}$ to estimate the interface location (i.e., where $\tilde{\phi} = 0$) by fitting a high-order polynomial and computing an improved estimate of the one-sided derivates $D_{x}^{\pm}$ and $D_{y}^{\pm}$ from the polynomial fit. A dimension-by-dimension approach is followed to fit the high-order polynomial. In addition, if $\tilde{\phi}$ is already close to the desired distance function, as is typically the case after advecting the level set field for only a single time step, further mitigation of spurious changes in mass is achieved by enforcing an immobile boundary condition near the zero level set. This approach is described by Son \[27\] and is easily implemented by fixing the nearest points to the interface, i.e. Eq. (2.29) is not applied to $\phi_{i,j}$ satisfying $\phi_{i,j}\phi_{p,q} \leq 0$ and $|\phi_{i,j}| \leq |\phi_{p,q}|$ for $(p,q) = (i \pm 1,j)$ or $(p,q) = (i,j \pm 1)$. Notice that if Eq. (2.29) is not applied to $\phi_{i,j}$, the subcell-fix is also effectively omitted for $\phi_{i,j}$. After iterating Eq. (2.29) to some desired convergence criteria, the level set function $\phi^{n+1}$ is updated, and the next physical time step is carried out. In the present work, we always reinitialize the level set every time step and declare convergence when the $L^2$ norm between subsequent pseudo-time iterations is

\[1\]In three spatial dimensions $H_G$ also includes $D_{z}^{\pm}$ terms.
smaller than some tolerance (taken to be $10^{-6}$ in the present work) or when a maximum number of iterations have been carried out (taken to be the maximum grid size in one direction) — whichever happens first.

We can use the signed distance property to define the material properties at cell-centers of the staggered grid. For a mesh with uniform grid spacing $\Delta x = \Delta y$, we can define a smoothed Heaviside function that has been regularized over $n_{\text{cells}}$ grid cells on either side of the interface,

$$
\tilde{H}_{i,j} = \begin{cases} 
0, & \phi_{i,j} < -n_{\text{cells}}\Delta x, \\
\frac{1}{2} \left(1 + \frac{1}{n_{\text{cells}}\Delta x} \phi_{i,j} + \frac{1}{\pi} \sin \left(\frac{\pi}{n_{\text{cells}}\Delta x} \phi_{i,j}\right)\right), & |\phi_{i,j}| \leq n_{\text{cells}}\Delta x, \\
1, & \text{otherwise}.
\end{cases}
$$

A given material property $\psi$ (such as $\rho$ or $\mu$) is then set in the whole domain via

$$
\psi_{i,j} = \psi_0 + (\psi_1 - \psi_0) \tilde{H}_{i,j},
$$

in which $\psi_0$ and $\psi_1$ denote the material property value for phases occupying $\Omega_0$ and $\Omega_1$ and we have assumed that $\Omega_0$ is represented by negative $\phi$ values (without loss of generality).

In all of the cases in the present work, we use either $n_{\text{cells}} = 1$ or 2. In general we find that for high inertia flows, 2 grid cells of smearing leads to slightly more distorted interfaces than $n_{\text{cells}} = 1$. However, $n_{\text{cells}} = 2$ leads to slightly better convergence properties for the iterative solver described in Sec. 2.4. We would not recommend using $n_{\text{cells}} \geq 3$ because, in our experience, larger smearing leads to diffuse interfaces and generates spurious vortex dynamics in the vicinity of those interfaces.
2.3.4. Surface tension force

We use the continuum surface tension model of Brackbill et al. [53] to define the volumetric surface tension force in terms of the level set field

\[
(2.32) \quad f_s = \sigma \kappa \nabla \tilde{C} = -\sigma \nabla \cdot \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right) \nabla \tilde{C},
\]

in which \( \sigma \) is the uniform surface tension coefficient, \( \kappa \) is the curvature of the interface computed directly from the signed distance function as

\[
\kappa = -\nabla \cdot n = -\nabla \cdot \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right),
\]

\( n \) is the unit normal to the surface, and \( \tilde{C} \) is a mollified version of the Heaviside function \( \tilde{H} \) that ensures the surface tension force is applied only near the zero level set. In this work we use Peskin’s four-point regularized delta function [5] to mollify the numerical Heaviside function, although other functions may also be used to smooth the transition region as described by Williams et al. [54]. Note that one could use a surface tension force function of the form \( f_s = \sigma \kappa \tilde{n} \tilde{\delta} \), in which \( \tilde{\delta} \) is a regularized version of the Dirac delta function [7]. However, this would not yield a discrete balance between the surface tension force and the pressure gradient [70], whereas Eq. (2.32) is discretely well-balanced with the pressure gradient because both \( \nabla \tilde{C} \) and \( \nabla p \) are discretized in the same manner and at the same degrees of freedom. We refer readers to the works of Brackbill et al. [53] and Williams et al. [54] for more details on the continuum surface tension model and Francois et al. [70] for its well-balanced implementation.

\[\text{This is because } \nabla \tilde{C} \approx n \tilde{\delta}.\]
2.3.5. Temporal discretization

Next, we describe the temporal discretization for both the non-conservative and conservative forms of the equations of motion. Within one time step \([t^n, t^{n+1}]\), we employ \(n_{\text{cycles}}\) cycles of fixed-point iteration to obtain an approximate solution to the fully-coupled mass transport and fluid flow problem. In this approach, the advective or convective terms and the body force are treated explicitly, and all other terms are treated implicitly. To begin time stepping, we set \(k = 0\) (\(u^{n+1,0} = u^n\), \(p^{n+\frac{1}{2},0} = p^{n-\frac{1}{2}}\), and \(\phi^{n+1,0} = \phi^n\)) and iterate until \(k = n_{\text{cycles}} - 1\). At the initial time step \(n = 0\), these quantities are obtained using the prescribed initial conditions. As described previously \([47]\), the initial value for pressure at the start of each time step \(p^{n+\frac{1}{2},0}\) does not affect the flow dynamics nor the pressure solution at the end of the time step \(p^{n+\frac{1}{2}}\); rather it serves an initial guess to iterative solution of the linear system.

Although both solvers converge for a wide range of density and viscosity contrasts, the non-conservative temporal discretization is only stable for density ratios up to \(\rho_1/\rho_0 \approx 100\). The fundamental cause for this is the discretely inconsistent transport of mass and momentum. As described by Raessi \([37]\) and, more recently, by Ghods and Herrmann \([40]\), inconsistencies in the the numerical mass and momentum fluxes used in the mass and momentum transport equations, respectively, can lead to numerical instabilities at density ratios greater than 100. This problem is especially prevalent in the level set methods, in which the phase mass is transported via an auxiliary field and no flux reconstruction is used. Numerical examples of instabilities for the non-conservative discretization of two phase flows with high density ratios will be shown in Sec. \([2,7]\).
To prevent these instabilities for high density ratios, we extend an approach described by Desjardins and Moureau \[39\], in which consistent transport was achieved by solving an additional mass balance equation, Eq. (2.3), and by ensuring that the same numerical mass flux is used for both mass and momentum transport. A discretization of the conservative form of Eqs. (2.1) and (2.2) is necessary to achieve discrete conservation and flux compatibility. The scheme described by Desjardins and Moureau was first-order accurate and diffusive (see Sec. 2.6.7). We extend this scheme to achieve at least second-order accuracy in velocity, pressure, and density.

2.3.5.1. Non-conservative discretization. During the time interval \([t^n, t^{n+1}]\), for the non-conservative discretization the time stepping proceeds as follows:

1. Advect the signed distance function \(\phi\):

   The level set function is updated by discretizing Eq. (2.9) via

   \[
   \frac{\phi^{n+1,k+1} - \phi^n}{\Delta t} + Q \left( u^{n+\frac{1}{2},k}_\text{adv}, \phi^{n+\frac{1}{2},k}_\text{lim} \right) = 0, \tag{2.33}
   \]

   in which \(Q \left( u^{n+\frac{1}{2},k}_\text{adv}, \phi^{n+\frac{1}{2},k}_\text{lim} \right) \approx \left[ \nabla \cdot \left( u^{n+\frac{1}{2},k}_\text{adv}, \phi^{n+\frac{1}{2},k}_\text{lim} \right) \right]_{i,j}\) is an explicit xsPPM7-limited approximation to the linear advection term on cell centers. The mid-point velocity and level set are given by \(u^{n+\frac{1}{2},k} = \frac{1}{2} (u^{n+1,k} + u^n)\), and \(\phi^{n+\frac{1}{2},k} = \frac{1}{2} (\phi^{n+1,k} + \phi^n)\). Here, the subscript “adv” indicates the staggered grid velocity on faces of cell-centered control volume, and the subscript “lim” indicates the limited value also defined on faces of the cell-centered control volume.

2. Reset the material properties \(\rho\) and \(\mu\):

   The density and viscosity in the computational domain are determined from the
signed distance function via

\begin{align}
\rho^{n+1,k+1} &= \rho_0 + (\rho_1 - \rho_0) \tilde{H}(\phi^{n+1,k+1}), \\
\mu^{n+1,k+1} &= \mu_0 + (\mu_1 - \mu_0) \tilde{H}(\phi^{n+1,k+1}),
\end{align}

in which \( \rho_i \) and \( \mu_i \) denote the density and viscosity for the two fluids, \( i = 0 \) or \( 1 \).

The regularized Heaviside function is given by Eq. \((2.30)\). Note that although the density and viscosity computed here are cell-centered, the notations \( \rho^{n+1,k+1} \) and \( \mu^{n+1,k+1} \) are also used to denote the interpolated material properties (to faces for density; to nodes in 2D, or edges in 3D for viscosity), as needed.

(3) Solve the incompressible Navier-Stokes equations for \( \mathbf{u} \) and \( p \):

The velocity and pressure are computed from the discretization of the non-conservative fluid momentum and continuity equations

\begin{align}
\rho^{n+1,k+1} \left( \frac{\mathbf{u}^{n+1,k+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}(\mathbf{u}_{\text{adv}}^{n+\frac{1}{2},k}, \mathbf{u}_{\text{lim}}^{n+\frac{1}{2},k}) \right) &= -Gp^{n+\frac{1}{2},k+1} + (L_{\mu} \mathbf{u})^{n+\frac{1}{2},k+1} + \mathbf{f}^{n+\frac{1}{2}}, \\
\mathbf{D} \cdot \mathbf{u}^{n+1,k+1} &= 0,
\end{align}

in which

\begin{align}
\mathbf{N}(\mathbf{u}_{\text{adv}}^{n+\frac{1}{2},k}, \mathbf{u}_{\text{lim}}^{n+\frac{1}{2},k}) &\approx \left[ \nabla \cdot \left( \mathbf{u}_{\text{adv}}^{n+\frac{1}{2},k} \cdot \mathbf{u}_{\text{lim}}^{n+\frac{1}{2},k} \right) \right]_{i-\frac{1}{2},j} - \left[ \nabla \cdot \left( \mathbf{u}_{\text{adv}}^{n+\frac{1}{2},k} \cdot \mathbf{v}_{\text{lim}}^{n+\frac{1}{2},k} \right) \right]_{i,j-\frac{1}{2}},
\end{align}

is an explicit CUI-limited approximation to the nonlinear convection term and
\begin{align*}
(L_{\mu} u)^{n+\frac{1}{2},k+1} &= \frac{1}{2} \left[(L_{\mu} u)^{n+1,k+1} + (L_{\mu} u)^n\right]
\end{align*}
is a semi-implicit approximation to the viscous strain rate. Here, the subscript “adv” indicates the interpolated advective velocity on the faces of face-centered control volume, and the subscript “lim” indicates the convective-limited value, as defined by Eq. (2.25). The above time-stepping scheme with \( n_{\text{cycles}} = 2 \) is similar to a combination of Crank-Nicolson for the viscous terms and explicit midpoint rule for the convective term, making it second-order accurate in time. Notice that this scheme is semi-implicit in that explicit approximations are used for \( \rho \) and \( \mu \) as well as the advective term.

After \( n_{\text{cycles}} \) cycles of fixed-point iteration, the final numerical solutions are given by
\begin{align*}
\begin{array}{ll}
\mathbf{u}^{n+1} &= \mathbf{u}^{n+1,n_{\text{cycles}}}, \\
p^{n+\frac{1}{2}} &= p^{n+\frac{1}{2},n_{\text{cycles}}}, \\
\phi^{n+1} &= \phi^{n+1,n_{\text{cycles}}}.
\end{array}
\end{align*}
We employ \( n_{\text{cycles}} = 2 \) cycles of fixed-point iteration for all the numerical examples, which yields second-order spatio-temporal accuracy. Numerical experiments indicate that the largest stable advective CFL number for the above scheme with \( n_{\text{cycles}} = 2 \) is 0.5.

\textbf{2.3.5.2. Conservative discretization.} During the time interval \([t^n, t^{n+1}]\), the time stepping proceeds as follows for the \underline{discretely consistent, conservative} discretization:

\begin{enumerate}
\item Advect the signed distance function \( \phi \):
\end{enumerate}

The level set function is updated by discretizing Eq. (2.9) via
\begin{equation}
\frac{\phi^{n+1,k+1} - \phi^n}{\Delta t} + Q \left( \mathbf{u}^{n+\frac{1}{2},k}_{\text{adv}}, \phi^{n+\frac{1}{2},k}_{\text{lim}} \right) = 0,
\end{equation}
which is the same update used for the non-conservative discretization described earlier.
(2) Reset the material property $\mu$:

The viscosity, but not the density, in the computational domain is determined from the signed distance function via

$$\mu^{n+1,k+1} = \mu_0 + (\mu_1 - \mu_0) \tilde{H}(\phi^{n+1,k+1}).$$

(2.40)

(3) Advect the face-centered density and compute convective derivative $C$:

We next update the density and compute the nonlinear convective term in a way that ensures discrete consistency in the mass and momentum fluxes. First, we solve a discretized density update equation on faces of the staggered grid using the third-order accurate strong stability preserving Runge-Kutta (SSP-RK3) time integrator [42]

$$\hat{\rho}^{(1)} = \hat{\rho}^n - \Delta t R(u_{adv}^n, \hat{\rho}_{lim}^n),$$

(2.41)

$$\hat{\rho}^{(2)} = \frac{3}{4} \hat{\rho}^n + \frac{1}{4} \hat{\rho}^{(1)} - \frac{1}{4} \Delta t R(u_{adv}^{(1)}, \hat{\rho}_{lim}^{(1)}),$$

(2.42)

$$\hat{\rho}^{n+1,k+1} = \frac{1}{3} \hat{\rho}^n + \frac{2}{3} \hat{\rho}^{(2)} - \frac{2}{3} \Delta t R(u_{adv}^{(2)}, \hat{\rho}_{lim}^{(2)}),$$

(2.43)

in which $R(u_{adv}, \hat{\rho}_{lim}) \approx \left[ (\nabla \cdot (u_{adv} \rho_{lim}^i))_{i-1,j}, (\nabla \cdot (u_{adv} \rho_{lim}^j))_{i,j-1} \right]$ is an explicit CUI-limited approximation to the linear density advection term. In contrast with the non-conservative form, the scalar density variable is defined and directly evolved on faces of the staggered grid. Hence, we distinguish $\hat{\rho}$, the face-centered density obtained via the SSP-RK3 integrator, from $\rho$, the face-centered density that is reset from the level set fields. Here, the subscript “adv” indicates the interpolated advective velocity on the faces of face-centered control volume,
and the subscript “lim” indicates the limited value, as defined by Eq. (2.25). Note that this time integration procedure is occurring within the overall fixed-point iteration scheme. We have found it to be crucial to use appropriately interpolated and extrapolated velocities to maintain the accuracy of the scheme. To wit, for the first cycle \((k = 0)\), the velocities are

\begin{align*}
\mathbf{u}^{(1)} &= 2\mathbf{u}^n - \mathbf{u}^{n-1}, \\
\mathbf{u}^{(2)} &= \frac{3}{2}\mathbf{u}^n - \frac{1}{2}\mathbf{u}^{n-1}.
\end{align*}

For all remaining cycles \((k > 0)\), the velocities are

\begin{align*}
\mathbf{u}^{(1)} &= \mathbf{u}^{n+1,k}, \\
\mathbf{u}^{(2)} &= \frac{3}{8}\mathbf{u}^{n+1,k} + \frac{3}{4}\mathbf{u}^n - \frac{1}{8}\mathbf{u}^{n-1}.
\end{align*}

Notice that \(\mathbf{u}^{(1)}\) is an approximation to \(\mathbf{u}^{n+1}\), and \(\mathbf{u}^{(2)}\) is an approximation to \(\mathbf{u}^{n+\frac{1}{2}}\). Similarly, \(\hat{\rho}^{(1)}\) is an approximation to \(\hat{\rho}^{n+1}\), and \(\hat{\rho}^{(2)}\) is an approximation to \(\hat{\rho}^{n+\frac{1}{2}}\).

4. Solve the incompressible Navier-Stokes equations for \(\mathbf{u}\) and \(p\):

Using the previously computed density and convective term, the velocity and
pressure are computed from the discretization of the conservative fluid momentum and continuity equations

\[
\frac{\tilde{\rho}^{n+1,k+1} u^{n+1,k+1} - \tilde{\rho}^n u^n}{\Delta t} + C \left( u_{\text{adv}}^{(2)} \tilde{\rho}_{\text{lim}}^{(2)} u_{\text{lim}}^{(2)} \right) = -G p^{n+\frac{1}{2},k+1} + (L_{\mu} u)^{n+\frac{1}{2},k+1} + f^{n+\frac{1}{2}},
\]

(2.48)

\[
D \cdot u^{n+1,k+1} = 0,
\]

(2.49)
in which the approximation to the convective derivative is given by

\[
C \left( u_{\text{adv}}^{(2)} \tilde{\rho}_{\text{lim}}^{(2)} u_{\text{lim}}^{(2)} \right) \approx \left[ \nabla \cdot \left( u_{\text{adv}}^{(2)} \tilde{\rho}_{\text{lim}}^{(2)} u_{\text{lim}}^{(2)} \right) \right]_{i-\frac{1}{2},j} - \left[ \nabla \cdot \left( u_{\text{adv}}^{(2)} \tilde{\rho}_{\text{lim}}^{(2)} u_{\text{lim}}^{(2)} \right) \right]_{i,j-\frac{1}{2}},
\]

(2.50)
and uses the same velocity \( u_{\text{adv}}^{(2)} \) and density \( \tilde{\rho}_{\text{lim}}^{(2)} \) used to update \( \tilde{\rho}^{n+1,k+1} \) in Eq (2.43). This is the key requirement for consistent mass and momentum transport.

Results presented in Sec. 2.7 demonstrate that the consistent discretization is stable for density ratios of at least \( 10^6 \) and produce significantly more accurate results than the inconsistent discretization for realistic two phase flow simulations.

For the conservative form as written, the density evolves along with the velocity and pressure at all times, with an initial value of \( \tilde{\rho} \) directly specified on the cell faces. We note the we can instead synchronize the face-centered density via the signed distance function (averaged to faces) at time step \( n \), i.e., \( \tilde{\rho}^n = \rho_0 + (\rho_1 - \rho_0) \tilde{H} (\phi^n) \), while still maintaining numerical stability. When resetting the mass density in each time step, the new density \( \tilde{\rho}^{n+1,k+1} \) (used in solving for \( u \) and \( p \)) is still obtained via the SSP-RK3 update; however
it is discarded at the end of the time step. When density synchronization is enabled, \( \rho^n \) is used in place of \( \hat{\rho}^n \) wherever a density field is needed at time level \( n \) in Eqs. (2.41) – (2.43) and Eq. (2.48). Finally, we emphasize that using a level-set synchronized density at time level \( n \) is recommended to avoid significant distortions in the interface for high density ratio flows, which are generated by accumulation of errors in advecting \( \hat{\rho} \) over the course of the simulation. Sec. 2.6.3 investigates differences between evolving and resetting the density field, and all of the numerical examples in Sec. 2.7 use density synchronization.

Note that in both Eqs. (2.36) and (2.48) we only considered the volumetric body force term \( f^{n+\frac{1}{2}} \). One can approximate the surface tension force \( f_s^{n+\frac{1}{2}} \) as a function of level set field by \( f_s(\phi^{n+\frac{1}{2},k+1}) \), in which \( \phi^{n+\frac{1}{2},k+1} = \frac{1}{2}(\phi^{n+1,k+1} + \phi^n) \).

Finally, as discussed previously, there is no guarantee that the level set function will remain a signed distance function under advection by an external velocity field. Thus, at the beginning of each time step, the reinitialization procedure described in Section 2.3.3 is used. Reinitialization is required to accurately evaluate regularized Heaviside functions near the interface, as needed both to determine material properties and to evaluate interfacial forces related to surface tension.

### 2.3.6. Adaptive mesh refinement

Some cases presented in this work use a structured adaptive mesh refinement (SAMR) framework to discretize the equations of motion. These discretization approaches describe the computational domain as composed of multiple grid levels, which together form a grid hierarchy. Assuming uniform and isotropic mesh refinement, a grid hierarchy with \( \ell \) levels

\[^{3}\text{This justifies the additional computational cost incurred by level set and volume-of-fluid methods.}\]
and with a grid spacing $\Delta x_0$, $\Delta y_0$, and $\Delta z_0$ on the coarsest grid level has grid spacings $\Delta x_{\min} = \Delta x_0/n_{\text{ref}}^{\ell-1}$, $\Delta y_{\min} = \Delta y_0/n_{\text{ref}}^{\ell-1}$, and $\Delta z_{\min} = \Delta z_0/n_{\text{ref}}^{\ell-1}$ on the finest grid level, in which $n_{\text{ref}}$ is the integer refinement ratio between levels. (Although not considered here, both the numerical method and software implementation allow for general refinement ratios.)

The locally refined meshes can be static, in that they occupy a fixed region in the domain $\Omega$, or adaptive, in that some criteria of interest is used to “tag” coarse cells for refinement. In our current implementation, cells are refined based on two criteria: 1) if the local magnitude of vorticity $\|\omega\|_{i,j} = \|\nabla \times u\|_{i,j}$ exceeds a relative threshold and 2) if the signed distance function $\phi_{i,j}$ is within some threshold of zero. This ensures that the important dynamics (e.g., regions of high velocity gradients or the multiphase interfaces) are always approximated using appropriate mesh spacings.

Each time the grid hierarchy is regenerated, quantities must be transferred from the old grid hierarchy to the new grid hierarchy. In newly refined regions, the fluid velocity $u$ is interpolated from the old coarse grid using a conservative, discretely divergence- and curl-preserving interpolation scheme [71]. Similarly, the level set $\phi$ and material properties $\rho$ and $\mu$ are interpolated from the old coarse grid using a conservative linear interpolation scheme, which ensures that positivity is maintained for density and viscosity. The pressure $p$ is interpolated using a simple non-conservative linear interpolation scheme as it is only used as an initial approximation to the updated pressure. In newly coarsened regions, all of the quantities are defined as conservative averages of the old fine-grid data. These interpolations are used to define ghost cell values at the coarse-fine interface, which enables composite-grid approximations to the linear operators $D$, $G$, $L_\mu$, and $L_\rho$. 
described earlier in Sec. 2.3.1. We refer readers to prior work by Griffith [72] for additional details on the AMR discretization methods.

2.3.7. Discretization of the viscous term in three spatial dimensions

The numerical treatment of the strain rate tensor $\nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)]$ in three spatial dimensions warrants additional discussion. For the velocity field

$$u(x, t) = (u(x, t), v(x, t), w(x, t)),$$

the continuous strain rate tensor form of the viscous term is

$$\nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] = \begin{bmatrix}
2 \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial u}{\partial z} + \mu \frac{\partial w}{\partial x} \right) \\
2 \frac{\partial}{\partial y} \left( \mu \frac{\partial v}{\partial y} \right) + \frac{\partial}{\partial x} \left( \mu \frac{\partial v}{\partial x} + \mu \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right) \\
2 \frac{\partial}{\partial z} \left( \mu \frac{\partial w}{\partial z} \right) + \frac{\partial}{\partial x} \left( \mu \frac{\partial w}{\partial x} + \mu \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial w}{\partial y} + \mu \frac{\partial v}{\partial z} \right)
\end{bmatrix},$$

which leads to a discretization of the form

$$L_\mu \mathbf{u} = \begin{bmatrix}
(L_\mu u)^x_{i-\frac{1}{2}, j, k} \\
(L_\mu u)^y_{i, j-\frac{1}{2}, k} \\
(L_\mu u)^z_{i, j, k-\frac{1}{2}}
\end{bmatrix}.$$

The two-dimensional discretization scheme required viscosity on both Cartesian cell centers, where $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial y}$ are naturally approximated, and nodes, where $\frac{\partial u}{\partial y}$ and $\frac{\partial v}{\partial x}$ are naturally approximated. In three spatial dimensions, the off-diagonal components of
\[ \mu (\nabla u + \nabla u^T) \] are naturally defined on the edges of the grid cells,

\[ \begin{align*}
\left( \mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x} \right)_{i-\frac{1}{2}, j-k-\frac{1}{2} \Delta x} &= \mu_{i-\frac{1}{2}, j-k-\frac{1}{2}} \left( \frac{u_{i-\frac{1}{2}, j,k} - u_{i-\frac{1}{2}, j-1,k}}{\Delta y} + \frac{v_{i,j-\frac{1}{2},k} - v_{i-1,j-\frac{1}{2},k}}{\Delta x} \right), \\
\left( \mu \frac{\partial u}{\partial z} + \mu \frac{\partial w}{\partial x} \right)_{i-\frac{1}{2}, j,k-\frac{1}{2} \Delta x} &= \mu_{i-\frac{1}{2}, j,k-\frac{1}{2}} \left( \frac{u_{i-\frac{1}{2}, j,k} - u_{i-\frac{1}{2}, j-1,k}}{\Delta z} + \frac{v_{i,j,k-\frac{1}{2},k} - v_{i-1,j,k-\frac{1}{2},k}}{\Delta x} \right), \\
\left( \mu \frac{\partial v}{\partial z} + \mu \frac{\partial w}{\partial y} \right)_{i, j-\frac{1}{2}, k-\frac{1}{2} \Delta y} &= \mu_{i, j-\frac{1}{2}, k-\frac{1}{2}} \left( \frac{v_{i,j-\frac{1}{2},k} - v_{i-1,j-\frac{1}{2},k}}{\Delta z} + \frac{w_{i,j,k-\frac{1}{2},k} - w_{i-1,j,k-\frac{1}{2},k}}{\Delta y} \right).
\end{align*} \]

Fig. 2.4 shows the location of these \(xy-, xz-,\) and \(yz-\)edges. The full discretization of the \(x\)-component of the strain rate tensor reads,

\[ (L_\mu u)^x_{i-\frac{1}{2}, j,k} = \frac{2}{\Delta x} \left[ \mu_{i,j,k} \frac{u_{i+\frac{1}{2}, j,k} - u_{i-\frac{1}{2}, j,k}}{\Delta x} - \mu_{i-\frac{1}{2}, j,k} \frac{u_{i-\frac{1}{2}, j,k} - u_{i-1, j,k}}{\Delta x} \right] + \frac{1}{\Delta y} \left[ \mu_{i-\frac{1}{2}, j+\frac{1}{2}, k} \frac{u_{i-\frac{1}{2}, j+1,k} - u_{i-\frac{1}{2}, j,k}}{\Delta y} - \mu_{i-\frac{1}{2}, j,k} \frac{u_{i-\frac{1}{2}, j,k} - u_{i-\frac{1}{2}, j-1,k}}{\Delta y} \right] + \frac{1}{\Delta y} \left[ \mu_{i-\frac{1}{2}, j+\frac{1}{2}, k} \frac{v_{i,j+\frac{1}{2},k} - v_{i, j+1,k}}{\Delta x} - \mu_{i-\frac{1}{2}, j,k} \frac{v_{i,j+\frac{1}{2},k} - v_{i-1,j+\frac{1}{2},k}}{\Delta x} \right] + \frac{1}{\Delta z} \left[ \mu_{i-\frac{1}{2}, j,k+\frac{1}{2}} \frac{u_{i-\frac{1}{2}, j,k+1} - u_{i-\frac{1}{2}, j,k}}{\Delta z} - \mu_{i-\frac{1}{2}, j,k} \frac{u_{i-\frac{1}{2}, j,k} - u_{i-\frac{1}{2}, j,k-1}}{\Delta z} \right] + \frac{1}{\Delta z} \left[ \mu_{i-\frac{1}{2}, j,k+\frac{1}{2}} \frac{w_{i,j,k+\frac{1}{2}} - w_{i-1,j,k+\frac{1}{2}}}{\Delta x} - \mu_{i-\frac{1}{2}, j,k} \frac{w_{i,j,k+\frac{1}{2}} - w_{i-1,j,k-\frac{1}{2}}}{\Delta x} \right], \]

which again uses cell centered viscosities and requires approximations to the viscosity on edges. The edge viscosities can be computed using arithmetic or harmonic averages of cell centered values, i.e., \(\mu_{i-\frac{1}{2}, j, k-\frac{1}{2}}\) would be computed from \(\mu_{i-1,j-1,k}, \mu_{i,j-1,k}, \mu_{i-1,j,k},\) and \(\mu_{i,j,k}\). Approximations to the remaining components in Eq. (2.53) can be determined.
Figure 2.4. Two adjacent 3D staggered grid cells on which the viscous component \( (L\mu u)_x \) is computed on a particular face (\( x \), red) using velocity derivatives and viscosity on cell centers (\( \bullet \), black), \( xy \)-edges (\( \nabla \), orange), and \( xz \)-edges (\( \Delta \), green). The \( yz \)-edges (\( \ast \), blue) are not used in the computation of \( (L\mu u)_x \), but would be required to compute \( (L\mu u)_y \) and \( (L\mu u)_z \) analogously. The other spatial discretizations described in Section 2.3 can be straightforwardly extended to three dimensions.

2.3.8. Physical boundary conditions

This section details the physical boundary condition treatment used in this work. The treatment of physical boundary conditions has been described in detail by Griffith for the constant-coefficient case [47], although we note that the imposition of normal traction boundary conditions used in this work differs slightly. We follow the same strategy as in Griffith [47] and restrict our attention to the vicinity of a single grid cell \( (N-1, j) \), \( 0 \leq j < N \) on the right side of the physical domain (Fig. 2.5). The treatment along other physical domain boundaries and in three dimensions is analogous.

2.3.8.1. Scalar boundary conditions. For the scalar fields, namely density \( \rho \), viscosity \( \mu \), and the signed distance function \( \phi \), we restrict our attention to homogenous Neumann
boundary conditions. For a given scalar field $\psi$, the boundary condition is

\begin{equation}
\frac{\partial \psi}{\partial x} (x_{N-\frac{1}{2}}, t) = 0,
\end{equation}

which is imposed using the standard ghost value treatment [61],

\begin{equation}
\psi^n_{N,j} = \psi^n_{N-1,j}.
\end{equation}

In cases where $n_G > 1$ ghost cell values are required for $\psi$, the condition is imposed by setting the $k^{th}$ ghost cell with the $k^{th}$ interior value, i.e., $\psi_{N+k-1} = \psi_{N-k}$ for $k = 1, \ldots, n_G$.

**2.3.8.2. Fluid boundary conditions.** Considering the right domain boundary once again, the outward unit normal is $\mathbf{n} = (1, 0)$, and the unit tangent vector is $\mathbf{\tau} = (0, 1)$. The viscous stress tensor for an incompressible fluid is

\begin{equation}
\sigma = -p\mathbf{I} + \mu \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right].
\end{equation}
Following Griffith [47], four types of physical boundary conditions for \( u \) and \( \sigma \) are considered in this work:

1. The normal velocity is prescribed at face \( x_{N-\frac{1}{2},j} \):
   
   Suppose that the normal velocity at position \( x_{N-\frac{1}{2},j} \) is given by \( u_{N-\frac{1}{2},j}^{\text{norm}}(t) \). The treatment of this boundary condition is straightforward because we can simply set

   \[
   u_{N-\frac{1}{2},j}^n = u_{N-\frac{1}{2},j}^{\text{norm}}(t^n).
   \]

   As in Griffith [47], no ghost values are required for \( p_{N,j} \) or \( u_{N+\frac{1}{2},j} \), nor may any pressure condition be prescribed at this boundary.

2. The tangential velocity is prescribed at node \( x_{N-\frac{1}{2},j-\frac{1}{2}} \):
   
   Suppose that the tangential velocity at position \( x_{N-\frac{1}{2},j-\frac{1}{2}} \) is given by \( v_{N-\frac{1}{2},j-\frac{1}{2}}^{\text{tan}}(t) \). This boundary condition is imposed via a linear fit of the prescribed velocity and the closest internal value,

   \[
   v_{N,j-\frac{1}{2}}^n = 2v_{N-\frac{1}{2},j-\frac{1}{2}}^{\text{tan}}(t^n) - v_{N-1,j-\frac{1}{2}}^n.
   \]

3. The tangential traction is prescribed at node \( x_{N-\frac{1}{2},j-\frac{1}{2}} \):
   
   Suppose that the tangential traction, \( (\tau \cdot \sigma \cdot n) = \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \), is given by \( F_{N-\frac{1}{2},j-\frac{1}{2}}^{\text{tan}}(t) \). In contrast with prior work [47], the spatially varying viscosity
must be treated when imposing this boundary condition. Writing out a second-order finite difference approximation to this boundary condition at time $t^n$ yields

\begin{equation}
\mu_{N-\frac{1}{2},j-\frac{1}{2}}^{n} \left( \frac{u_{N,j-\frac{1}{2}}^{n} - u_{N-1,j-\frac{1}{2}}^{n}}{\Delta x} + \frac{u_{N-\frac{1}{2},j}^{n} - u_{N-\frac{1}{2},j-1}^{n}}{\Delta y} \right) = F_{N-\frac{1}{2},j-\frac{1}{2}}^{n}(t^n),
\end{equation}

which is an expression for the ghost velocity $v_{N,j-\frac{1}{2}}^{n}$ in terms of the interior values.

Note that in the above expression the node-centered viscosity value is needed, which is already being computed and used for the discretization of the viscous operator (Eqs. (2.17) and (2.18)).

(4) The normal traction is prescribed at face $x_{N-\frac{1}{2},j}$:

Suppose that the normal traction, $(\mathbf{n} \cdot \sigma \cdot \mathbf{n}) = -p + 2 \mu \frac{\partial u}{\partial x}$, is given by $F_{N-\frac{1}{2},j}^{\text{norm}}(t^n)$. Again, the spatially varying viscosity must be handled when imposing this boundary condition. This boundary condition requires ghost values $u_{N+\frac{1}{2},j}$ and $p_{N,j}$ to be set, which implies a second condition must be enforced in order to uniquely specify these values. As in prior work [47], we enforce the divergence-free condition $\nabla \cdot \mathbf{u}(x_{N,j},t) = 0$ in the ghost cell surrounding $x_{N,j}$. A second-order finite difference approximation to this condition at time $t^n$ is

\begin{equation}
\frac{u_{N+\frac{1}{2},j}^{n} - u_{N-\frac{1}{2},j}^{n}}{\Delta x} + \frac{v_{N,j+\frac{1}{2}}^{n} - v_{N,j-\frac{1}{2}}^{n}}{\Delta y} = 0,
\end{equation}

which yields

\begin{equation}
u_{N+\frac{1}{2},j}^{n} = u_{N-\frac{1}{2},j}^{n} - \frac{\Delta x}{\Delta y} \left( v_{N,j+\frac{1}{2}}^{n} - v_{N,j-\frac{1}{2}}^{n} \right).
\end{equation}
Now that we have obtained the ghost value $u_{N+\frac{1}{2},j}$, we can write out the finite difference approximation to the traction boundary condition,

\[
-\frac{p_{N,j}^{n+\frac{1}{2}} + p_{N-1,j}^{n+\frac{1}{2}}}{2} + \mu_{N-\frac{1}{2},j}^{n+\frac{1}{2}} \left( \frac{3u_{N-\frac{1}{2},j}^{n} - 4u_{N-\frac{3}{2},j}^{n} + u_{N-\frac{5}{2},j}^{n}}{2\Delta x} + \frac{3u_{N-\frac{1}{2},j}^{n+1} - 4u_{N-\frac{3}{2},j}^{n+1} + u_{N-\frac{5}{2},j}^{n+1}}{2\Delta x} \right) = F_{N-\frac{1}{2},j}^{\text{norm}}(t^{n+\frac{1}{2}}),
\]

(2.66)

in which we have used a one-sided second-order finite difference approximation to $\frac{\partial u}{\partial x}$ at $x_{N-\frac{1}{2},j}$. The viscosity on the face $\mu_{N-\frac{1}{2},j}^{n+\frac{1}{2}}$ is computed using an arithmetic or harmonic average of the viscosity in the neighboring cells $\mu_{N-1,j}^{n+\frac{1}{2}}$ and $\mu_{N,j}^{n+\frac{1}{2}}$. Rearranging Eq. (2.66) yields an expression for the desired ghost value for pressure $p_{N,j}^{n+\frac{1}{2}}$ based on interior values.

We note that the numerical imposition of the normal traction boundary condition using (one-sided) second-order derivative here differs from the one-sided first-order approximation to $\frac{\partial u}{\partial x}$ at $x_{N-\frac{1}{2},j}$ previously used by Griffith [47].

2.4. Solution methodology

This section describes the linear solvers required to compute a solution to the fully-coupled, time-dependent incompressible Stokes system

\[
\begin{bmatrix}
\frac{1}{\Delta t} \varphi^{n+1,k+1} - \frac{1}{2} L_{\mu} u^{n+1,k+1} \\
-D \cdot & 0
\end{bmatrix}
\begin{bmatrix}
u^{n+1,k+1} \\
p^{n+\frac{1}{2},k+1}
\end{bmatrix}
= \begin{bmatrix}
f_\mu \\
0
\end{bmatrix},
\]

(2.67)

in which $\varphi^{n+1,k+1}$ is a diagonal matrix of face-centered densities corresponding to each velocity degree of freedom ($\varphi \equiv \rho$ as defined in Eq. (2.34) for the non-conservative discretization, and $\varphi \equiv \tilde{\rho}$ as defined in Eq. (2.43) for the conservative discretization), and
the right-hand side of the momentum equation is lumped into $f_u$, whose value depends on the discretization type. For non-conservative form, it is

$$f_u = \left( \frac{1}{\Delta t} \rho^{n+1,k+1} + \frac{1}{2} L \mu^\rho \right) u^n - \rho^{n+1,k+1} N \left( u^{n+\frac{1}{2},k} u^{n+\frac{1}{2},k}_l \right) + f^{n+\frac{1}{2}} + f_s^{n+\frac{1}{2}},$$

and for conservative form with density synchronization, it is

$$f_u = \left( \frac{1}{\Delta t} \rho^n + \frac{1}{2} L \mu^n \right) u^n - C \left( u^{(2)}_{adv}, \tilde{\rho}^{(2)} \right) + f^{n+\frac{1}{2}} + f_s^{n+\frac{1}{2}}.$$

The operator on the left-hand side of Eq. (2.67) is the time-dependent incompressible staggered Stokes operator. To solve this system of equations, we use a flexible GMRES (FGMRES) Krylov solver [73] preconditioned by a variable-coefficient projection method solver that is hybridized with a local-viscosity solver. The efficient preconditioner enables rapid convergence of the variable-coefficient iterative Stokes solver; between 1 and 20 Krylov iterations are observed for all cases considered here. We briefly describe the hybrid preconditioner here and refer the readers to the work of Griffith [47] and Cai et al. [48] for further details on the constant-coefficient and variable-coefficient preconditioners, respectively.

### 2.4.1. Projection preconditioner

The Stokes system can be succinctly written as

$$B \begin{bmatrix} x_u \\ x_p \end{bmatrix} = \begin{bmatrix} A & G \\ -D \cdot & 0 \end{bmatrix} \begin{bmatrix} x_u \\ x_p \end{bmatrix} = \begin{bmatrix} b_u \\ b_p \end{bmatrix},$$

where $B = \begin{bmatrix} I & -K \end{bmatrix}$.
in which \( B \) is the Stokes operator, 
\[
A = \frac{1}{\Delta t} \partial_t \delta^{n+1,k+1} - \frac{1}{2} L_{\mu}^{n+1,k+1},
\]
\( x_u \) and \( x_p \) denote the velocity and pressure degrees of freedom, and \( b_u \) and \( b_p = 0 \) denote the right-hand sides of the momentum and continuity equations, respectively. Krylov methods such as FGMRES require routines to compute the action of \( B \) on vectors, i.e. \( Bx \). Note that the matrix form of the operators \( A, G, \) and \( D \cdot \) are merely a convenience. Rather these operators are “applied” by direct evaluation of the finite difference operators defined on locally refined grids.

Saddle-point problems such as Eq. 2.70 are ill-conditioned, and effective preconditioning strategies are needed to obtain scalable Krylov methods for such equations. The particular preconditioner used in the present work is based on the fractional-step projection method, which is commonly used to solve the incompressible Navier-Stokes equations \[10, 11\] in an operator-splitting manner. The conversion of the projection method to a projection preconditioner \( P^{-1}_{proj} \) was described for constant material properties by Griffith \[47\], and for variable material properties by Cai et al. \[48\].

For theoretical completeness and to interpret solver scalability results of section 2.6.5 we briefly outline the derivation and form of \( P^{-1}_{proj} \) and provide some practical considerations for the linear solvers towards the end of the section. To begin, we first note that for the Stokes preconditioner, \( x_u \) and \( x_p \) should be interpreted as error in velocity and pressure degrees of freedom, respectively, and \( b_u \) and \( b_p \neq 0 \) should be interpreted as residuals of the momentum and continuity constraint equations, respectively. As is done in the conventional projection method, we first compute an intermediate approximation...
to $x_u$ by solving

$$A\tilde{x}_u = b_u. \tag{2.71}$$

Note that this approximation does not in general satisfy the discrete continuity equation i.e., $-D \cdot \tilde{x}_u \neq b_p$. This condition can be satisfied by introducing an auxiliary scalar field $\theta$ and writing out a fractional timestep

$$\frac{\varnothing}{\Delta t} \left( x_u - \tilde{x}_u \right) = -G\theta, \tag{2.72}$$

$$-D \cdot x_u = b_p. \tag{2.73}$$

Multiplying Eq. (2.72) by $\varnothing^{-1}$, taking the discrete divergence $D$, and substituting in Eq. (2.73) yields the density-weighted Poisson problem

$$-D \cdot \varnothing^{-1}G\theta = -L\varnothing\theta = -\frac{1}{\Delta t} \left( b_p + D \cdot \tilde{x}_u \right). \tag{2.74}$$

The updated velocity solution can be computed as

$$x_u = \tilde{x}_u - \Delta t\varnothing^{-1}G\theta. \tag{2.75}$$

In the conventional projection method for constant viscosity, the pressure solution can be computed as $x_p = \left( I - \frac{\Delta t}{2} \mu L \varnothing \right) \theta$. For preconditioning purposes, it was shown in [48] that a reasonable approximation to the pressure solution in the presence of a spatially varying viscosity can be written as

$$x_p \approx \left( I - \frac{\Delta t}{2} L \varnothing^2 \mu \right) \theta. \tag{2.76}$$
in which $\mu$ is a diagonal matrix of cell-centered viscosities corresponding to each pressure degree of freedom. The above form of Eq. (2.76) comes from an approximate Schur complement of the Stokes system Eq. (2.70) and we refer readers to [48] for more details.

Finally, the projection preconditioner can be written in matrix form as

$$
(2.77) \quad P_{\text{proj}}^{-1} = \begin{bmatrix}
I - \Delta t \sigma^{-1} G & I & 0 \\
0 & I - \frac{\Delta t}{2} L_\sigma 2\mu & 0 (-L_\sigma)^{-1} \\
0 & -\frac{1}{\Delta t} D & -\frac{1}{\Delta t} I & I
\end{bmatrix}
$$

in which $\mu$ is a diagonal matrix of cell-centered viscosities corresponding to each pressure degree of freedom. This preconditioner can be obtained from an approximate block factorization of the Stokes system; see Cai et al. [48] for further details.

There are several advantages to using the projection method as a preconditioner rather than as a solver. We summarize them here:

- The projection method is an operator splitting approach that requires the specification of artificial boundary conditions for the velocity and pressure fields. This split affects the global order of accuracy of the solution [74]. For example, consider the imposition of normal traction on domain boundary points $x_b \in \partial \Omega$, i.e., $n \cdot \sigma \cdot n = -p + 2\mu \frac{\partial u_n}{\partial n} = F(x_b, t)$ as detailed in Sec. 2.3.8. Here, $u_n$ represents the normal velocity and $F(x_b, t)$ is the imposed function (possibly of time) defined on the boundary $\partial \Omega$. The normal traction boundary condition requires a linear combination of discretized pressure and velocity variables. This combination can be accounted for in the Stokes operator $B$ directly. However, it is not possible to split the linear combination a priori into the velocity operator $A$ and pressure operator $L_\sigma$ used by the projection solver. In contrast, this artificial split (or
artificial boundary conditions) in the projection preconditioner\footnote{Since preconditioners employ homogenous versions of the boundary conditions, one can use homogenous Dirichlet conditions for pressure and homogeneous Neumann conditions for the normal velocity component. This treatment is followed in the current work.} does not affect the final solution of velocity and pressure obtained from the outer Krylov solver. A preconditioner can only affect the convergence rate of the iterative solver, not the accuracy of the converged solution.

- Projection methods are derived by assuming that certain operators commute. Typically, this assumption is only satisfied in the case of constant-coefficient operators defined on periodic computational domains. For variable-coefficient operators, there is an unavoidable commutator error that is associated with using the projection method as a solver.

- The projection method as a solver introduces large errors for zero Reynolds number incompressible Stokes flow due to splitting of the velocity and pressure solutions; see the introduction section in \cite{48} for a discussion. Stokes flow is a purely elliptic system that describes an instantaneous force equilibrium between pressure and viscous forces (and possibly other body forces), and therefore requires a simultaneous solution of velocity and pressure variables. As explained in Sec. 2.4.3, a modified projection preconditioner can be used to solve the incompressible Stokes equations simultaneously.

- Finally, using the projection method as a preconditioner is no less efficient than using it as a solver, as demonstrated by Griffith \cite{47} and Cai et al. \cite{48}. Our tests in Sec. 2.6.5 also support these previous findings.
We remark that in contrast with the constant coefficient projection preconditioner \[47\], both the Stokes operator \(B\) and the preconditioner \(P_{\text{proj}}^{-1}\) will change from time step to time step, and even cycle-to-cycle, because the density and viscosity generally will varying in both space and time. For the Krylov solver, it is vital that \(B\) be updated every time the material properties are updated. It is easy to achieve this in practice by using a matrix-free implementation of the present method. In contrast, we find that there is no need to update \(P_{\text{proj}}^{-1}\) at the same frequency. Specifically, we find that in many cases, it is computationally efficient to update \(P_{\text{proj}}^{-1}\) only every 5 to 50 time steps to avoid impairing the convergence of the FGMRES iterations.

2.4.2. Subdomain solvers

Because Krylov methods require only the evaluation of the action of the preconditioning operator, we do not actually form \(P_{\text{proj}}^{-1}\), which would be a dense matrix. Notice, however, that evaluating the action of the projection preconditioner \(P_{\text{proj}}^{-1}\) requires the solution to momentum and pressure-Poisson equations. In practice, these equations are only solved approximately by so-called subdomain solvers: 1) the velocity subdomain system, which requires the (approximate) application of \(A^{-1}\), and 2) the pressure subdomain system, which requires the (approximate) application of \(L_{\text{p}}^{-1}\). We again employ iterative methods for these inner subdomain solvers to produce approximate solutions up to a specified relative residual tolerance \(\epsilon_{\text{sub}}\). It is neither recommended nor required that tight tolerances be used. We find that for the cases considered in this work, \(\epsilon_{\text{sub}} = 10^{-2}\) is sufficient to produce a convergent outer solver that achieves a relative residual tolerance of \(\epsilon_{\text{Stokes}} = 10^{-12}\).
For Eq. (2.70) within 1 to 20 FGMRES iterations, even for highly contrasting material properties, or on highly refined grids.

For both the velocity and pressure subdomain problems, we employ a Richardson solver with a single multigrid V-cycle of an FAC (fast adaptive composite) preconditioner [75], respectively. For each multigrid level, for both the velocity and pressure problems, 3 iterations of Gauss-Seidel smoothing are used. The solvers chosen here are shown to converge with density and viscosity contrasts up to $10^6$ in Sec. 2.6. Homogeneous Dirichlet or Neumann boundary conditions, similar to those used in a projection method-based solver, are prescribed for the subdomain problems. We remark that choosing boundary conditions for the inner solvers that are incompatible with the physical boundary conditions imposed in the outer solver can cause the outer FGMRES iterations to fail to converge. More details on the choice of compatible subdomain solver boundary conditions are provided by Griffith [47]. Unlike the case of a projection method-based solver, however, the choice of these boundary conditions do not affect the final solution or the accuracy of the outer solver so long as that outer solver converges.

2.4.3. Projection preconditioner for steady Stokes flow

In Sec. 2.4.1 we outline the derivation of the projection preconditioner for the incompressible Navier-Stokes system Eq. (2.70). That derivation assumed a finite Reynolds number with nonzero inertia. In contrast, this section describes the form of the projection preconditioner for the linear, steady Stokes problem in the limit of zero Reynolds number $Re \to 0$ or infinite viscous CFL number $\beta = \frac{\mu \Delta t}{\rho \Delta x^2} \to \infty$. 
For steady Stokes flow, the inertial terms involving density in the time-dependent incompressible Navier-Stokes system vanish. With a backward Euler time stepping scheme used for the viscous Laplacian operator, the discretized linear system reads as

\[
\begin{bmatrix}
-\mu L^{n+1} & G \\
-D & 0
\end{bmatrix}
\begin{bmatrix}
u^{n+1} \\
p^{n+1}
\end{bmatrix}
= 
\begin{bmatrix}
f^{n+1} \\
0
\end{bmatrix},
\]

which corresponds to the same residual-error system described in Eq. 2.70 except with \( A = -\mu L^{n+1} \). Hence, we again solve for the intermediate approximation to \( u \) as

\[
A\tilde{x}_u = b_u.
\]

Since this approximation does not in general satisfy the discrete continuity equation i.e., \(-D \cdot \tilde{x}_u \neq b_p\), we introduce an auxiliary scalar field \( \theta \) and write out a fractional step:

\[
\begin{align*}
x_u - \tilde{x}_u &= -G\theta, \\
-D \cdot x_u &= b_p.
\end{align*}
\]

Taking the discrete divergence \( D \cdot \) of Eq. (2.80), and substituting in Eq. (2.81) yields the Poisson problem

\[
- D \cdot G\theta = -(b_p + D \cdot \tilde{x}_u).
\]

The updated velocity solution is then computed as

\[
x_u = \tilde{x}_u - G\theta,
\]
and a reasonable approximation to the pressure solution in the presence of a spatially
varying viscosity can be written as

\[
\mathbf{x}_p \approx (\mathbf{I} - (\mathbf{D} \cdot \mathbf{G})2\mu) \theta.
\]

Finally, the projection preconditioner for steady Stokes flow can be written in matrix form as

\[
\mathbf{P}^{-1}_{proj} = \begin{bmatrix}
\mathbf{I} & -\mathbf{G} \\
0 & \mathbf{I} - (\mathbf{D} \cdot \mathbf{G})2\mu
\end{bmatrix}
\begin{bmatrix}
\mathbf{I} & 0 \\
0 & (-\mathbf{D} \cdot \mathbf{G})^{-1}
\end{bmatrix}
\begin{bmatrix}
\mathbf{I} & 0 \\
-D \cdot -\mathbf{I}
\end{bmatrix}
\begin{bmatrix}
\mathbf{A}^{-1} & 0 \\
0 & \mathbf{I}
\end{bmatrix}.
\]

2.5. Software implementation

The algorithms and fluid solver described here are implemented in the IBAMR li-
brary [14], which is open-source C++ simulation software focused on immersed bound-
ary methods with adaptive mesh refinement. All of the numerical examples presented
here are publicly available via [https://github.com/IBAMR/IBAMR](https://github.com/IBAMR/IBAMR). IBAMR relies on
SAMRAI [15, 16] for Cartesian grid management and the AMR framework. Linear and
nonlinear solver support in IBAMR is provided by the PETSc library [17, 18, 19].

All of the example cases in the present work made use of distributed-memory par-
allelism using the Message Passing Interface (MPI) library. Various computing environ-
ments were used to execute the simulations, including:

- a personal desktop computer comprised of 16 cores (processors) and 24 GB total
  memory;
• the Quest cluster at Northwestern University, which is comprised of 679 compute nodes (16,028 cores) interconnected by an InfiniBand network and 128 GB memory per node;
• the XSEDE Bridges cluster at the Pittsburgh Supercomputing Center, which contains 752 Regular Shared Memory nodes (21,056 cores) and 128 GB memory per node.

Between 1 and 128 processors were used in all the cases described here.

2.6. Solver accuracy and performance

This section investigates the accuracy and convergence rates for the multiphase flow solver, in both non-conservative and conservative form, with various choices of boundary conditions. In all cases, the computed solution from the solver is compared against manufactured solutions on uniform and locally refined grids. We also demonstrate the robustness and scalability of the flow solver for highly contrasting densities and viscosities and also under grid refinement in this section. We employ $n_{cycles} = 2$ cycles of fixed-point iteration for all cases considered in this section. To assess the order of accuracy, errors in the cell centered pressure $p$ are computed using standard formulae for the $L^1$ and $L^\infty$ norms

$$
\|p\|_1 = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} |p_{i,j}| \Delta x \Delta y,
$$

(2.86)

$$
\|p\|_\infty = \max_{x_{i,j} \in \Omega} |p_{i,j}|.
$$

(2.87)
The $L^1$ error in the face centered velocity $u$ is computed using a simple modification for the $L^1$ norms described in [47], which ensures that $\| (1, 0) \|_1 = \| (0, 1) \|_1 = 1$; the $L^1$ norm is computed as $\| u \|_1 = \| (u, v) \|_1 = \| u \|_1 + \| v \|_1$, with

$$\| u \|_1 = \frac{1}{2} \sum_{j=0}^{N_y-1} |u_{\frac{1}{2}, j}| \Delta x \Delta y + \sum_{i=1}^{N_x-1} \sum_{j=0}^{N_y-1} |u_{i-\frac{1}{2}, j}| \Delta x \Delta y + \frac{1}{2} \sum_{j=0}^{N_y-1} |u_{N_x-\frac{1}{2}, j}| \Delta x \Delta y,$$

(2.88)

$$\| v \|_1 = \frac{1}{2} \sum_{i=0}^{N_x-1} |v_{i, \frac{1}{2}}| \Delta x \Delta y + \sum_{j=1}^{N_y-1} \sum_{i=0}^{N_x-1} |v_{i, j-\frac{1}{2}}| \Delta x \Delta y + \frac{1}{2} \sum_{i=0}^{N_x-1} |v_{i, N_y-\frac{1}{2}}| \Delta x \Delta y.$$

(2.89)

Finally, the $L^\infty$ norm of the face centered velocity is computed as $\| u \|_\infty = \| (u, v) \|_\infty = \max (\| u \|_\infty, \| v \|_\infty)$, with

$$\| u \|_\infty = \max_{x, y} \left| u_{x, y} \right|,$$

(2.90)

$$\| v \|_\infty = \max_{x, y} \left| v_{x, y} \right|.$$

(2.91)

The presented convergence rates are essentially the same independent of the time at which these errors are evaluated. Unless otherwise stated, a grid size of $N \times N$ is used to discretize the computational domain and a relative convergence tolerance of $\epsilon_{\text{Stokes}} = 10^{-12}$ is specified for the FGMRES solver in all of these cases.

For many of the manufactured solutions considered in this section, a smoothing parameter $\delta$ is used to ensure that the density and viscosity fields are transitioned from low values to high values over a spatial region of constant width. The specific values of $\delta$ is chosen to ensure that the coarsest grid in each convergence study is able to resolve
this region over at least 1.5 grid cells. Moreover, a constant spatial transition width ensures that the same exact analytical solution is considered on all grids, which makes the order-of-accuracy results consistent.

2.6.1. Non-conservative form: Effect of boundary conditions

We begin by considering the non-conservative set of equations, (2.4)–(2.5). A manufactured solution for velocity and pressure is given by

\begin{align}
(2.92) \quad u(x, t) &= 2\pi \cos(2\pi x) \cos(2\pi t - 2\pi y), \\
(2.93) \quad v(x, t) &= -2\pi \sin(2\pi x) \sin(2\pi t - 2\pi y) - \sin(2\pi t - 2\pi x), \\
(2.94) \quad p(x, t) &= -2\pi \sin(2\pi t - 2\pi x) \cos(2\pi t - 2\pi y),
\end{align}

together with time-independent density and viscosity fields of the form

\begin{align}
(2.95) \quad \rho(x) &= \rho_0 + \frac{\rho_1}{2} \left( \tanh \left( \frac{0.1 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{\delta} \right) + 1 \right), \\
(2.96) \quad \mu(x) &= \mu_0 + \mu_1 + \mu_1 \sin(2\pi x) \cos(2\pi y),
\end{align}

The density, viscosity, and initial velocity fields are shown in Fig. 2.6. Plugging Eqs. (2.92)-(2.96) into the non-conservative momentum equation (2.4) yields a forcing term \( f(x, t) \) that produces the specified solution. We set \( \rho_0 = 1, \rho_1 = -\rho_0 + 10^3, \mu_0 = 10^{-4}, \) and \( \mu_1 = -\mu_0 + 10^{-2}, \) which yields variations in density and viscosity are set to be similar to that of air and water. The computational domain is the unit square, \( \Omega = [0, L]^2 = [0, 1]^2. \) The smoothing parameter is set to \( \delta = 0.05L. \)
We impose periodic boundary conditions in the $x$-direction and various boundary conditions in the $y$-direction. The boundary conditions in the $y$-direction are:

- specified normal and tangential velocities, denoted as “vel-vel”;
- specified normal velocity with specified tangential traction, denoted as “vel-tra”;
- specified normal traction with specified tangential velocity, denoted as “tra-vel”;
- specified normal and tangential tractions, denoted as “tra-tra”;
- periodic boundary conditions denotes as “periodic”.

The maximum velocities in the domain for this manufactured solution are $\mathcal{O}(1)$, hence a relevant time scale is $L/U$ with $U = 1$. Errors in the velocity and pressure are computed at time $T = tU/L = 0.1$ with a uniform time step $\Delta t = 1/(15.625N)$, which yields an approximate CFL number of 0.5. Figs. 2.7–2.11 show the discrete $L^1$ and $L^\infty$ errors for velocity and pressure as a function of grid size. Second-order convergence rates are
observed for both velocity and pressure in both norms. We note that at lower resolutions, the pointwise convergence rates at coarser resolutions are less than two for vel-vel boundary conditions (see Fig. 2.8(a)), although second-order convergence rates are ultimately obtained on finer grids. We remark that reduction in accuracy at lower resolution is observed for small base viscosity; at higher (base) viscosities, we see second-order convergence rates at all the resolutions considered for the vel-vel boundary conditions (data not shown). This is also consistent with what was observed for the unsplit discretization of the constant coefficient incompressible Navier-Stokes equations in Griffith [47]. These tests show that the present numerical discretization and boundary treatment maintains the pointwise second-order accuracy for a variety of physical boundary conditions.
Figure 2.8. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the non-conservative manufactured solution with specified normal and tangential velocity (vel-vel) boundary conditions: (a) convergence rate for $u$; (b) convergence rate for $p$.

Figure 2.9. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the non-conservative manufactured solution with specified normal velocity and tangential traction (vel-tra) boundary conditions: (a) convergence rate for $u$; (b) convergence rate for $p$. 
Figure 2.10. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the non-conservative manufactured solution with specified normal traction and tangential velocity (tra-vel) boundary conditions: (a) convergence rate for $u$; (b) convergence rate for $p$.

Figure 2.11. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the non-conservative manufactured solution with specified normal and tangential traction (tra-tra) boundary conditions: (a) convergence rate for $u$; (b) convergence rate for $p$. 
2.6.2. Non-conservative form: Effect of density and viscosity ratios

Next, we verify the accuracy of the solver/discretization for a wide range of density and viscosity ratios. A manufactured solution for the non-conservative form is taken to be

\begin{align}
    u(x, t) &= -\cos(2\pi t - 2\pi x) \sin(2\pi t - 2\pi y), \\
    v(x, t) &= \sin(2\pi t - 2\pi x) \cos(2\pi t - 2\pi y) + \cos(t - 2\pi x), \\
    p(x, t) &= \sin(2\pi t - 2\pi x) \sin(2\pi t - 2\pi y),
\end{align}

with time-independent density and viscosity fields,

\begin{align}
    \rho(x) &= \rho_0 \left( \frac{R_\rho - 1}{2} \tanh \left( \frac{0.1 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{\delta} \right) + \frac{R_\rho + 1}{2} \right), \\
    \mu(x) &= \mu_0 \left( \frac{R_\mu - 1}{2} \tanh \left( \frac{0.1 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{\delta} \right) + \frac{R_\mu + 1}{2} \right).
\end{align}

Physically, this describes a “bubble” centered in the computational domain \( \Omega = [0, L]^2 = [0, 1]^2 \). The density (viscosity) in the domain varies smoothly between \( \rho_0 \) (\( \mu_0 \)) outside of the domain to \( R_\rho \cdot \rho_0 \) (\( R_\mu \cdot \mu_0 \)), indicating a density (viscosity) ratio of \( R_\rho \) (\( R_\mu \)). The smoothing parameter is set to \( \delta = 0.03L \). The density, viscosity, and initial velocity fields are shown in Fig. 2.12.

The effect of varying density ratio \( R_\rho \) is considered first. The outer viscosity is set to \( \mu_0 = 10^{-4} \) with viscosity ratio \( R_\mu = 10^1 \), while the outer density is set to \( \rho_0 = 10^0 \). Density ratios of \( R_\rho = 10^1, 10^2, 10^3, 10^4, 10^5, 10^6 \) are considered. For all physical boundaries, specified normal velocity and tangential traction boundary conditions are used. The maximum velocities in the domain for this manufactured solution are \( \mathcal{O}(1) \),
hence a relevant time scale is $L/U$ with $U = 1$. Errors in the velocity and pressure are computed at time $T = tu/L = 0.1$ using a uniform time step $\Delta t = 1/(6.25N)$, which yields an approximate CFL number of 0.3.

Fig. 2.13 shows the $L^1$ and $L^\infty$ errors for velocity and pressure as a function of grid size. Second-order convergence rates are achieved for velocity and pressure in both norms. It is clear that for a given grid size, the error increases as a function of density ratio. Additionally, this increase in error is more pronounced in the pressure than in the velocity. This is unsurprising: high density ratio simulations need to be adequately refined in order to accurately resolve the pressure jump across a dense region. We remark that for many air-water interface impact problems that are prevalent in ocean and marine engineering, pressure forces dominate viscous traction. Therefore, high accuracy in the pressure is desirable for this class of applications.
Next, we consider the effect of varying viscosity ratio. The outer density is set to \( \rho_0 = 10^0 \) with density ratio \( R_\rho = 10^1 \), while the outer viscosity is set to \( \mu_0 = 10^{-4} \).
Viscosity ratios of $R_\mu = 10^1, 10^2, 10^3, 10^4, 10^5, 10^6$ are considered. The same grid sizes and time step from the varying density case are used.

Fig. 2.14 shows the $L^1$ and $L^\infty$ errors for velocity and pressure as a function of grid size. We again observe second-order convergence rates for both velocity and pressure. At a given grid size, the errors increase as a function of viscosity ratio, although the growth is not as significant as in the varying density case. These tests show that the present numerical discretization maintains the desired order of accuracy for a wide range of fluid properties.

2.6.3. Conservative form: Effect of density evolution and synchronization

In this next case, we verify the effect of consistent mass density update and conservative discretization on the order of accuracy of the computed solution. For this test case we take the velocity and pressure solutions to be

(2.102) \hspace{1cm} u(x, t) = -y \cos(t),

(2.103) \hspace{1cm} v(x, t) = x \cos(t),

(2.104) \hspace{1cm} p(x, t) = \sin(t) \sin(x) \sin(y),
and we prescribe

\begin{align}
(2.105) \\
\rho(x, t) &= \rho_0 + \frac{\rho_1}{2} \left( \tanh \left( \frac{0.1 - \sqrt{x^2 + y^2}}{\delta} \right) + 1 \right) + x \cos(\sin(t)) + y \sin(\sin(t)) + 2, \\
(2.106) \\
\mu(x) &= \mu_0 + \mu_1 + \mu_1 \sin(2\pi x) \cos(2\pi y).
\end{align}

Figure 2.14. Errors as a function of grid size $N$ for varying viscosity ratios $\mu_R = 10^1 - 10^6$ with $\mu_0 = 10^{-4}$ and constant density ratio $\rho_R = 10^1$ with $\rho_0 = 10^0$ for a non-conservative manufactured solution. Normal velocity and tangential traction boundary conditions are specified at all physical boundaries in every case. (a) $L^1$ convergence rate for $u$; (b) $L^{\infty}$ convergence rate for $u$; (c) $L^1$ convergence rate for $p$; (d) $L^{\infty}$ convergence rate for $p$. 
Figure 2.15. The (a) initial density and (b) viscosity fields, along with the initial velocity vectors for the manufactured solution described in Sec. 2.6.3.

Plugging Eqs. (2.92)–(2.106) into the conservative momentum equation (5.1) yields a forcing term $f(x, t)$ that produces the desired conservative manufactured solution. Moreover, it can be verified that the time-dependent density function in Eq. (2.105) satisfies the conservative mass balance equation (2.3). Also notice that density function in Eq. (2.105) is only $C^0$ continuous, and, in particular, that it has singular spatial derivatives at the origin. We remark that unlike the non-conservative form of equations, in which an arbitrary density function can be selected for the manufactured solution, constructing manufactured solutions for the conservative form is non-trivial because the velocity and density fields must satisfy both the mass balance and divergence-free condition. The viscosity and initial velocity and density fields are shown in Fig. 2.15.

For this case, the variations in density are set to $\rho_0 = 1$ and $\rho_1 = 10^3 - \rho_0$, and the variations in viscosity are set to $\mu_0 = 10^{-2}$ and $\mu_1 = 1 - \mu_0$. The computational domain is $\Omega = [-L, L]^2 = [-1, 1]^2$. The smoothing parameter is set to $\delta = 0.05L$. Normal
velocity and tangential traction boundary conditions are used at all physical boundaries.

The maximum velocities in the domain for this manufactured solution are $O(1)$, hence a relevant time scale is $L/U$ with $U = 1$. Errors in the velocity and pressure are computed at time $T = tU/L = 0.6$ with a uniform time step size $\Delta t = 1/(1.042N)$, which yields an approximate CFL number of 0.5.

Figure 2.16. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the conservative manufactured solution with specified normal velocity and tangential traction boundary conditions on all boundaries. In these cases, the density $\rho$ is not reset between time steps. (a) convergence rate for $u$; (b) convergence rate for $p$; (c) convergence rate for $\rho$. 
As a first test, we consider the evolution of density along with velocity and pressure. The time-independent viscosity is set at the initial time and is not evolved during the simulation. Fig. 2.16 shows the $L^1$ and $L^\infty$ errors for velocity, pressure, and density as a function of grid size. Second-order convergence rates are achieved for velocity and pressure in both norms. The convergence rate for the $L^1$ density error is at least second order. Local reductions in pointwise convergence rates (less than second-order) are seen for density. This can be attributed to the $C^0$ spatial continuity of $\rho(x, t)$. We do indeed obtain full second-order convergence rates for all norms when considering a smooth density manufactured solution (see Sec. 2.6.6).

In our next test, we consider the same conservative manufactured solution as above, but we instead reset the density field at the beginning of each time step by computing the face-centered $\rho^n$ directly from Eq. (2.105) at time $t^n$. Hence, within each time step, $\rho^n$ is numerically evolved to $\hat{\rho}^{n+1,k+1}$ using the SSP-RK3 update described in Sec. 2.3.5.2. To wit, the density evolution reads as

\begin{align}
\hat{\rho}^{(1)} &= \rho^n - \Delta t \mathbf{R} \left( \mathbf{u}_{\text{adv}}^n, \rho_{\text{lim}}^n \right), \\
\hat{\rho}^{(2)} &= \frac{3}{4} \rho^n + \frac{1}{4} \hat{\rho}^{(1)} - \frac{1}{4} \Delta t \mathbf{R} \left( \mathbf{u}_{\text{adv}}^{(1)}, \hat{\rho}_{\text{lim}}^{(1)} \right), \\
\hat{\rho}^{n+1,k+1} &= \frac{1}{3} \rho^n + \frac{2}{3} \hat{\rho}^{(2)} - \frac{2}{3} \Delta t \mathbf{R} \left( \mathbf{u}_{\text{adv}}^{(2)}, \hat{\rho}_{\text{lim}}^{(2)} \right).
\end{align}

This evolved quantity is only used in the conservative discretization of the momentum equation (2.48). Upon numerically solving for the velocity and pressure, this density approximation is discarded and we begin the next time step with $\rho^{n+1}$ computed from Eq. (2.105) at time $t^{n+1}$. On the final time step, the density error norms are computed
with respect to the evolved $\hat{\rho}$ in order to determine the order of accuracy of a single time step of SSP-RK3 integration. This density resetting procedure emulates the level set synchronization approach described in Sec. 2.3.5.2 which is used in all of the numerical examples presented in Sec. 2.7. This methodology is commonly used in other interface capturing models as well, such as volume of fluid [20] and phase field [76] approaches, in which the material properties are set via an auxiliary indicator function.

Fig. 2.17 shows the $L^1$ and $L^\infty$ errors for velocity, pressure, and density as a function of grid size. We again observe full second-order convergence rates for velocity and pressure. We also no longer see a reduction in the order of accuracy for density, even for the $C^0$ density field; this is because the error is essentially computed after a single time step of density evolution. These tests therefore show that the present numerical discretization maintains the desired order of accuracy for a conservative formulation with consistent mass update.

2.6.4. Non-conservative/Conservative form: Effect of local mesh refinement

We now consider the non-conservative and conservative manufactured solutions of the previous sections, 2.6.1 and 2.6.3, on locally refined grids and analyze the error convergence rate. In all of the cases considered here, we use $\ell = 2$ mesh levels with a refinement ratio of $n_{\text{ref}} = 4$. The refined region is always assigned to the center of the domain, which contains the dense region, as shown in Fig. 2.18. It should be noted that for practical multiphase simulations, we always restrict the fluid-gas interface to the finest mesh level in the grid hierarchy.
Figure 2.17. $L^1$ (■, blue) and $L^\infty$ (○, red) errors as a function of grid size $N$ for the conservative manufactured solution with specified normal velocity and tangential traction boundary conditions on all boundaries. In these cases, the density $\rho$ is reset between time steps. (a) convergence rate for $u$; (b) convergence rate for $p$; (c) convergence rate for $\rho$.

In our first test, we consider the non-conservative manufactured solution detailed in Sec. 2.6.1. We consider vel-tra boundary conditions, i.e., periodic boundary conditions in the $x$-direction and specified normal velocity and tangential traction in the $y$-direction. Fig. 2.19 shows the $L^1$ and $L^\infty$ errors for velocity and pressure as a function of coarsest
Figure 2.18. Velocity vectors and pressure field plotted on the grid hierarchy used for the locally refined mesh convergence study. In both figures, the purple box indicates a locally refined grid with grid spacing $\Delta x_{\text{min}} = \Delta x_0/4$.

Figure 2.19. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of coarsest grid spacing $\Delta x_0$ for the non-conservative manufactured solution described in Sec. 2.6.1 on a locally refined mesh. The specified boundary conditions are normal velocity and tangential traction (vel-tra). (a) convergence rate for $u$; (b) convergence rate for $p$.

grid spacing. We again obtain second-order convergence rates for velocity and pressure on the locally refined grid.
Next, we consider the conservative manufactured solution described in Sec. 2.6.3. In this case, we do not reset the density at the beginning of time steps. Fig. 2.20 shows the $L^1$ and $L^\infty$ errors for velocity, pressure, and density as a function of coarsest grid spacing. The observed convergence rates observed on this locally refined mesh are nearly identical to those shown for the uniform mesh case.

### 2.6.5. Conservative form: Solver scalability

Finally, we investigate the scalability of the preconditioned FGMRES solver for the staggered Stokes system given by Eq. (2.70) and also show the linear solver iteration results for the velocity and pressure subdomain problems. A bubble is placed in a computational domain $\Omega = [0, L]^2$ with $L = 1$, which is discretized by an $N \times N$ grid. The radius of the bubble is $R = 0.2L$ with initial center position $(X_0, Y_0) = (L/2, L/2)$. The bubble has density $\rho_i$ and viscosity $\mu_i$ and is placed within ambient fluid of density $\rho_o$ and $\mu_o$. The density and viscosity jump is smeared over $n_{\text{cells}} = 2.5$ grid cells on each side of the interface and harmonic averaging is used to average viscosity from cell-centers to nodes of the staggered grid. The initial velocity and pressure are set to

\begin{align*}
(u(x, 0) &= \cos(2\pi x) \sin(2\pi y), \\
(v(x, 0) &= -\sin(2\pi x) \cos(2\pi y), \\
p(x, 0) &= 0.
\end{align*}

Homogenous normal velocity and homogeneous tangential traction boundary conditions are used at all sides of the domain, and homogenous Neumann boundary conditions for
\( L^1 \) (■, blue) and \( L^\infty \) (○, red) errors as a function of coarsest grid spacing \( \Delta x_0 \) for the conservative manufactured solution with specified normal velocity and tangential traction boundary conditions on all boundaries. In these cases, the density \( \rho \) is not reset between time steps. \( \square \) convergence rate for \( u \); \( \square \) convergence rate for \( p \); \( \square \) convergence rate for \( \rho \).

\( \phi, \rho, \) and \( \mu \) are specified on \( \partial \Omega \). No additional body forces are applied to the momentum equations (i.e., \( f(x, t) = 0 \)). The maximum velocities in the domain for this manufactured solution are \( \mathcal{O}(1) \), hence a relevant time scale is \( L/U \) with \( U = 1 \). Each case is run until \( T = tU/L = 0.1 \) with a uniform time step \( \Delta t = 1/(3.125N) \), which yields an approximate
CFL number of 0.3. Residual values are presented at the final time from the second fixed-point cycle.

In the first case, the bubble is set to have $\rho_i = 10^3$ and $\mu_i = 8.9 \times 10^{-4}$, and the exterior fluid is set to have $\rho_o = 1.225$ and $\mu_o = 1.81 \times 10^{-5}$. This corresponds to the material properties of water and air, respectively, in meter-kilogram-second (MKS) units. The initial problem set up is shown in Fig. 2.21. The Reynolds number based on the properties of the exterior fluid, the diameter of the bubble, and the maximum initial velocity in the domain is $Re = 2.71 \times 10^4$. The subdomain solvers with loose relative convergence tolerance of $\epsilon_{sub} = 10^{-2}$ are used for this problem. For the outer FGMRES solver, a tight relative convergence tolerance of $\epsilon_{Stokes} = 10^{-16}$ is employed. Fig. 2.22 summarizes iteration results for the three linear solvers considered. We only show residual data for the velocity and pressure solves from the final FGMRES iteration of each simulation, although we note that each outer iteration requires essentially the same number of inner solver iterations. Hence, the number of inner solver V-cycles per Stokes cycle can be approximated as the number of subdomain solver iterations multiplied by the number of FGMRES iterations shown in Fig. 2.22(e). For the velocity subdomain solver at all grid sizes, we see that it only takes a single iteration to reduce the residual by several orders of magnitude, well below the desired tolerance. This is expected for this particular case because the system is strongly diagonally dominant. For the pressure subdomain solver, we see that it takes between three and six iterations to reduce the relative residual below $\epsilon_{sub}$. Although the required number of iterations increases as a function of grid size, it does so at only a modest rate. For the full staggered Stokes system, the FGMRES solver converges to machine precision within five iterations for all
Figure 2.21. The initial location of the air-suspended water droplet, and the initial velocity vectors for the problem set up described in Sec. 2.6.5.

the grid sizes $N$. To converge the Stokes system for the wide range of grid sizes shown here to a relative residual tolerance of $\epsilon_{\text{Stokes}} = 10^{-12}$, we would expect to carry out around 3–4 velocity V-cycles and 12–18 pressure V-cycles. To obtain accurate results for practical multiphase fluid flow problems, it is reasonable to converge the Stokes system to a relative residual of around $10^{-6}$, requiring only a single FGMRES iteration, a single V-cycle for the velocity and around 3–6 V-cycles for the pressure.

In the second case, the bubble is set to have $\rho_i = 1$ and $\mu_i = 8.9 \times 10^{-3}$, whereas the outside fluid is set to have $\rho_o = 1.225 \times 10^{-3}$ and $\mu_o = 1.81 \times 10^{-4}$. This corresponds to the material properties of water and air, respectively, in centimeter-gram-second (CGS) units. The Reynolds number based on the properties of the exterior fluid, the diameter of the bubble, and the maximum initial velocity in the domain is $\text{Re} = 2.71$. Fig. 2.23 summarizes iteration results for the three linear solvers considered. In contrast with the higher Reynolds number case, it generally takes between two and three iterations for the velocity subdomain solver to converge below the desired tolerance. Again, this is
Figure 2.22. Relative residual as a function of iteration count for the final solves of the water bubble problem described in Sec. 2.6.5, for varying grid sizes $N = 32 - 1024$. MKS units are used to specify the material properties of both water and air, yielding a Reynolds number of $\text{Re} = 2.71 \times 10^4$. Expected, because at this Reynolds number, $\mathbf{A}$ is not as diagonally dominant. Similarly, the pressure subdomain solver requires between two and three iterations for all grid sizes. The full staggered Stokes system requires between 14 and 19 iterations to converge to machine precision. Hence, to converge the Stokes system to a relative residual tolerance
Figure 2.23. Relative residual as a function of iteration count for the final solves of the water bubble problem described in Sec. 2.6.5, for varying grid sizes $N = 32 - 1024$. CGS units are used to specify the material properties of both water and air, yielding a Reynolds number of $Re = 2.71 \times 10^9$.  

(a) Velocity subdomain solver relative residual $\|A\hat{x}_u - b_u\|_2 / \|b_u\|_2$ vs. number of iterations.  
(b) Pressure subdomain solver relative residual $\|\mathbf{L}_D\theta + \frac{1}{\Delta t} (b_p + D \cdot \hat{x}_u)\|_2 / \|\mathbf{L}_D(b_p + D \cdot \hat{x}_u)\|_2$ vs. number of iterations.  
(c) Staggered Stokes solver relative residual $\|Bx - b\|_2 / \|b\|_2$ vs. number of iterations.

of $\epsilon_{\text{Stokes}} = 10^{-12}$, we would expect to carry out around 20–39 velocity $V$-cycles and a comparable number of pressure $V$-cycles. To achieve a relative residual tolerance of $10^{-6}$, we would expect to carry out around 6–9 $V$-cycles for both the velocity and pressure.
The results here suggest that the preconditioned iterative solver is scalable for high density and high viscosity ratio flows, including practical air-water interfacial flows at both low and high Reynolds numbers. The number of iterations required to converge the outer and inner solvers are relatively insensitive to grid size. Similar convergence behaviors are observed for other combinations of physical boundary conditions (results not shown). In the following sections, we mainly focus on multiphase flows involving air-water interaction, for which the solver parameters discussed here are reasonably efficient.

### 2.6.6. Conservative form: Smooth density profile evolution

In Sec. 2.6.3, we note that the $C^0$ spatial continuity of the manufactured solution density field can lead to reductions in the pointwise convergence rate. This section demonstrates that using the smooth density field,

$$(2.113) \quad \rho(x, t) = 2 + x \cos(\sin(t)) + y \sin(\sin(t)),$$

yields second-order pointwise convergence rates for $u$, $p$, and $\rho$. The same velocity (Eqs. (2.102)–(2.103)), pressure (Eq. (2.104)), and viscosity (Eq. (2.106)) fields are used to produce the desired forcing term. Note that this density field also satisfies the conservative mass balance equation (2.3). For all physical boundaries, specified normal velocity and tangential velocity boundary conditions are used. The domain $\Omega = [-L, L] = [-1, 1]^2$ is discretized by an $N \times N$ grid. The maximum velocities in the domain for this manufactured solution are $O(1)$, hence a relevant time scale is $L/U$ with $U = 1$. Errors in the velocity and pressure are computed at time $T = tU/L = 0.6$ with a uniform time step
size $\Delta t = 1/(1.042N)$, which yields an approximate CFL number of 0.5. A relative convergence tolerance of $\epsilon_{\text{Stokes}} = 10^{-12}$ is specified for the FGMRES solver. For these cases, density is evolved and not reset between time steps. Fig. 2.24 shows the $L^1$ and $L^\infty$ errors.
for velocity, pressure, and density as a function of grid size. Second-order convergence rates are achieved for velocity, pressure, and density in both norms.

2.6.7. First-order density update scheme

Here, we demonstrate the importance of the SSP-RK3 and CUI limited density update for maintaining second-order convergence rates. Using the same manufactured solution of Sec. 2.6.6, we instead use forward Euler timestepping for the density update and an first-order upwind scheme to approximate the shifted control volume density $\rho_{\text{lim}}$. We still use CUI to approximate the shifted control volume velocity $u_{\text{lim}}$ for the convective derivative. A similar scheme was considered in [39]. Fig. 2.25 shows that first order convergence rates are obtained not only for the density field, but also for pressure and velocity. Therefore, the additional complexity of the primary discretization scheme is justified.

2.6.8. Three-dimensional manufactured solution

To complete our discussion, we consider the following manufactured solution in three spatial dimensions for the non-conservative set of equations,

\begin{align}
  u(x, t) &= 2\pi \cos(2\pi x) \sin(2\pi z) \cos(2\pi t - 2\pi y), \\
  v(x, t) &= -2\pi \sin(2\pi x) \cos(2\pi z) \sin(2\pi t - 2\pi y), \\
  w(x, t) &= \pi \sin(2\pi x) \cos(2\pi t - 2\pi y)(-2\sin(2\pi z) - 2\cos(2\pi z)), \\
  p(x, t) &= 2\pi \sin(2\pi t - 2\pi x) \cos(2\pi t - 2\pi y) \sin(2\pi t - 2\pi z),
\end{align}
Figure 2.25. $L^1$ (■, blue) and $L^\infty$ (●, red) errors as a function of grid size $N$ for the conservative manufactured solution with specified normal velocity and normal velocity boundary conditions on all boundaries, and smooth density profile given by Eq. (2.113). In these cases, the density $\rho$ is not reset between time steps. Forward Euler timestepping is used to update $\rho$, while an upwind scheme is used to approximate $\rho_{\text{lim}}$ on the faces of the shifted control volumes. \(\text{(a)}\) convergence rate for $u$; \(\text{(b)}\) convergence rate for $p$; \(\text{(c)}\) convergence rate for $\rho$.

Together with time-independent density and viscosity fields of the form

\begin{equation}
\rho(\mathbf{x}) = \rho_0 + \frac{\rho_1}{2} \left( \tanh \left( \frac{0.1 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2 + (z - 0.5)^2}}{\delta} \right) + 1 \right),
\end{equation}

\begin{equation}
\mu(\mathbf{x}) = \mu_0 + \mu_1 + \mu_1 \sin(2\pi x) \cos(2\pi y) \sin(2\pi z).
\end{equation}
The variations in density and viscosity are set to be similar to that of air and water: 
\[ \rho_0 = 1, \quad \rho_1 = -\rho_0 + 10^3, \quad \mu_0 = 10^{-4}, \quad \text{and} \quad \mu_1 = -\mu_0 + 10^{-2}. \]
The computational domain is the unit square \( \Omega = [0, L]^2 = [0, 1]^2 \), which is discretized by \( N \) grid cells in each direction. The smoothing parameter is set to \( \delta = 0.05L \). The maximum velocities in the domain for this manufactured solution are \( \mathcal{O}(1) \), hence a relevant time scale is \( L/U \) with \( U = 1 \). Errors in the velocity and pressure are computed at time \( T = tU/L = 0.1 \) with a uniform time step \( \Delta t = 1/(31.25N) \), which yields an approximate CFL number of 0.3. A relative convergence tolerance of \( \epsilon_{\text{Stokes}} = 10^{-12} \) is specified for the FGMRES solver. We impose periodic boundary conditions in the \( x \)- and \( z \)-direction and consider normal velocity and tangential traction boundary conditions in the \( y \)-direction (vel-tra).

Fig. 2.26 shows the \( L^1 \) and \( L^\infty \) errors for velocity and pressure as a function of grid size. Second-order convergence rates are achieved for both velocity and pressure in both norms, which demonstrates that the solver is maintaining the expected order of accuracy in three spatial dimensions.

### 2.7. Numerical examples

This section investigates several multiphase flow problems to verify the accuracy, consistency, and stability of the present numerical method. Both conservative and non-conservative formulations are used, and differences between the formulations are examined. We also compare our results to benchmark problems drawn from the multiphase flow literature. We employ \( n_{\text{cycles}} = 2 \) cycles of fixed-point iteration for all the examples in this section. We note that errors lower than machine precision may be truncated during computation, and we only report errors lower than \( 10^{-16} \) as \( \leq 10^{-16} \).
2.7.1. Static bubble with surface tension

We first demonstrate that our treatment of surface tension is well-balanced with respect to the treatment of the pressure gradient in the hydrostatic limit. A bubble is placed in a computational domain $\Omega = [0,L]^2$ with $L = 1$, which is discretized by a uniform $N \times N$ grid. The radius of the bubble is $R = 0.25L$ with initial center position $(X_0, Y_0) = (L/2, L/2)$. Both the fluids inside and outside the bubble are initially at rest. Zero normal and tangential velocity boundary conditions are imposed on the boundaries of the computational domain. The only momentum body force considered for this case is the surface tension force; other body forces such as gravity are neglected. The conservative discretization is used for all cases considered here, but the non-conservative discretization yields similar results because these flows are not dominated by convection (data not available)}
shown). A relative convergence tolerance of $\epsilon_{\text{Stokes}} = 10^{-15}$ is specified for the FGMRES solver, which can corresponds to absolute tolerances in the range of $10^{-10} - 10^{-15}$. Two grid cells ($n_{\text{cells}} = 2$) of smearing are used on either side of the interface.

As a first test case, we consider an inviscid flow with inner and outer viscosities $\mu_i = \mu_o = 0$. The grid size is taken to be $N = 80$, and a constant time step size $\Delta t = 1/(12.5N)$ is used. Density ratios of $\rho_i/\rho_o = 1, 10^3, \text{and } 10^6$ are considered, and each case is run for 50 time steps. For this particular case, the surface tension should exactly balance the pressure difference across the interface, resulting in a stationary bubble with zero velocity everywhere in the domain. The exact pressure difference is analytically given by the Young-Laplace equation

\begin{equation}
\Delta p_{\text{exact}} = \frac{\sigma \kappa}{R},
\end{equation}

in which the surface tension coefficient is set to $\sigma = 73$, and the exact curvature of the bubble is $\kappa = 1/R$. This case has been extensively investigated by a number of previous studies, including Williams et al. 54, Francois et al. 70, and Patel and Natarajan 44. If an unbalanced treatment of surface tension and pressure gradient is used, significant parasitic currents will be generated during the first time step, and the exact pressure jump will not be captured. As in previous studies, we use the exact curvature for the surface tension force calculation. Because the fluid should remain at rest, any nonzero velocities are the result of solver error. We assess the accuracy of the solver in terms of the $L^1$ and $L^\infty$ norms of the velocity along with the relative error in the pressure jump

\begin{equation}
E(\Delta p) = \frac{|\Delta p - \Delta p_{\text{exact}}|}{|\Delta p_{\text{exact}}|},
\end{equation}
Table 2.1. Errors in velocity and pressure after a single time step for the two-dimensional static and inviscid bubble. The exact curvature $\kappa = 1/R$ is used for the surface tension force calculation.

<table>
<thead>
<tr>
<th>$\rho_i/\rho_o$</th>
<th>$|u|_1$</th>
<th>$|u|_\infty$</th>
<th>$E(\Delta p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$3.76 \times 10^{-14}$</td>
<td>$1.11 \times 10^{-13}$</td>
<td>$9.95 \times 10^{-14}$</td>
</tr>
<tr>
<td>$10^3$</td>
<td>$2.03 \times 10^{-15}$</td>
<td>$1.20 \times 10^{-13}$</td>
<td>$\leq 10^{-16}$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>$8.44 \times 10^{-13}$</td>
<td>$4.73 \times 10^{-11}$</td>
<td>$5.84 \times 10^{-16}$</td>
</tr>
</tbody>
</table>

Table 2.2. Errors in velocity and pressure after a 50 time steps for the two-dimensional static and inviscid bubble. Exact curvature $\kappa = 1/R$ is used for the surface tension force calculation.

<table>
<thead>
<tr>
<th>$\rho_i/\rho_o$</th>
<th>$|u|_1$</th>
<th>$|u|_\infty$</th>
<th>$E(\Delta p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$4.17 \times 10^{-16}$</td>
<td>$9.36 \times 10^{-15}$</td>
<td>$5.84 \times 10^{-16}$</td>
</tr>
<tr>
<td>$10^3$</td>
<td>$1.95 \times 10^{-14}$</td>
<td>$5.16 \times 10^{-13}$</td>
<td>$\leq 10^{-16}$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>$3.61 \times 10^{-12}$</td>
<td>$1.81 \times 10^{-10}$</td>
<td>$\leq 10^{-16}$</td>
</tr>
</tbody>
</table>

in which $\Delta p = p_i - p_o$ is the numerically computed pressure difference across the interface and $\Delta p_{\text{exact}} = 73/0.25 = 292$. Tables 2.1 and 2.2 show the errors in velocity and pressure at $t = \Delta t$ and $t = 50\Delta t$, respectively. For all density ratios, we see that the errors are close to machine precision, indicating negligible spurious currents and a numerical balance between pressure and surface tension forces.

Next, we consider the temporal evolution of velocity and pressure errors for a viscous droplet suspended in fluid. We consider the same geometric parameters as the previous example for the bubble and the domain size. The density ratio is held fixed at $\rho_i/\rho_o = 10^3$, and we consider grids with $N = 40, 80, 160$, and 320. A constant time step size of $\Delta t = 1/(50N)$ is used, and the surface tension coefficient is again set to $\sigma = 73$. The viscosity in each fluid is determined using the dimensionless Laplace number $La = \rho D \sigma / \mu^2 = 12000$, \ldots
Figure 2.27. Temporal evolution of errors for a viscous bubble with $\rho_i/\rho_o = 10^3$ and $La = 12000$, for grid sizes $N = 40\text{–}320$. Here, curvature is computed numerically from the level set function. (a) Pressure difference relative error $E(\Delta p)$ vs. nondimensional time. (b) $L^1$ norm of velocity vs. nondimensional time. (c) $L^\infty$ norm of velocity vs. nondimensional time.

in which $D = 2R$ is the diameter of the bubble. Using the parameters described here yields a viscosity ratio of $\mu_i/\mu_o = 31.62$, and both fluids are initially at rest. In contrast with the previous example, here the curvature is computed directly from the level set function. Time is nondimensionalized using the capillary time scale $t_\sigma = \sqrt{\rho_o D^3/\sigma}$. Similar cases have been numerically investigated by Popinet [77] and by Abadie et al. [78].
Fig. 2.27 shows the time evolution of $E(\Delta p)$, $\|u\|_1$, and $\|u\|_\infty$. We observe that the errors do not grow substantially as a function of time, and that each computation maintains stability. These results are consistent with the results of other level set based multiphase flow solvers. In particular, it is known that the reinitialization procedure will slightly shift the interface every time step, which can prevent the spurious velocities from decaying to machine precision [78]. In contrast, certain geometric VOF methods exhibit exponential decay of the spurious velocities because of the absence of any redistancing process [77]. Nonetheless, the errors shown in shown in Fig. 2.27 decrease as the resolution increases, yielding convergence rates of 1.98 for $E(\Delta p)$, 1.94 for $\|u\|_1$, and 1.75 for $\|u\|_\infty$ between the two finest cases $N = 160$ and $N = 320$. Fig. 2.28 shows the velocity vectors and pressure for the finest case $N = 320$ at $t/t_\sigma = 60.41$. Although we do see some spurious velocities across the computational domain, the large internal pressure required to maintain the bubble’s shape is accurately resolved. These results indicate that the surface tension treatment described here is well-balanced and does not significantly reduce the expected order of accuracy of the fluid solver.

### 2.7.2. Gas/liquid tank

This section demonstrates that our treatment of the gravitational body force is well-balanced with respect to the treatment of the pressure gradient. The computational domain $\Omega = [0, L]^2$ with $L = 1$ is filled halfway from $y = 0$ to $y = L/2$ with a heavy fluid of density $\rho_h$. The remainder of the tank, from $y = L/2$ to $y = L$, is occupied by a lighter fluid with density $\rho_l = 1$, and the density ratio $\rho_h/\rho_l$ is varied. The viscosity is set to zero for both fluids, $\mu_l = \mu_h = 0$. The domain is discretized by a uniform $N \times N$ grid, and both
Figure 2.28. Velocity vectors and pressure of the static bubble with $\rho_i/\rho_o = 10^3$ and $La = 12000$, for grid size $N = 320$ at time $t/t_\sigma = 60.41$. Curvature is computed numerically from the level set function.

the fluids are initially at rest. A gravitational acceleration $\mathbf{g} = (0, -g_y) = (0, -9.81)$ is specified, and surface tension is neglected. The conservative discretization is used for all cases considered here, but we found that the non-conservative discretization yields similar results because these flows are not dominated by convection (data not shown). A relative convergence tolerance of $\epsilon_{\text{Stokes}} = 10^{-15}$ is specified for the FGMRES solver, and two grid cells of smearing ($n_{\text{cells}} = 2$) are used on either side of the interface.

As a first test case, we prescribe homogenous normal velocity conditions on the left, bottom, and right boundaries and homogenous normal traction conditions on the top boundary along with homogenous tangential velocity conditions on all boundaries of the computational domain. The grid is held fixed with size $N = 100$ and a constant time step size $\Delta t = 1/(10N)$ is used. Density ratios of $\rho_h/\rho_l = 10^1, 10^3, \text{and} 10^6$ are considered and each case is run for 50 time steps. For this particular case, the gravitational force should exactly balance the pressure difference across the interface, resulting in a stationary tank of
fluid with zero velocity everywhere in the domain. Similar cases has been computationally investigated by Montazeri et al. [79] and by Patel and Natarajan [44]. Because the pressure is defined on cell-centers of the staggered grid, the analytical pressure jump for this problem can be computed by integrating the momentum equation (ignoring velocity terms):

$$
\int_{\Delta y/2}^{L-\Delta y/2} \nabla p(y) \, dy = \int_{\Delta y/2}^{L-\Delta y/2} \rho(y)g \, dy = -g_y \left( \int_{L/2}^{L-\Delta y/2} \rho_1 \, dy + \int_{\Delta y/2}^{L/2} \rho_h \, dy \right)
$$

(2.122)

$$
\Rightarrow \Delta p_{\text{exact}} = p \left|_{\Delta y/2}^{L-\Delta y/2} \right. = -\frac{1}{2} g_y (\rho_h + \rho_1) (L - \Delta y),
$$

in which the pressure is evaluated at the cell centers adjacent to the top and bottom computational boundaries and $\Delta y = L/N$ is the vertical grid spacing for this particular problem. If an unbalanced treatment of gravity and pressure gradient is used, significant parasitic currents will be generated, and the exact pressure jump will not be captured. Similar to the static bubble case in the previous section, two quantitative measurements are used to assess the accuracy of the simulation: the $L^\infty$ norm in velocity, and the relative error in pressure jump $E(\Delta p)$ using the numerically computed pressure difference between $y = \Delta y/2$ and $y = L - \Delta y/2$, and the exact pressure difference $\Delta p_{\text{exact}} = -9.81/2 \times (\rho_h + \rho_1) \times (1 - 0.01)$.

Tables 2.3 and 2.4 show the errors in velocity and pressure at $t = \Delta t$ and $t = 50\Delta t$, respectively. For all density ratios considered, we see that the errors are close to machine precision, indicating negligible spurious currents and a numerical balance between pressure and gravitational body force.
Table 2.3. Errors in velocity and pressure after a single time step for the two-dimensional static and inviscid fluid column.

<table>
<thead>
<tr>
<th>$\rho_h/\rho_l$</th>
<th>$|\mathbf{u}|_\infty$</th>
<th>$E(\Delta p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>$2.53 \times 10^{-16}$</td>
<td>$5.45 \times 10^{-15}$</td>
</tr>
<tr>
<td>$10^3$</td>
<td>$7.21 \times 10^{-15}$</td>
<td>$9.32 \times 10^{-14}$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>$4.40 \times 10^{-12}$</td>
<td>$5.42 \times 10^{-12}$</td>
</tr>
</tbody>
</table>

Table 2.4. Errors in velocity and pressure after a 50 time steps for the two-dimensional static and inviscid fluid column.

<table>
<thead>
<tr>
<th>$\rho_h/\rho_l$</th>
<th>$|\mathbf{u}|_\infty$</th>
<th>$E(\Delta p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^4$</td>
<td>$\leq 10^{-16}$</td>
<td>$1.33 \times 10^{-16}$</td>
</tr>
<tr>
<td>$10^3$</td>
<td>$\leq 10^{-16}$</td>
<td>$\leq 10^{-16}$</td>
</tr>
<tr>
<td>$10^6$</td>
<td>$1.62 \times 10^{-16}$</td>
<td>$1.92 \times 10^{-16}$</td>
</tr>
</tbody>
</table>

Next, we consider the dynamic case of a tank being filled with fluid. The same numerical parameters from the previous case are used, except that the normal velocity at the bottom boundary is set to be unity. Over time, the fluid will displace the gas out of the domain until the tank consists only of the fluid phase. Only the case for density ratio of $\rho_h/\rho_l = 10^3$ is presented here, although the results are similar for a wide range of density ratios. Montazeri et al. demonstrated that the use of an unbalanced formulation will produce significant interface deformation resulting from the spurious currents, which eventually destabilizes the computation [79].

Fig. 2.29 shows three snapshots in time for the evolution of the filling water tank. We see the fluid-gas interface evolve upward with no deformation because negligible spurious horizontal velocities generated. These cases demonstrate that even though we are treating gravity as a momentum body force acting throughout the entire domain (and not as an...
interfacial force as done in by Montazeri et al. [79], the staggered-grid discretization provides a well-balanced pressure gradient and gravity forcing. The reason is attributed to the fact that the pressure gradient and the gravity forces are defined at the same spatial location, namely, the Cartesian grid faces. It was also recently recognized by Patel and Natarajan [44] that it is not necessary to express gravity force in terms of a gradient operator to obtain a well-balanced formulation.

2.7.3. Convection of high density droplet

In this section, we demonstrate the importance of consistent mass and momentum transport to achieve stability in high density ratio flows. A dense bubble is placed in a fully periodic computational domain $\Omega = [0, L]^2$ with $L = 1$, which is discretized by an $N \times N$ grid. The radius of the bubble is $R = 0.2L$ with initial center position $(X_0, Y_0) = (L/4, L/2)$. The density ratio between the bubble and the outer fluid is $\rho_i/\rho_o = 10^6$, and the viscosity is set to $\mu = 0$ in the entire domain. The initial velocity is set via a smoothed Heaviside
function defined on faces

\[
\tilde{H}_{i-\frac{1}{2},j}^f = \begin{cases} 
0, & \phi_{i-\frac{1}{2},j} < -\Delta x \\
\frac{1}{2} \left( 1 + \frac{1}{2\Delta x} \phi_{i-\frac{1}{2},j} + \frac{1}{\pi} \sin \left( \frac{\pi}{\Delta x} \phi_{i-\frac{1}{2},j} \right) \right), & |\phi_{i-\frac{1}{2},j}| \leq \Delta x \\
1, & \text{otherwise}
\end{cases}
\]

in which \( \phi(x,0) = \sqrt{(x - X_0)^2 + (y - Y_0)^2 - R} \) is the initial signed distance function away from the circular interface and the face centered level set is obtained by averaging \( \phi \) in the two adjacent cell centers \( \phi_{i-\frac{1}{2},j} = \frac{1}{2} (\phi_{i-1,j} + \phi_{i,j}) \). The initial horizontal velocity \( u(x,0) \) is set to be unity inside the bubble and zero outside the bubble, i.e. \( u_{i-\frac{1}{2},j} = 1 - \tilde{H}_{i-\frac{1}{2},j}^f \). The vertical velocity \( v \) is initialized to be identically zero.

Notice that this initial velocity profile is not discretely divergence-free. Because this is an inviscid case, numerical instabilities and errors that are generated during the first time step can persist for the duration of the simulation. Hence, we apply a density-weighted projection to construct a discretely mass-conserving initial velocity \( u_p(x,0) \). This can be achieved by numerically solving the Poisson problem

\[
\rho_0(x) \frac{1}{\rho(x,0)} \mathbf{G} \psi(x) = -\mathbf{D} \cdot \mathbf{u}(x,0),
\]

for \( \psi(x) \) and then adding its gradient to the initial velocity field,

\[
u_p(x,0) = u(x,0) - \frac{1}{\rho(x,0)} \mathbf{G} \psi(x).
\]

This problem has been studied numerically by Bussman et al. [36], Desjardins and Moureau [39], Ghods and Herrmann [40], and Patel and Natarajan [44]. For all of
the following cases, the density is set via the level set function at the beginning of each

time step with one grid cell of smearing ($n_{\text{cells}} = 1$) on either side of the interface. We find
that smearing over two or more cells leads to a more distorted interface for this particular
problem. This is expected because analytically the dense bubble mimics a thick solid
blob, but numerically, as the interface of the blob thickens, it acts more like a fluidic blob
whose interface tends to “flow” under the pressure drag. Each case is run until $t = 1$ with
a constant time step size of $\Delta t = 1/(31.25N)$. Under the prescribed conditions, the dense
rigid blob should maintain its circular shape and return exactly to its initial position,
analytically.

As a first test, we demonstrate the importance of consistent mass and momentum
transport for the stability of high density ratio flows. Fig. 2.30 shows the shape of the
bubble for four different grid spacings, $N = 32, 64, 128, 256$, when simulated with in-
consistent mass and momentum transport, i.e., by using a non-conservative momentum
integrator. There is clear distortion in the bubble at time $t = 1$ at the lower resolutions
(Figs. 2.30(a) and 2.30(b)), whereas the simulations quickly become unstable at higher
grid resolutions (Figs. 2.30(c) and 2.30(d)). Lowering the time step even further did not
resolve these stability problems. However the simulations are stable when using consistent
mass and momentum transport (Fig. 2.31), and the bubble’s shape converges towards the
analytical solution as the resolution increases. The bubble’s distortion at lower resolutions
can be attributed to discretization errors from advection and interface tracking. For these
cases, xsPPM7 is used to advect the level set, and CUI limiters are used for mass and
convective fluxes. Therefore, these cases also demonstrate that the level set and mass up-
dates can have dissimilar limiters and still maintain numerical stability for the consistent
Figure 2.30. Initial condition/exact solution (---, red) and numerical solution (—, black) for convection of a high density droplet with density ratio \( \rho_i/\rho_o = 10^6 \) and initial velocity \( \mathbf{u}_p(x, 0) \). Inconsistent transport of mass and momentum occurs while employing a non-conservative momentum integrator. PPM is used for advection of the level set and CUI limiter is used for the convective operator. (a) Grid size \( N = 32 \) at time \( t = 1 \) (b) Grid size \( N = 64 \) at time \( t = 1 \) (c) Grid size \( N = 128 \) at time \( t = 0.5 \); the simulation becomes unstable shortly after the shown time (d) Grid size \( N = 256 \) at time \( t = 0.5 \); the simulation becomes unstable shortly after the shown time.

and conservative momentum integrator. This is in contrast to the numerical scheme of Patel and Natarajan [44], which remains consistent only when same limiter is employed for the advection of algebraic VOF-scalar and momentum flux.
Figure 2.31. Initial condition/exact solution (---, red) and numerical solution (—, black) for convection of a high density droplet with density ratio $\rho_i/\rho_o = 10^6$ and initial velocity $u_p(x, 0)$. Consistent transport of mass and momentum is used for these cases. PPM is used for advection of the level set and CUI limiters are used for mass and convective fluxes. (a) Grid size $N = 32$ at time $t = 1$. (b) Grid size $N = 64$ at time $t = 1$. (c) Grid size $N = 128$ at time $t = 1$. (d) Grid size $N = 256$ at time $t = 1$.

We next demonstrate the importance of using a CBC satisfying limiter for the mass density update. We consider the same simulation parameters above (for the consistent transport case) at a grid resolution of $N = 256$ and compare the shape of the bubble at $t = 1$ for four different limiters: xsPPM7 [66] along with high resolution versions of CUI [41], M-Gamma [41], and FBICS [43]. Figs. 2.32(a)–2.32(c) show the shape of the
bubble for the CUI, M-Gamma, and FBICS limiters. All of these limiters maintain numerical stability and produce reasonable results when compared to the analytical solution. Results for the PPM limiter are not shown here because the simulation quickly becomes unstable within ten time steps. Note that of these four limiters, PPM is the only one that can produce undershoots and overshoots for advected quantities [80]. It is required that the density remain positive throughout the computation to obtain a physically accurate solution. Fig. 2.32(d) shows the minimum updated density $\rho^{n+1}$ within each time step in the domain over the first eight time steps. It is clear that the CBC satisfying limiters maintain physically realistic minimum (maximum) densities whereas PPM generates undershoots (overshoots) that eventually corrupt the simulation. Throughout the remainder of the simulation, the CBC satisfying limiters do not undershoot the minimum or overshoot the maximum initial density (data not shown for overshoots).

Finally, we demonstrate that the consistent transport scheme produces stable results in three spatial dimensions and with adaptive mesh refinement. Similar to the previous case, a dense bubble is placed in a fully periodic computational domain $\Omega = [0, L]^3$ with $L = 1$, which is discretized by a two-level ($\ell = 2$) locally refined grid with coarsest grid spacing $\Delta x_0 = \Delta y_0 = \Delta y_0 = 1/64$ and refinement ratio $n_{\text{ref}} = 2$. At the finest level, the grid spacing is $\Delta x_{\text{min}} = \Delta y_{\text{min}} = \Delta z_{\text{min}} = 1/128$. The radius of the bubble is $R = 0.2L$ with initial center position $(X_0, Y_0, Z_0) = (L/4, L/2, L/2)$. The density ratio between the bubble and the outer fluid is again $\rho_i/\rho_o = 10^6$, and the viscosity is set to $\mu = 0$ in the entire domain. The initial velocity $u_p(x, 0)$ is again set via density weighted projection, which is zero for the non-horizontal velocity components. Each case is run until $t = 1$ with a constant time step size $\Delta t = 2\Delta x_0/125$. Coarse grid cells are tagged
for refinement where the local vorticity magnitude exceeds a relative threshold of 0.25, or where $\phi \leq 2\Delta x_0$, which ensures that the surface and interior of the bubble are always placed on the finest grid level. Fig. 2.33 shows evolution of the bubble at three snapshots in time. The top row shows the case for inconsistent transport using the non-conservative integrator, which quickly becomes unstable, as in the two-dimensional case. The bottom row shows the case for consistent transport, which remains stable and exhibits similar
Figure 2.33. Three-dimensional convection of a high density ratio droplet (blue) with density ratio $\rho_i/\rho_o = 10^6$ and initial velocity $u_i(x,0)$. The coarse grid has grid spacing $\Delta x_0 = \Delta y_0 = \Delta y_0 = 1/64$, with each simulation using $\ell = 2$ grid levels and refinement ratio $n_{\text{ref}} = 2$. The coarse grid level is bounded by the red boxes, while the fine grid level is bounded by green boxes. (a)-(c) Three snapshots in time for inconsistent transport of mass and momentum, which becomes unstable shortly after $t = 0.6$. (d)-(f) Three snapshots in time for inconsistent transport of mass and momentum, which remains stable throughout the simulation time.

... distortion because of numerical errors as the two dimensional case with a comparable spatial resolution.
2.7.4. Collapsing water column

This section demonstrates the importance of consistent mass and momentum transport to achieve stability for air-water density ratios. A water column of initial height and width of $a = 5.715 \times 10^{-2}$ m is placed in a two-dimensional computational domain $\Omega = [7a, 1.75a]$ that is discretized using a $4N \times N$ grid. No-slip boundary conditions are imposed along $\partial \Omega$. At the initial time, the bottom left corner of the water column coincides with the bottom left corner of the computational domain. The density of water is $\rho_l = 1000$ kg/m$^3$, the density of air is $\rho_g = 1.226$ kg/m$^3$, the viscosity of water is $\mu_l = 1.137 \times 10^{-3}$ kg · m/s, the viscosity of air is $\mu_g = 1.78 \times 10^{-5}$ kg · m/s, the surface tension coefficient is $\sigma = 0.0728$ N/m, and the gravitational acceleration is $g = 9.81$ m/s$^2$ (directed in the negative $y$-direction). Both fluids are initially at rest. This problem has been studied numerically by Rezende et al. [81] and by Patel and Natarajan [44]. We also consider the analogous case in three spatial dimensions, in which a cubic water block with side length $a = 5.715 \times 10^{-2}$ m is placed in a computational domain of size $\Omega = [7a, a, 1.75a]$ discretized by a $4N \times 9N/16 \times N$ grid. In this case, gravity is directed in the negative $z$-direction. This problem has been studied numerically by Gu et al. [82] and experimentally by Martin and Moyce [83]. For all of the following cases, the density is set via the level set function at the beginning of each time step with one grid cell of smearing ($n_{\text{cells}} = 1$) on either side of the interface.

Using $N = 128$ and uniform time step $\Delta t = 1/(62.5N)$, we carry out the two-dimensional simulation using both inconsistent and consistent mass and momentum transport. Fig. 2.34 shows the evolution of the water column over time. Unphysical deformations and numerical instabilities plague the inconsistent approach whereas the consistent
transport remains stable and produces a physically accurate solution that compares favorably to the experimental study [83]. To demonstrate the qualitative accuracy of the present method in three dimensions, we carry out the three-dimensional simulation using consistent transport with \( N = 64 \) (Fig. 2.35). The 3D case differs from the 2D case because that the water column makes contact with an additional pair of walls. It is seen that the numerical solution is again physically reasonable and stable, and that the three-dimensional results qualitatively agree with the two-dimensional simulation.

Finally, we quantitatively compare our results to those of other numerical and experimental studies. The front position and height (nondimensionalized by \( a \)) of the water column is plotted against time (nondimensionalized by \( \sqrt{a/g} \)) in Fig. 2.36. Excellent agreement is demonstrated between the present study and previous works. We again emphasize that this test case demonstrates how vitally important consistent mass and momentum transport is for stable simulation of practical multiphase flows, for which air-water density ratios are ubiquitous.

2.7.5. Droplet splashing on thin liquid film

2.7.5.1. Two spatial dimensions. This section investigates the problem of a droplet splashing on a thin liquid film, which is representative of real-world applications such as spray cooling [84] and inkjet printing [85]. A circular droplet of initial diameter \( D = 1 \) is placed just above a liquid film in a two-dimensional computational domain of size \( \Omega = [0, 8D] \times [0, 2D] \), which is discretized by an \( 4N \times N \) uniform grid with \( N = 400 \). The droplet has initial downward velocity \( \mathbf{u}(\mathbf{x}, 0) = (0, -U) \), with \( U = 1 \), which is projected to produce a discretely divergence-free initial condition. The droplet has initial center
(a) Inconsistent \( t = 0.0 \) s

(b) Consistent \( t = 0.0 \) s

(c) Inconsistent \( t = 0.1 \) s

(d) Consistent \( t = 0.1 \) s

(e) Inconsistent \( t = 0.2 \) s

(f) Consistent \( t = 0.2 \) s

(g) Inconsistent \( t = 0.3 \) s

(h) Consistent \( t = 0.3 \) s

Figure 2.34. Two dimensional evolution of the spreading water column with density ratio \( \rho_l/\rho_g = 815.66 \) and \( \mu_l/\mu_g = 63.88 \) at four different time instances: (left) Inconsistent and (right) consistent transport of mass and momentum is used with grid size \( 512 \times 128 \).

\((X_0, Y_0) = (4D, 0.75D)\) and the liquid film has initial height \( 0.2D \). The boundary normal and tangential velocities are set to zero, imposing no-slip boundary conditions on \( \partial \Omega \).

The density ratio between the liquid droplet (or the film phase) and the surrounding gas phase is \( \rho_l/\rho_g = 815 \), with \( \rho_g = 1 \). The Reynolds number based on the gas phase, \( \text{Re} = \rho_g U D/\mu_g \), is used to determine the gas phase viscosity, and the viscosity ratio between liquid and gas is \( \mu_l/\mu_g = 55 \). The Weber number based on the gas phase,
Figure 2.35. Three dimensional evolution of the spreading water column with density ratio $\rho_l/\rho_g = 815.66$ and $\mu_l/\mu_g = 63.88$ at four different time instances. Consistent transport of mass and momentum is used for this case with grid size $256 \times 36 \times 64$.

We = $\rho_g U^2 D/\sigma = 0.126$, is used to determine the surface tension coefficient $\sigma$. Gravity is neglected for this particular problem, which, because of the high impact velocity, is a convection and surface tension driven flow. Time is nondimensionalized as $T = tU/D$, and each case is run until $T = 1.5$ with a constant time step size of $\Delta t = 1/(10N)$. The conservative discretization is used for all cases considered here, but we found that the non-conservative discretization yields similar results (data not shown). For all of the cases, the density is synchronized via the level set function at the beginning of each time step with one grid cell of smearing ($n_{\text{cells}} = 1$) on either side of the interface. Similar problems have been studied numerically using the VOF methods by Coppola et al. [86] and by Patel and Natarajan [44], and using a lattice Boltzmann method by Li et al. [87].
As a first test we consider $Re = 6.6$ (left panels in Fig. 2.37), which can be directly compared to the results of Patel and Natarajan [44]. It is seen that the droplet merges into the liquid film and produces a thick, symmetric liquid sheet upon impact. Because of the relatively low Reynolds number, no satellite droplets are formed. The results are in excellent qualitative agreement with prior work [44], including resolution of the gas entrapment by the droplet into the liquid film. A second test case is carried out at $Re = 66$ (right panels in Fig. 2.37), which not only produces a splash with higher vertical reach than the lower Reynolds number case but also exhibits satellite droplets that break away from film. The results compare favorably to those of Patel and Natarajan [44]. As described by Coppola et al. [86], the jet base location $x_J$ can be computed by averaging the $x$-position
(a) Re = 6.6, T = 0.2
(b) Re = 66, T = 0.2
(c) Re = 6.6, T = 0.5
(d) Re = 66, T = 0.5
(e) Re = 6.6, T = 0.7
(f) Re = 66, T = 0.7
(g) Re = 6.6, T = 1.0
(h) Re = 66, T = 1.0
(i) Re = 6.6, T = 1.5
(j) Re = 66, T = 1.5

Figure 2.37. Temporal evolution of the droplet splashing on a thin liquid film with density ratio $\rho_l/\rho_g = 815$ and $\mu_l/\mu_g = 55$ at five different time instances with grid size $1600 \times 400$: (left) Re = 6.6 and (right) Re = 66.

of the two neck points of the liquid sheet (see Fig. 2.38(a)). According to the theoretical studies of Josserand and Zaleski [88] and Howison et al. [89], the normalized jet base location $(x_j - X_0)/D$ should scale with the square root of dimensionless time $(Ut/D)^{1/2}$. 
As shown in Fig.2.38(b), our simulation shows excellent quantitative agreement with this power law. These cases demonstrate that surface tension flows with complex merging dynamics are adequately simulated by the present numerical scheme.

### 2.7.5.2. Three spatial dimensions

For our final case, we consider the problem of a three-dimensional droplet splashing on a thin liquid film. Experimental and theoretical consideration \[90, 91\] of this particular problem has inspired numerous numerical studies of this phenomena, including those of Nikolopoulos et al. \[92, 93\] and Manik et al. \[94\]. In particular, we follow a similar setup as described in \[93\]. A spherical droplet of initial diameter \(D_0 = 1\) is placed in a computational domain of size \(\Omega = [0, L]^3 = [0, 8D_0]^3\), which is discretized by a base uniform grid of size \(N^3\) (with \(N = 20\)) and four grid levels \(\ell = 4\) with refinement ratio \(n_{\text{ref}} = 2\). Hence at the finest level, the grid spacing is \(\Delta x_{\text{min}} = \Delta y_{\text{min}} = \Delta z_{\text{min}} = 8/160\). The domain is filled with a liquid film of height \(a\), with the dimensionless height set as \(\bar{a} = a/D_0 = 0.116\) and the droplet has initial
center \((X_0, Y_0, Z_0) = (0.5L, 0.5L, 2.8D_0)\). The density ratio between the liquid droplet (or the film phase) and the surrounding gas phase is \(\rho_l/\rho_g = 1000\), with \(\rho_l = 1000\). The viscosity ratio between liquid and gas is \(\mu_l/\mu_g = 40\), with \(\mu_l = 8.9 \times 10^{-4}\). The droplet has initial downward velocity \(\mathbf{u}(\mathbf{x}, 0) = (0, -U_0)\), which is projected to produce a discretely divergence-free initial condition \(\mathbf{u}_p(\mathbf{x}, 0)\). The Reynolds number based on the liquid phase, \(\text{Re} = \rho_l U_0 D_0 / \mu_l = 11294\), is used to specify the initial downward velocity. The Weber number based on the liquid phase, \(\text{We} = \rho_l U_0^2 D_0 / \sigma = 250\), is used to specify the surface tension coefficient \(\sigma\). The Froude number, \(\text{Fr} = U_0^2 / (ga) = 363\), is used to specify the gravitational constant \(g\). No-slip boundary conditions are applied on all computational boundaries. A constant time step size of \(\Delta t = 12.5 \Delta x_{\text{min}}\) is used, yielding an approximate initial CFL number of 0.125. The CFL number throughout the simulation remains less than 0.5 for all time steps. The conservative discretization is used for this case. The density is synchronized using the level set function at the beginning of each time step with one grid cell of smearing \((n_{\text{cells}} = 1)\) on either side of the interface.

Time is made dimensionless via \(T = (t - t_i) U_0 / D_0\). We again begin by demonstrating the importance of using consistent transport of mass and momentum by attempting to run this case with the non-conservative discretization. As seen in Fig. 2.39, the droplet undergoes significant (unphysical) deformation as it travels towards the liquid film. As a result, a coherent lamella does not form, and we see unphysical impact dynamics. In contrast, Fig. 2.40 shows the post-impact splashing dynamics of the liquid interface when this case is simulated with the conservative and consistent integrator. As the droplet travels downwards, it deforms because of the resistance from the gas phase. Upon impacting the
liquid sheet at around $t_i = 218.75$ ($T = 0$), the droplet is no longer a sphere. A lamella begins to emanate from the film and forms a secondary ring thereafter. The ring eventually splits from the top of the lamella at around $T = 3.5$, which then further disintegrates into smaller satellite droplets. The results are in very good qualitative agreement with [93], with small dissimilarities in the formation of satellite droplets at later times. This can be attributed to the differences in interface tracking methodology and computational set up of the problem.

We plot the dimensionless radial distance at the bottom of the rim ($R_b/D_0$) and dimensionless height of the lamella ($a_l/D_0$) as a function of time in Fig. 2.41. The results for the present work compare favorably to previous computational studies [92, 93, 94] that use the VOF method for interface tracking. This case demonstrates that the complex surface tension and gravity driven splashing dynamics are accurately simulated by the present (consistent) numerical scheme, even when significant adaptive mesh refinement is used.

2.8. Conclusions

This study extended the monolithic variable-coefficient incompressible Navier-Stokes solver of Cai et al. [48] and used it for practical high density ratio and high shear multiphase flow applications. We demonstrated that our solver yields second-order convergence in a variety of conditions where analytic solutions are available, including with nontrivial physical boundary conditions, both on uniform and on locally refined computational grids. Further, we demonstrated the importance of consistent mass and momentum transport to
Figure 2.39. (a)-(c) Temporal evolution of the 3D droplet splashing on a thin liquid film with density ratio $\rho_l/\rho_g = 1000$ and $\mu_l/\mu_g = 40$ at three different time instances. Inconsistent transport of mass and momentum occurs while employing a non-conservative momentum integrator. Contrast these results to those shown in Fig. 2.40 in which consistent transport is used.

maintain numerical stability for convection-dominated high density ratio flows. Achieving consistency between mass and momentum transport also necessitates the use of the conservative form of the momentum equation.

The projection preconditioner described here, and by Griffith [47] and Cai et al. [48], has several distinct advantages over the projection method solver. For variable-coefficient operators, there is an unavoidable commutator error that is associated with using the projection method as a solver. By considering the coupled velocity-pressure Stokes system, we avoid this source of error. Moreover, operator splitting approaches require the specification of artificial boundary conditions for the velocity and pressure fields. These affect the overall solution accuracy of the projection method solver, but only affect the convergence rate of the projection method preconditioner. Finally, we again emphasize
Figure 2.40. (a)-(e) Temporal evolution of the 3D droplet splashing on a thin liquid film with density ratio $\rho_l/\rho_g = 1000$ and $\mu_l/\mu_g = 40$ at five different time instances. (f) Locations of the different refined mesh levels from coarsest to finest: red, green, pink, orange. Consistent transport of mass and momentum is used for these cases. Unlike the results shown in Fig. 2.39, here we observe a physical solution.

that the using the projection method as a preconditioner to the coupled Stokes system is no less efficient than using it as a solver.

We also demonstrated that the solver has scalable convergence properties at both low and high Reynolds numbers and for high density and viscosity ratios. In our numerical experiments, we have seen that a number of factors can affect the convergence of the FGMRES solver. Generally speaking, smoother problems (i.e., that use more grid cells
Figure 2.41. Temporal evolution of (a) dimensionless bottom radius and (b) dimensionless lamella height for the present 3D (•, red) simulations, along with 3D simulation data (■, blue) from Nikolopoulos et al. 2005 [92], 3D simulation data (▲, green) from Nikolopoulos et al. 2007 [92], and 3D simulation data (♦, grey) from Manik et al. [94].

of interface smearing) allow for more rapid linear solver convergence than problems with sharp interfaces. Additionally, harmonic averaging of material properties leads to fewer iterations than simple arithmetic averaging. We also see that larger Reynolds numbers require fewer iterations whereas smaller Reynolds numbers require more iterations. Finally, the subdomain solvers can be further optimized for higher or lower (than air-water combination) density and viscosity ratios by tweaking $\epsilon_{sub}$, or by using stronger Krylov subdomain solvers or multigrid preconditioners.

A key contribution of this chapter is that it extends the first-order conservative discretization algorithm of Desjardins and Moureau [39] and Ghods and Hermann [40] to a second-order accurate scheme. This is achieved here by using a SSP-RK3 time integrator for integrating the mass balance equation and employing higher-order CBC and
TVD satisfying limiters. Our implementation also allows for the use of a forward Euler or SSP-RK2 integrator for the mass balance equation, although we have found that these can become numerically unstable for high density ratio and convection dominated flows. In fact, Sec. 2.6.7 demonstrates that we revert to a first-order accurate solution by switching to the forward Euler mass integrator and upwind limiter as done originally in [39] and [40]. Compatibility between mass and momentum transport is achieved at a discrete level by using the same mass flux in both the density evolution equation and the momentum equation. The consistent discretization is also shown to be well-balanced with respect to the pressure gradient force and the gravitational and surface tension forces. By coupling this robust fluid solver with a level-set approach to interface tracking, we enable the simulation of complex multiphase flows such as spreading, merging, and splashing dynamics. We also demonstrated that level set function can be advected independently of the fluid solver, and can use a different (and possibly a non-CBC satisfying) convective limiter without affecting the overall stability of the scheme. Furthermore, we note that the present consistent mass and momentum update scheme remains stable even when conventional projection method solvers [10, 11] are used [95].

The present numerical method can be easily extended to allow for fluid-structure interaction in presence of multiple phases. In particular, future work involving the coupling between this solver and a constraint-based immersed boundary method (CIB) [13] is already underway. In addition to this, the treatment of the viscous term described here allows for an implicit treatment of eddy viscosity, which is a key ingredient to turbulence modeling using both RANS and LES formulations [96, 97, 98]. Integrating this
solver with CIB and turbulence modeling will enable simulation of many important industrial and engineering applications, including high inertia vehicles, wave-energy converter devices, and windmills. Finally, efforts are also underway to compute initial distance functions from CAD and STL files directly. This will allow complex geometries to be represented on Cartesian grids and further enhance our level set based solvers for realistic applications.
CHAPTER 3

A DLM immersed boundary method based wave-structure interaction solver for high density ratio multiphase flows

In this chapter we present a robust immersed boundary (IB) method for high density ratio multiphase flows that is capable of modeling complex wave-structure interaction (WSI) problems arising in marine and coastal engineering applications. The IB/WSI methodology is enabled by combining the distributed Lagrange multiplier (DLM) method of Sharma and Patankar (J Comp Phys, 2005) with a robust level set method based multiphase flow solver. The fluid solver integrates the conservative form of the variable-coefficient incompressible Navier-Stokes equations using a hybrid preconditioner and ensures consistent transport of mass and momentum at a discrete level. The consistent transport scheme preserves the numerical stability of the method in the presence of large density ratios found in problems involving air, water, and an immersed structure. The air-water interface is captured by the level set method on an Eulerian grid, whereas the free-surface piercing immersed structure is represented on a Lagrangian mesh. The Lagrangian structure is free to move on the background Cartesian grid without conforming to the grid lines. The fluid-structure interaction (FSI) coupling is mediated via Peskin’s regularized delta functions in an implicit manner, which obviates the need to integrate the hydrodynamic stress tensor on the complex surface of the immersed structure. The IB/WSI numerical scheme is implemented within an adaptive mesh refinement (AMR)
framework, in which the Lagrangian structure and the air-water interface are embedded on the finest mesh level to capture the thin boundary layers and the vortical structures arising from WSI. We use a well-balanced force discretization for gravity force that eliminates spurious velocity currents in the hydrostatic limit due to density variation in the three phases (air, water and solid). We also show that using a non-conservative and an inconsistent fluid solver can lead to catastrophic failure of the numerical scheme for large density ratio variations that are prevalent in WSI applications. An effective wave generation and absorption technique for a numerical wave tank is presented and used to simulate a benchmark case of water wave distortion due to a submerged structure. The numerical scheme is tested on several benchmark WSI problems from numerical and experimental literature in both two and three dimensions to demonstrate the applicability of the IB/WSI method to practical marine and coastal engineering problems.

3.1. Introduction

Wave-structure interaction phenomena are critical design considerations for marine engineers to ensure the safe operability of coastal and offshore structures. In marine and coastal engineering applications, complex floating structures, such as floating oil platforms, wave energy converter (WEC) devices, and foundations of offshore wind turbines, are subject to wave loading, wave run-up, wave scattering, and wave breaking effects, which can severely damage or affect the performance of these structures.

Recently, development of marine renewable energy has received renewed interest within the scientific community due to fluctuating oil prices and the negative impact of fossil fuels on the environment. It is estimated that $2.11 \pm 0.05$ TW of coastal wave energy is
available globally, with equal amounts in the Northern and Southern hemispheres [99]. Simulations of WECs can help increase their power extraction capacity by interrogating the underlying physics. However, many existing numerical models of WECs are based on simplified flow physics, i.e. by assuming inviscid potential flow equations that are mostly linear [100, 101] or weakly nonlinear [102, 103]. Viscous drag in such models is generally accounted for by using the Morison’s equation [104], which is valid only for slender offshore structures. Moreover, the Morison equation has been obtained empirically from experimental measurements for limited wave conditions and is not valid over all flow regimes [105, 106]. Therefore, these methods cannot handle free-surface and wave breaking effects around the structure, which are highly nonlinear in nature. Neglecting realistic sea or ocean conditions can lead to suboptimal design for WEC devices. Fully-resolved wave-structure interaction simulations of WECs are closer to reality as they model all three phases, but are considerably costlier than potential flow models in turnaround time.

Traditionally, fluid-structure interaction (FSI) problems involving the full incompressible Navier-Stokes (INS) system of equations have been modeled by using Arbitrary Lagrangian-Eulerian (ALE) methods [107] on body conforming grids. The main advantage of an ALE-like approach is that the boundary conditions on the fluid-structure [107] or the fluid-fluid [108] interface can be satisfied exactly. For single-phase FSI applications, ALE methods can be used to obtain high-resolution results, albeit at the cost of frequent re-meshing of the entire computational domain due to the structural displacement [109]. However for WSI applications where the air-water interface undergoes non-smooth and
non-continuous topological changes due to wave-breaking processes, the application of ALE methods is not practical.

To overcome these limitations, fictitious domain \cite{7,110} or immersed boundary (IB) methods \cite{5}, combined with level-set \cite{21} or volume of fluid (VOF) approaches \cite{20} are gaining popularity for both single phase FSI applications and two-phase WSI applications. There are two major implementation categories of the IB method — diffuse and sharp. In the diffuse IB approach, the fluid equations are extended inside the structure domain so that regular and fast Cartesian solvers can be used to solve the INS equations everywhere in the computational domain. An additional body force is applied in the structure domain, which is conveniently represented on a Lagrangian mesh to constrain the motion of the fluid occupying the solid region as a rigid body motion. The most efficient way to compute the FSI body force is through distributed Lagrange multipliers (DLM), a method pioneered by Patankar et al. \cite{7}. A fractional time stepping approach is used to impose the DLM-based rigidity constraint, which is suitable for moderate to high Reynolds number flows \cite{110,13}. For zero Reynolds number Stokes flow, the DLM or constraint force needs to be computed simultaneously by solving an extended saddle point system, along with fluid velocity and pressure degrees of freedom. This is because Stokes flow is a purely elliptic system describing a force equilibration process and any fractional time stepping scheme introduces a large numerical error in its solution. Kallimov et al. \cite{111} and Usabiaga et al. \cite{112} describe an efficient preconditioner for the monolithic fluid-DLM solver. The FSI coupling for diffuse IB methods is mediated via Peskin’s regularized delta functions, in which the Lagrangian DLM force is spread onto the background Eulerian grid and the fluid velocity is interpolated onto the Lagrangian
mesh. The use of regularized delta functions smears the fluid-structure interface over a few grid cells (according to the delta function support), which makes the interface diffuse rather than keeping it sharp. In this work we use an efficient fractional time stepping, diffuse DLM approach to model WSI. Sharp IB methods, on the other hand, imposes the velocity of the fluid-structure interface at the nearby “IB nodes”. This is achieved by fitting a spatial polynomial (linear or quadratic) through the solid interface and fluid nodes. The INS equations are solved only at the fluid nodes, with the IB nodes acting as velocity boundary conditions. The velocity and pressure values for the interior solid nodes are zeroed-out during the solution procedure, which then creates the effect of “punctured” domains. The most notable sharp IB method implementations and their extensions have been carried by Sotiropoulos et al. [113], Mittal et al. [114], Udaykumar et al. [115], and Ferziger et al. [116].

There are several advantages and disadvantages to both diffuse and sharp IB methods. For example, diffuse IB methods permit a continuous solution of velocity and pressure in the entire domain, which eliminates “spurious force oscillations” (SFO) in the time histories of the integrated drag and lift quantities for the moving immersed bodies. In contrast, spurious force oscillations are an outstanding issue for the sharp IB methodology because of the punctured domain effect [117, 118, 119]. Since the solution is continuous throughout the domain for diffuse IB methods, there are no issues with “fresh” and “dead” fluid cells when the structure changes it location in the domain, which is an challenging issue for sharp IB methods. Diffuse IB methods also allow for an implicit coupling of the fluid and structure domains without requiring hydrodynamic stress tensor computations on the (possibly complex) surface of the immersed structure. In contrast,
sharp IB methods compute pointwise hydrodynamic forces on the immersed surface and often require several fluid and structure solver iterations to converge to a stable solution within a single time step \cite{120}. The main disadvantage of diffuse IB methods is the smearing of the fluid-structure interface over few grid cells, which reduces the accuracy of the solution near the interface. The order of accuracy for diffuse IB methods is generally between one and two; the former for non-smooth and the latter for sufficiently smooth FSI problems. In contrast, sharp IB methods retain full second-order accuracy by sharply resolving the fluid-structure interface. Diffuse IB methods are also known to produce non-smooth pointwise hydrodynamic stress on the immersed surface even though the net hydrodynamic force and torque are smooth and SFO-free. This issue can however be mitigated by interpolating the hydrodynamic stress sufficiently far away from the fluid-structure interface. The lack of geometric information for the immersed surface also makes the implementation of wall functions required for turbulence modeling difficult for diffuse IB methods. For sharp IB methods, application of Robin-type boundary conditions and implementing wall functions is quite natural. The SFO in sharp IB methods can be mitigated by increasing the grid resolution and using larger time steps. However, very refined meshes can make the simulations extremely expensive and the use of large time steps can make them unstable unless fully implicit time stepping schemes are used for the INS equations. The demarcation of grid nodes into “IB nodes”, “fluid nodes” and “solid nodes” is a computationally taxing task as well and a novice procedure to reconstruct the IB node velocity (from interface and fluid nodes) can lead to numerical instabilities for certain geometric configurations of the interface relative to the background Cartesian grid \cite{121,116,114}. We remark that in spite of the aforementioned shortcomings of the
diffuse and sharp IB methods, both have been applied successfully to solve complicated engineering problems. Combined with level set or volume of fluid methods that can capture the air-water interface on Eulerian grids, these IB methods allow for an efficient solution of topologically complex WSI problems.

An issue that is unique to WSI or two-phase multiphase flows is the presence of highly contrasting density ratios in the computational domain. High density ratio multiphase flows are known to develop numerical instabilities whenever convection is the dominant physical process \cite{119, 44, 37, 38, 51, 39, 40}. Recently the multiphase community (including us) has proposed several stabilizing remedies for convection-dominated, high density ratio multiphase flows, for solvers based on both volume of fluid and level set methods \cite{44, 51}. The underlying cause of the instability is the inconsistent transport of mass and momentum at a discrete level. In this work we achieve a consistent transport of mass and momentum by solving an additional mass balance equation using a strong-stability preserving Runge-Kutta (SSP-RK3) integrator \cite{42}. The mass flux that updates the density variable is also used to construct a discrete convective operator for the momentum equation. This necessarily requires solving the conservative form of the mass balance and momentum equations. The strong coupling between (discrete) mass and momentum convective operators preserves the stability of the numerical scheme for density ratios as high as $10^6$. Our multiphase flow solver is based on the level set method, which makes the implementation of the proposed IB/WSI methodology relatively easier (than VOF methods) on locally refined meshes. We employ a hybrid preconditioner that solves the velocity and pressure degrees of freedom simultaneously, i.e., we do not use a projection-method
(which is an operator-splitting approach) to solve the INS equations [48]. Only the distributed Lagrange multipliers for the FSI coupling are imposed via operator-splitting. For computational efficiency the air-water interface and the immersed structure are resolved on the finest mesh level, whereas the rest of the computational domain is resolved on progressively coarser grids. Therefore, we are able to capture important flow features at a substantially reduced computational cost, especially in 3D. Since we extend the fluid equations inside the solid domain and since the density of the structure is different than surrounding fluid (almost always heavier than air for WSI applications), the gravitational body force can produce spurious velocity currents near the fluid-structure interface for certain cases. Similarly, due to the density contrast of air and water, spurious velocity currents can also form near the air-water interface. In this work we employ a well-balanced gravity force discretization that eliminates such spurious currents near the air-water-solid interface even in the hydrostatic limit. Section 3.8.4 provides a numerical example that highlights this problem and shows the numerical “fix”.

The remainder of the chapter is organized as follows. We first introduce the continuous and discrete system of equations in Secs. 3.2 and 3.3, respectively. Next we discuss the solution methodology in Sec. 3.4. Section 3.5 comments on the well-balanced gravity force implementation and Sec. 3.8.4 presents the corresponding numerical example. Software implementation is described in Sec. 3.6. Section 3.7 describes the implementation of a numerical wave tank based on the level set method, and demonstrates the interaction of a Stokes second-order wave in the presence of a submerged structure. Finally, more complicated three-phase flow examples that demonstrate the applicability of the proposed
IB/WSI methodology to simulate free-surface piercing and floating structures are presented in Sec. 3.8. We also contrast the consistent results from the conservative flow solver against the unstable results obtained from an inconsistent and non-conservative flow solver to highlight the importance of consistent mass and momentum transport for practical WSI applications. Wherever possible, simulation results from locally refined grids are presented.

3.2. The continuous equations of motion

3.2.1. Multiphase constraint immersed boundary formulation

We begin by stating the governing equations for a multiphase fluid-structure system occupying a fixed region of space \( \Omega \subset \mathbb{R}^d \), for \( d = 2 \) or 3 spatial dimensions. In the immersed boundary formulation, a fixed Eulerian coordinate system \( x = (x_1, \ldots, x_d) \in \Omega \) is used to describe the momentum equation and divergence-free condition for both the fluid and structure. It is convenient to employ a Lagrangian description of the immersed body configuration, in which \( s = (s_1, \ldots s_d) \in B \) denotes the fixed material coordinate system attached to the structure and \( B \subset \mathbb{R}^d \) is the Lagrangian curvilinear coordinate domain. The position of the immersed structure occupying a volumetric region \( V_b(t) \subset \Omega \) at time \( t \) is denoted by \( X(s, t) \). In contrast with the previous formulation of the DLM or constraint immersed boundary method [13], we allow for a spatially and temporally varying density \( \rho(x, t) \) and dynamic viscosity \( \mu(x, t) \), implying that the structure can be heavier or lighter than the surrounding fluids. Hence, the equations of motion for the coupled
fluid-structure system in conservative form are

\[
\begin{align*}
\frac{\partial \rho \mathbf{u}(x,t)}{\partial t} + \nabla \cdot \rho \mathbf{u}(x,t)\mathbf{u}(x,t) &= -\nabla p(x,t) + \nabla \cdot [\mu (\nabla \mathbf{u}(x,t) + \nabla \mathbf{u}(x,t)^T)] \\
&\quad + \rho \mathbf{g} + \mathbf{f}_s(x,t) + \mathbf{f}_c(x,t), \tag{3.1}
\end{align*}
\]

\[
\nabla \cdot \mathbf{u}(x,t) = 0, \tag{3.2}
\]

\[
\mathbf{f}_c(x,t) = \int_B \mathbf{F}(s,t) \delta(x - \mathbf{X}(s,t)) \, ds, \tag{3.3}
\]

\[
\mathbf{U}(s,t) = \int_\Omega \mathbf{u}(x,t) \delta(x - \mathbf{X}(s,t)) \, dx, \tag{3.4}
\]

\[
\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t). \tag{3.5}
\]

Eqs. (3.1) and (3.2) are the incompressible Navier-Stokes momentum and continuity equations written in Eulerian form, in which \( \mathbf{u}(x,t) \) is the velocity, \( p(x,t) \) is the pressure, and \( \mathbf{f}_c(x,t) \) is the Eulerian constraint force density, which is non-zero only in the structure region. The gravitational acceleration is denoted by \( \mathbf{g} = (g_1, \ldots, g_d) \), and \( \mathbf{f}_s(x,t) \) is the continuum surface tension force. The interactions between Eulerian and Lagrangian quantities are facilitated by Dirac delta function kernels, in which the \( d \)-dimensional delta function is \( \delta(x) = \prod_{i=1}^d \delta(x_i) \). Eq. (3.3) converts the Lagrangian force density \( \mathbf{F}(s,t) \) into an equivalent Eulerian density \( \mathbf{f}_c(x,t) \), in an operation called force spreading. Eq. (3.4) determines the physical velocity of each Lagrangian material point from the background Eulerian velocity field in an operation called velocity interpolation. This ensures that the immersed structure moves according to the local value of the velocity field \( \mathbf{u}(x,t) \) (Eq. (3.5)), and thus the no-slip condition is satisfied at fluid-solid interfaces. Using short-hand notation, the force spreading operation is denoted by \( \mathbf{f}_c = \mathcal{S}[\mathbf{X}^t] \mathbf{F} \), in which
\( S[X] \) is the force-spreading operator and the velocity interpolation operation is denoted by \( \frac{\partial X}{\partial t} = U = \mathcal{J}[X]u \), in which \( \mathcal{J}[X] \) is the velocity-interpolation operator. It can be shown that if \( S \) and \( \mathcal{J} \) are taken to be adjoint operators, i.e. \( S = \mathcal{J}^* \), then the Lagrangian-Eulerian coupling conserves energy [5].

The specific rigidity constraint imposed within the structure domain, written in Lagrangian form, is given by

\[
(3.6) \quad \frac{1}{2} \left[ \nabla U(s, t) + \nabla U(s, t)^T \right] = 0,
\]

which states that the body has zero deformation rate and must undergo a rigid body motion [7]. In the present work, we compute a discrete approximation to the constraint force \( F(s, t) \), although a numerical method that enforces this constraint exactly for a range of Reynolds numbers (including zero Reynolds number Stokes flow) has been described by one of us in [111, 112].

Note that the momentum equation (Eq. (3.1)) can also be cast to an equivalent non-conservative form. However, it has been shown that direct discretization of the non-conservative form can lead to numerical instabilities for high density ratio multiphase flows [37, 38, 39, 40]. The differences between the conservative and non-conservative flow solvers will be discussed in later sections.

### 3.2.2. Interface tracking for material properties

Next, we describe the governing equations for tracking and transporting material properties. Suppose a liquid of density \( \rho_l \) and viscosity \( \mu_l \) occupies a region \( \Omega_l(t) \subset \Omega \), while a gas of density \( \rho_g \) and viscosity \( \mu_g \) occupies a region \( \Omega_g(t) \subset \Omega \). The codimension-1
interface between these two fluids is denoted by $\Gamma(t) = \Omega_l \cap \Omega_g$ can be tracked as the zero contour of a scalar function $\phi(x, t)$, which is the so-called level set function \cite{21, 59, 49},

$$\Gamma(t) = \{ x \in \Omega \mid \phi(x, t) = 0 \}.$$  

(3.7)

Level set methods are particularly well-suited for tracking liquid-gas interfaces undergoing complex topological changes and are relatively simple to implement in both two and three spatial dimensions, and on locally refined meshes. It is also useful to define an additional level set function $\psi(x, t)$ to track the boundary of the immersed structure $S_b(t) = \partial V_b(t)$. Using this auxiliary field, the density $\rho_s$ and the viscosity $\mu_s$ in the solid region can be readily prescribed in the Eulerian regions occupied by the solid $V_b(t) \subset \Omega$. Both level set functions are passively advected by the incompressible fluid velocity, which in conservative form reads

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = 0,$$

(3.8)

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \psi \mathbf{u} = 0.$$

(3.9)

The material properties including density and viscosity in the three phases are determined as a function of these two scalar fields by

$$\rho(x, t) = \rho(\phi(x, t), \psi(x, t)),$$

(3.10)

$$\mu(x, t) = \mu(\phi(x, t), \psi(x, t)).$$

(3.11)

The discretized form of Eqs. (3.10) and (3.11) are defined in Section 3.4.1 using regularized Heaviside functions.
One particularly useful level set function is the **signed distance function**, which can be prescribed as initial conditions to Eqs. (3.8) and (3.9)

\[
\phi(x, 0) = \begin{cases} 
\min_{y \in \Gamma(0)} ||x - y||, & x \in \Omega_g(0), \\
- \min_{y \in \Gamma(0)} ||x - y||, & x \in \Omega_l(0), 
\end{cases}
\]

(3.12)

\[
\psi(x, 0) = \begin{cases} 
\min_{y \in S_b(0)} ||x - y||, & x \notin V_b(0), \\
- \min_{y \in S_b(0)} ||x - y||, & x \in V_b(0). 
\end{cases}
\]

(3.13)

However, we note that \( \phi \) and \( \psi \) generally will not remain signed distance functions under advection by Eqs. (3.8) and (3.9). A **reinitialization** or redistancing procedure is used to maintain the signed distance property at every time step. When the fluid properties are determined from the level set fields, we need initial conditions for \( \phi \) and \( \psi \) but not for \( \rho \) or \( \mu \).

### 3.3. Spatial discretization

This section describes the discrete form of the governing equations for the coupled fluid-structure system. Eulerian quantities are discretized on a staggered Cartesian grid, whereas Lagrangian quantities are approximated on a collection of immersed markers that can be arbitrarily positioned on the grid. A regularized version of the Dirac delta function is used to facilitate the velocity interpolation and force spreading operations. Therefore, we are not employing a body-conforming mesh to the fluid-structure interface since the structure markers need not conform to the Eulerian grid.
Throughout this section, we describe the discretization for $d = 3$ spatial dimensions; the discretization in two spatial dimensions is analogous. For simplicity, we describe the case for which there is no local grid refinement in the domain, although this is not a limitation of the present formulation. Details on adaptive mesh refinement are delegated to Sec. 3.3.4. Finally, we note that evaluating the discrete operators described in this section near boundaries of the computational domain and locally refined mesh boundaries requires the specification of adjacent “ghost” cells. For more details on the treatment of boundary conditions and coarse-fine interfaces, we refer readers to [13, 47, 72].

3.3.1. Eulerian discretization

We employ a staggered grid discretization for quantities described in the Eulerian frame. A $N_x \times N_y \times N_z$ Cartesian grid covers the physical, rectangular domain $\Omega$ with mesh spacing $\Delta x$, $\Delta y$, and $\Delta z$ in each direction. Without loss of generality, we assume that the bottom left corner of the domain is situated at the origin $(0, 0, 0)$. Therefore, each cell center of the grid has position $x_{i,j,k} = ((i + \frac{1}{2})\Delta x, (j + \frac{1}{2})\Delta y, (k + \frac{1}{2})\Delta z)$ for $i = 0, \ldots, N_x - 1$, $j = 0, \ldots, N_y - 1$, and $k = 0, \ldots, N_z - 1$. For a given cell $(i, j, k)$, $x_{i-\frac{1}{2},j,k} = (i\Delta x, (j + \frac{1}{2})\Delta y, (k + \frac{1}{2})\Delta z)$ is the physical location of the cell face that is half a grid space away from $x_{i,j,k}$ in the $x$-direction, $x_{i,j-\frac{1}{2},k} = ((i + \frac{1}{2})\Delta x, j\Delta y, (k + \frac{1}{2})\Delta z)$ is the physical location of the cell face that is half a grid cell away from $x_{i,j,k}$ in the $y$-direction, and $x_{i,j,k-\frac{1}{2}} = ((i + \frac{1}{2})\Delta x, (j + \frac{1}{2})\Delta y, k\Delta z)$ is the physical location of the cell face that is half a grid cell away from $x_{i,j,k}$ in the $z$-direction. The pressure degrees of freedom are approximated at cell centers and are denoted by $p_{i,j,k}^n \approx p(x_{i,j,k}, t^n)$, in which $t^n$ is the
time at time step $n$. Similarly, the flow and structure level set functions are also defined at cell centers and are denoted by $\phi_{i,j,k}^n \approx \phi \left( \mathbf{x}_{i,j,k}, t^n \right)$ and $\psi_{i,j,k}^n \approx \psi \left( \mathbf{x}_{i,j,k}, t^n \right)$, respectively.

Velocity components are staggered and defined on their respective cell faces: $u_{i-\frac{1}{2},j,k}^n \approx u \left( \mathbf{x}_{i-\frac{1}{2},j,k}, t^n \right)$, $v_{i,j-\frac{1}{2},k}^n \approx v \left( \mathbf{x}_{i,j-\frac{1}{2},k}, t^n \right)$, and $w_{i,j,k-\frac{1}{2}}^n \approx w \left( \mathbf{x}_{i,j,k-\frac{1}{2}}, t^n \right)$. The components of various body forces on the right-hand side of the momentum equation (Eq. (3.1)) are similarly approximated on respective faces of the staggered grid. The density and viscosity are approximated at cell centers and are denoted by $\rho_{i,j,k}^n \approx \rho \left( \mathbf{x}_{i,j,k}, t^n \right)$ and $\mu_{i,j,k}^n \approx \mu \left( \mathbf{x}_{i,j,k}, t^n \right)$. These quantities are interpolated onto the required degrees of freedom as needed.

Standard second-order finite differences are used to approximate spatial derivative operators and are denoted with $h$ subscripts; i.e. $\nabla \approx \nabla_h$. The full description of these staggered grid discretizations have been recorded in various prior studies and we refer readers to [48, 47, 61, 62] for more details.

### 3.3.2. Lagrangian discretization

Quantities attached to the structure are described in a Lagrangian frame on immersed markers that are free to arbitrarily cut through the background Cartesian mesh. These nodes are indexed by $(l, m, n)$ with curvilinear mesh spacings $(\Delta s_1, \Delta s_2, \Delta s_3)$. An arbitrary quantity can be discretely approximated on a marker points as $\Phi_{l,m,n}^n \approx \Phi \left( \mathbf{s}_{l,m,n}, t^n \right) = \Phi \left( l\Delta s_1, m\Delta s_2, n\Delta s_3, t^n \right)$ at time $t^n$. Henceforth the position, velocity, and force of a marker point are denoted as $\mathbf{X}_{l,m,n}$, $\mathbf{U}_{l,m,n}$, and $\mathbf{F}_{l,m,n}$. In this work we only consider rigid bodies without any constitutive model applied in the structure domain, and therefore
Figure 3.1. (a) Sketch of the immersed structure contained within a domain containing liquid and gas phases. (b) Numerical discretization of the domain $\Omega$ into Eulerian grid cells (■, purple) and Lagrangian markers (■, orange). (c) A single Cartesian grid cell on which the components of the velocity field $\mathbf{u}$ are approximated on the cell faces ($\rightarrow$, black); the pressure $p$ and level sets $\phi$ and $\psi$ are approximated on the cell center (●, black); and the Lagrangian quantities are approximated on the marker point (■, orange), which can be arbitrarily placed on the Eulerian grid. Explicit mesh connectivity information is not needed [13]. See Fig. 3.1 for a sketch of the discretization in two spatial dimensions.
3.3.3. Lagrangian-Eulerian interaction

Finally, the transfer of quantities between the Eulerian and Lagrangian coordinate systems requires discrete approximations to the velocity interpolation and force spreading operations to be defined. We briefly summarize them here to complete the description of the spatial discretization.

3.3.3.1. In the interior domain. For a given fluid velocity defined on faces of the staggered grid, the discretized velocity interpolation operation for a particular configuration of Lagrangian markers (i.e. $U = \mathcal{J}_h[\mathbf{X}]\mathbf{u}$) away from the physical boundary follows the standard treatment

\[
U_{l,m,n} = \sum_{x_{i,j,k} \in \Omega} u_{i-\frac{1}{2},j,k} \delta_h \left( x_{i-\frac{1}{2},j,k} - X_{l,m,n} \right) \Delta x \Delta y \Delta z, \tag{3.14}
\]

\[
V_{l,m,n} = \sum_{x_{i,j,k} \in \Omega} v_{i,j-\frac{1}{2},k} \delta_h \left( x_{i,j-\frac{1}{2},k} - X_{l,m,n} \right) \Delta x \Delta y \Delta z, \tag{3.15}
\]

\[
W_{l,m,n} = \sum_{x_{i,j,k} \in \Omega} w_{i,j,k-\frac{1}{2}} \delta_h \left( x_{i,j,k-\frac{1}{2}} - X_{l,m,n} \right) \Delta x \Delta y \Delta z, \tag{3.16}
\]

in which $\delta_h(\mathbf{x})$ is a regularized version of the $d$-dimensional Dirac delta function based on a four-point kernel function [5]. For a given force density defined on Lagrangian markers,
the discretized force spreading operation $f = S_h[X]F$ reads

\[(3.17) \quad (f_1)_{i-\frac{1}{2},j,k} = \sum_{X_{l,m,n} \in V_b} (F_1)_{l,m,n} \delta_h \left( x_{i-\frac{1}{2},j,k} - X_{l,m,n} \right) \Delta s_1 \Delta s_2 \Delta s_3,\]

\[(3.18) \quad (f_2)_{i,j-\frac{1}{2},k} = \sum_{X_{l,m,n} \in V_b} (F_2)_{l,m,n} \delta_h \left( x_{i,j-\frac{1}{2},k} - X_{l,m,n} \right) \Delta s_1 \Delta s_2 \Delta s_3,\]

\[(3.19) \quad (f_3)_{i,j,k-\frac{1}{2}} = \sum_{X_{l,m,n} \in V_b} (F_3)_{l,m,n} \delta_h \left( x_{i,j,k-\frac{1}{2}} - X_{l,m,n} \right) \Delta s_1 \Delta s_2 \Delta s_3.\]

We refer readers to [13, 5] for more details on properties and implementation of the grid transfer operations.

**3.3.3.2. Near the physical boundary.** When a Lagrangian marker is near the physical boundary, the support of the standard IB kernel extends beyond the computational domain. In this case $S$ and $J$ operators are modified to $S_{BC}$ and $J_{BC}$, respectively, to satisfy the discrete adjointness property $S_{BC} = J_{BC}^*$ near the physical boundary. Briefly, $J_{BC}$ is obtained by first filling the ghost cell values abutting the physical domain to satisfy the imposed boundary conditions (say for velocity) and then using the standard weights of the $J$ operator to interpolate onto the Lagrangian marker. The adjoint spreading operator near the boundary $S_{BC}$ is obtained by first spreading to ghost (and interior) cells beyond the physical boundary and then adding back values to the interior cells by identifying their mirror images in the ghosted region. More details on this construction can be found in the Appendix of Kallemov et al. [111].
3.3.4. Adaptive mesh refinement

Some cases presented in this work make use of a structured adaptive mesh refinement (SAMR) framework to discretize the multiphase fluid-structure interaction equations. These discretization approaches describe the computational domain as composed of multiple grid levels, which is hereafter known as a grid hierarchy. Assuming uniform and isotropic mesh refinement, a grid hierarchy with \( \ell \) levels and coarsest grid spacings \( \Delta x_0, \Delta y_0, \) and \( \Delta z_0 \) has grid spacings \( \Delta x_{\text{min}} = \Delta x_0 / n_{\text{ref}}^{\ell-1}, \Delta y_{\text{min}} = \Delta y_0 / n_{\text{ref}}^{\ell-1}, \) and \( \Delta z_{\text{min}} = \Delta z_0 / n_{\text{ref}}^{\ell-1} \) on the finest grid level, in which \( n_{\text{ref}} \) is the integer refinement ratio between levels. Although not considered here, both the numerical method and software implementation allow for general refinement ratios.

The locally refined meshes can be static, in that they occupy a fixed region in the domain \( \Omega \), or adaptive, in that some criteria of interest is used to “tag” coarse cells for refinement. In our current implementation, cells are refined based on two criteria: 1) if the local magnitude of vorticity \( \| \omega \|_{i,j,k} = \| \nabla \times u \|_{i,j,k} \) exceeds a relative threshold and 2) if the flow level set function \( \phi_{i,j,k} \) is within some threshold of zero. This ensures that the important dynamics (e.g., regions of high velocity gradients or the multiphase interfaces) are always approximated on the most resolved mesh. Additionally, we find that restricting the liquid-gas interface to the finest grid level can greatly mitigate spurious mass changes typically seen in level set methods. We also note that the immersed structure is always placed on the finest grid level, ensuring adequate accuracy near the fluid-solid interface. We refer readers to prior work by Griffith [72] for additional details on the AMR discretization methods, which includes a description of the refine and coarsen operations.
carried out during hierarchy regridding and a treatment of the coarse-fine interface ghost cells.

3.4. Solution methodology

Our strategy for solving the coupled fluid-structure interaction system of equations is similar to that of Bhalla et al. \cite{13}. The numerical method relies on a time-splitting approach, in which we first solve the incompressible Navier-Stokes equations (Eqs. (3.1) and (3.2)) without accounting for the constraints associated with the motion of the immersed body. We then correct the velocity field to comply with the constrained Lagrangian velocity field via a projection step. This section also describes additional complexities related to the multiphase nature of the problems considered in this work.

3.4.1. Interface tracking and reinitialization

As described in Sec. \ref{sec:3.2.2}, two level set functions are defined for the present numerical method: 1) the scalar field $\phi(x,t)$ whose zero contour represents the liquid-air interface $\Gamma(t)$ and 2) the scalar field $\psi(x,t)$ whose zero contour represents the boundary of the immersed structure $S_b(t)$. The transition between different materials on the Eulerian grid can be completely described by these two level set functions. Indeed if $\phi$ and $\psi$ represent signed distance functions to their respective interfaces, we can define smoothed Heaviside functions that have been regularized over $n_{\text{cells}}$ grid cells on either side of the interfaces (assuming $\Delta x = \Delta y = \Delta z$),
\[
\tilde{H}_{i,j,k}^{\text{flow}} = \begin{cases} 
0, & \phi_{i,j,k} < -n_{\text{cells}}\Delta x, \\
\frac{1}{2} \left( 1 + \frac{1}{n_{\text{cells}}\Delta x} \phi_{i,j,k} + \frac{1}{\pi} \sin \left( \frac{\pi}{n_{\text{cells}}\Delta x} \phi_{i,j,k} \right) \right), & |\phi_{i,j,k}| \leq n_{\text{cells}}\Delta x, \\
1, & \text{otherwise,}
\end{cases}
\]

\[
\tilde{H}_{i,j,k}^{\text{body}} = \begin{cases} 
0, & \psi_{i,j,k} < -n_{\text{cells}}\Delta x, \\
\frac{1}{2} \left( 1 + \frac{1}{n_{\text{cells}}\Delta x} \psi_{i,j,k} + \frac{1}{\pi} \sin \left( \frac{\pi}{n_{\text{cells}}\Delta x} \psi_{i,j,k} \right) \right), & |\psi_{i,j,k}| \leq n_{\text{cells}}\Delta x, \\
1, & \text{otherwise,}
\end{cases}
\]

in which we have assumed that the number of transition cells is the same across \( \Gamma \) and \( S_b \).

This is not an inherent limitation of the numerical method, but is true for all the cases considered in the present work. A given material property \( \zeta \) (such as \( \rho \) or \( \mu \)) is then set in the whole domain using a two-step process. First, the material property in the “flowing” phase is set via the liquid-gas level set function

\[
\zeta_{i,j,k}^{\text{flow}} = \zeta_l + (\zeta_g - \zeta_l) \tilde{H}_{i,j,k}^{\text{flow}}.
\]

Next, the material property is set on cell centers throughout the computational domain, taking into account the solid phase

\[
\zeta_{i,j,k} = \zeta_s + (\zeta_{i,j,k}^{\text{flow}} - \zeta_s) \tilde{H}_{i,j,k}^{\text{body}}.
\]

Hence the solid level set always takes precedent over the flow phase. Note that we have assumed that the liquid phase is represented by negative \( \phi \) values and the solid phase
Figure 3.2. Sketch of the two-stage process for setting the density and viscosity in the computational domain. (a) Material properties are first prescribed in the “flowing” phase based on the liquid-gas level set function $\phi$ (—, black) and ignoring the structure level set function $\psi$ (---, orange). (b) Material properties are then corrected in the phase occupied by the immersed body.

is represented by negative $\psi$ values, without loss of generality. Even if $\phi$ and $\psi$ are initially set to be the signed distance function from their respective interfaces, they are not guaranteed to retain the signed distance property under linear advection, Eqs. (3.8) and (3.9). Let $\tilde{\phi}^{n+1}$ denote the flow level set function following an advective transport after time stepping through the interval $[t^n, t^{n+1}]$. The flow level set is reinitialized to obtain a signed distance field $\phi^{n+1}$ by computing a steady-state solution to the Hamilton-Jacobi equation

\begin{align}
\frac{\partial \phi}{\partial \tau} + \text{sgn}(\tilde{\phi}^{n+1}) (||\nabla \phi|| - 1) &= 0, \\
\phi(x, \tau = 0) &= \tilde{\phi}^{n+1}(x),
\end{align}
which will yield a solution to the Eikonal equation $\|\nabla \phi\| = 1$ at the end of each time step. We refer the readers to Sec. 2.3.3 for more details on the specific discretization of Eqs. (3.24) and (3.25), which employs second-order ENO finite differences combined with a subcell-fix method described by Min [28], and an immobile interface condition described by Son [27].

Since we only consider relatively simple body geometries in the present work, we can make use of the positions of the Lagrangian markers to reinitialize the structure level set function. As an example, let a volumetric sphere body with radius $R$ be made up of $N_s$ Lagrangian markers. At time $t^{n+1}$, its center of mass can be computed as

$$X_{\text{com}}^{n+1} = \frac{1}{N_s} \sum_{x_{l,m,n} \in V_{b}^{n+1}} X_{l,m,n}^{n+1},$$

and the structure level set function can be directly recomputed as $\psi^{n+1}(x_{i,j,k}) = \|x_{i,j,k} - X_{\text{com}}^{n+1}\| - R$. (Note that whenever $n$ appears as a superscript, it refers to a time step number, whereas $n$ as a subscript refers to the indexing of Lagrangian particles). For more complicated immersed bodies, one can make use of constructive solid geometry (CGS) concepts or R-functions (see Shapiro [122]) to determine analytical expressions for various signed distance functions. In the present work, we always reinitialize both level set functions every time step.

---

$^1$R-functions tend to smooth sharp corners of geometries. We prefer CGS over R-functions wherever the former is applicable.
3.4.2. Full time stepping scheme

Next, we describe the temporal discretization over the interval \([t^n, t^{n+1}]\) for the coupled fluid-structure equations of motion. We employ \(n_{\text{cycles}}\) cycles of fixed-point iteration per time step, with \(n_{\text{cycles}} = 2\) being used for all the cases in the present work. Note that \(k\) appears in superscript to denote the cycle number. The full time stepping scheme consists of three major operations:

1. Advect the signed distance functions, obtaining \(\phi^{n+1,k+1}\) and \(\psi^{n+1,k+1}\), and the cell-centered viscosity \(\mu^{n+1,k+1}\) using the signed distance functions \(^2\).

2. Solve the incompressible Navier-Stokes equations, obtaining \(\tilde{u}^{n+1,k+1}\) and \(p^{n+\frac{1}{2},k+1}\).

3. Enforce the rigidity constraint, obtaining \(X^{n+1,k+1}\) and the corrected fluid velocity \(u^{n+1,k+1}\).

At the beginning of each time step we set \(k = 0\), with \(u^{n+1,0} = u^n\), \(p^{n+\frac{1}{2},0} = p^{n-\frac{1}{2}}\), \(\phi^{n+1,0} = \phi^n\), \(\psi^{n+1,0} = \psi^n\), and \(X^{n+1,0} = X^n\). At the initial time step \(n = 0\), these quantities are obtained using the prescribed initial conditions. The midpoint, time-centered approximations to these quantities are given by \(u^{n+\frac{1}{2},k} = \frac{1}{2} (u^{n+1,k} + u^n)\), \(\tilde{u}^{n+\frac{1}{2},k} = \frac{1}{2} (\tilde{u}^{n+1,k} + u^n)\), \(\phi^{n+\frac{1}{2},k} = \frac{1}{2} (\phi^{n+1,k} + \phi^n)\), \(\psi^{n+\frac{1}{2},k} = \frac{1}{2} (\psi^{n+1,k} + \psi^n)\), and \(X^{n+\frac{1}{2},k} = \frac{1}{2} (X^{n+1,k} + X^n)\). Below, we describe in detail the solution methodology for all three steps.

\(^2\)We first set the cell-centered viscosity and then use harmonic averaging to interpolate onto the appropriate degrees of freedom.
3.4.2.1. Scalar advection. The level set functions are updated by discretizing Eqs. (3.8) and (3.9), which reads

\[
\frac{\phi^{n+1,k+1} - \phi^n}{\Delta t} + Q\left( u^{n+\frac{1}{2},k}, \phi^{n+\frac{1}{2},k} \right) = 0,
\]

(3.27)

\[
\frac{\psi^{n+1,k+1} - \psi^n}{\Delta t} + Q\left( u^{n+\frac{1}{2},k}, \psi^{n+\frac{1}{2},k} \right) = 0,
\]

(3.28)

in which \( Q(\cdot, \cdot) \) represents an explicit piecewise parabolic method (xsPPM7-limited) approximation to the linear advection terms on cell centers. We refer the readers to [47, 66] for more details on the numerical implementation of this flux limiter. Homogenous Neumann boundary conditions for \( \phi \) and \( \psi \) are imposed on \( \partial \Omega \), using a standard ghost value treatment [61].

3.4.2.2. Incompressible Navier-Stokes solver: Consistent transport. The incompressible Navier-Stokes equations Eqs. (3.1) and (3.2) are discretized and solved for in conservative form as

\[
\frac{\rho^{n+1,k+1}}{\Delta t} \tilde{u}^{n+1,k+1} - \rho^n u^n + C^{n+1,k} = -\nabla h \rho^{n+\frac{1}{2},k+1} + (L_\mu \tilde{u})^{n+\frac{1}{2},k+1} + \rho^{n+1,k+1} g + f_s^{n+\frac{1}{2},k+1},
\]

(3.29)

\[
\nabla_h \cdot \tilde{u}^{n+1,k+1} = 0,
\]

(3.30)

in which \( C^{n+1,k} \) is an explicit cubic upwind interpolation (CUI-limited) [63, 64, 41] approximation to the nonlinear convection term, and

\[
(L_\mu \tilde{u})^{n+1,k+1} = \frac{1}{2} \left[ (L_\mu \tilde{u})^{n+1,k+1} + (L_\mu u)^n \right],
\]

(3.31)
is a semi-implicit approximation to the viscous strain rate with

\[(L_\mu)^n = \nabla_h \cdot \left[ \mu^n \left( \nabla_h u + \nabla_h u^T \right)^n \right].\]

The above time-stepping scheme with \(n_{\text{cycles}} = 2\) is similar to a combination of explicit midpoint rule for the convective term and Crank-Nicolson for the viscous terms. We note that the newest approximation to viscosity \(\mu^{n+1,k+1}\) is obtained via the procedure described in Eqs. (3.22) and (3.23). The newest approximation to density \(\tilde{\rho}^{n+1,k+1}\) in Eq. (3.29) is obtained by solving a discretized mass update equation directly on the faces of the staggered grid from the previous time step and level set synchronized density field \(\rho^n\) (obtained after averaging \(\phi^n\) and \(\psi^n\) onto faces). The discretized density update equation is solved using the third-order accurate strong stability preserving Runge-Kutta (SSP-RK3) time integrator [42] as follows

\[
\tilde{\rho}^{(1)} = \rho^n - \Delta t R(u^a_\text{adv}, \rho^n_\text{lim}),
\]

\[
\tilde{\rho}^{(2)} = \frac{3}{4} \rho^n + \frac{1}{4} \tilde{\rho}^{(1)} - \frac{1}{4} \Delta t R(u^{(1)}_\text{adv}, \tilde{\rho}^{(1)}_\text{lim}),
\]

\[
\tilde{\rho}^{n+1,k+1} = \frac{1}{3} \rho^n + \frac{2}{3} \tilde{\rho}^{(2)} - \frac{2}{3} \Delta t R(u^{(2)}_\text{adv}, \tilde{\rho}^{(2)}_\text{lim}),
\]

in which

\[
R(u_\text{adv}, \varphi_\text{lim}) \approx \left[ \left( \nabla \cdot (u_\text{adv} \varphi_\text{lim}) \right)_{i-\frac{1}{2},j,k}, \left( \nabla \cdot (u_\text{adv} \varphi_\text{lim}) \right)_{i,j-\frac{1}{2},k}, \left( \nabla \cdot (u_\text{adv} \varphi_\text{lim}) \right)_{i,j,k-\frac{1}{2}} \right],
\]

is an explicit CUI-limited approximation to the linear density advection term; \(\varphi\) is either \(\rho\) or \(\tilde{\rho}\). We distinguish \(\tilde{\rho}\), the density vector obtained via the SSP-RK3 integrator
from \( \rho \), the density vector that is set from the level set fields. The subscript “adv” indicates the interpolated advective velocity on the faces of face-centered control volume, and the subscript “lim” indicates the limited value (see Sec. 2.3.2 for details on obtaining advective and flux-limited fields). We remark that the density integration procedure is occurring within the overall fixed-point iteration scheme. We have found it to be crucial to use appropriately interpolated and extrapolated velocities to maintain the second-order accuracy of the INS scheme. To wit, for the first cycle \((k = 0)\), the velocities are

\[
\begin{align*}
\mathbf{u}^{(1)} &= 2\mathbf{u}^n - \mathbf{u}^{n-1}, \\
\mathbf{u}^{(2)} &= \frac{3}{2}\mathbf{u}^n - \frac{1}{2}\mathbf{u}^{n-1}.
\end{align*}
\]

For all remaining cycles \((k > 0)\), the velocities are

\[
\begin{align*}
\mathbf{u}^{(1)} &= \mathbf{u}^{n+1,k}, \\
\mathbf{u}^{(2)} &= \frac{3}{8}\mathbf{u}^{n+1,k} + \frac{3}{4}\mathbf{u}^n - \frac{1}{8}\mathbf{u}^{n-1}.
\end{align*}
\]

Notice that \( \mathbf{u}^{(1)} \) is an approximation to \( \mathbf{u}^{n+1} \), and \( \mathbf{u}^{(2)} \) is an approximation to \( \mathbf{u}^{n+\frac{1}{2}} \). Similarly, \( \dot{\rho}^{(1)} \) is an approximation to \( \dot{\rho}^{n+1} \), and \( \dot{\rho}^{(2)} \) is an approximation to \( \dot{\rho}^{n+\frac{1}{2}} \). To ensure consistent transport of mass and momentum fluxes, the convective derivative in Eq. (2.48) is given by

\[
\begin{align*}
\mathbf{C}\left( \mathbf{u}^{(2)}_{\text{adv}}, \dot{\rho}^{(2)}_{\text{lim}} \mathbf{u}^{(2)}_{\text{lim}} \right) \approx \begin{bmatrix}
\nabla \cdot \left( \mathbf{u}^{(2)}_{\text{adv}} \dot{\rho}^{(2)}_{\text{lim}} \mathbf{u}^{(2)}_{\text{lim}} \right)_{i-\frac{1}{2},j,k} \\
\nabla \cdot \left( \mathbf{u}^{(2)}_{\text{adv}} \dot{\rho}^{(2)}_{\text{lim}} \mathbf{u}^{(2)}_{\text{lim}} \right)_{i,j-\frac{1}{2},k} \\
\nabla \cdot \left( \mathbf{u}^{(2)}_{\text{adv}} \dot{\rho}^{(2)}_{\text{lim}} \mathbf{u}^{(2)}_{\text{lim}} \right)_{i,j,k-\frac{1}{2}}
\end{bmatrix}
\end{align*}
\]
which uses the same velocity $u_{\text{adv}}^{(2)}$ and density $\rho_{\text{lim}}^{(2)}$ used to update $\rho^{n+1}$ in Eq (2.43). This is the key step required to strongly couple the mass and momentum convective operators. Results presented in Sec. 3.8 demonstrate that the consistent discretization is stable for practical air-water density ratio of $10^3$ and produce significantly more accurate results than the inconsistent discretization for realistic three phase WSI simulations.

3.4.2.3. Incompressible Navier-Stokes solver: Inconsistent transport. One can directly use the face-centered density field $\rho^{n+1,k+1}$ obtained through the updated level set information, $\phi^{n+1,k+1}$ and $\psi^{n+1,k+1}$, and integrate the INS equations from $[t^n, t^{n+1}]$. In this scenario the time stepping scheme reads

\begin{align}
\rho^{n+1,k+1} \left( \frac{\tilde{u}^{n+1,k+1} - u^n}{\Delta t} + N^{n+\frac{1}{2},k} \right) &= -\nabla_h \rho^{n+\frac{1}{2},k+1} + (L_\mu \tilde{u})^{n+\frac{1}{2},k+1} \\
&\quad + \rho^{n+1,k+1} g + f_s^{n+\frac{1}{2},k+1},
\end{align}

in which $N^{n+\frac{1}{2},k}$ is an explicit CUI-limited approximation to the nonlinear convection term in non-conservative form (i.e. $N^{n+\frac{1}{2},k} \approx \nabla \cdot (u^{n+\frac{1}{2},k} u^{n+\frac{1}{2},k})$). Integrating INS equations in the above manner decouples the mass and momentum advection and this results in an inconsistent transport of mass flux in the two discrete operators.

The performance of these two solvers are compared for some of the numerical examples considered in Sec. 3.8. In particular, we will show that the non-conservative solver is numerically unstable for highly contrasting air-water density ratios. When stable, both schemes are second-order accurate in time. The continuum surface tension force
\( f_{c}^{n+\frac{1}{2},k+1} \) is computed as a function of the flow level set field \( \phi^{n+\frac{1}{2},k+1} \), and its treatment is described in Chapter 2.

### 3.4.2.4. Incompressible Navier-Stokes solver: Iterative solver.

We obtain the updated velocity \( \tilde{u}^{n+1,k+1} \) and pressure \( p^{n+\frac{1}{2},k+1} \) fields by simultaneously solving Eqs. (3.29) and (3.30) (or Eqs. (3.42) and (3.43)) using the flexible GMRES (FGMRES) Krylov solver \[73\] preconditioned by a variable-coefficient projection method that is hybridized with a local-viscosity solver \[47, 48\]. The solvers have been shown to be second-order accurate in space and to converge for density and viscosity ratios of up to \( 10^6 \) in Sec. 2.6.2.

Unless otherwise stated, a relative convergence tolerance of \( 10^{-10} \) is specified for the FGMRES solver, which leads to a converged solution in between 1 and 7 iterations for all of the cases considered here.

### 3.4.2.5. Rigid body projection.

In general, the velocity field computed from the conservative (Eqs. (3.29) and (3.30)) and non-conservative (Eqs. (3.42) and (3.43)) flow solvers will not satisfy the constraints placed in the structure domain (Eq. (3.6)). To correct the velocity in \( V_b(t) \), we carry out the following projection step \[13\]

\[
\phi^{n+1,k+1} \left( \frac{u^{n+1,k+1} - \tilde{u}^{n+1,k+1}}{\Delta t} \right) = f_{c}^{n+1,k+1},
\]

in which \( f_{c}^{n+1,k+1} \) is the Eulerian constraint force that imposes the rigidity constraint.

This force can be computed using the difference between the desired body velocity and
the interpolated uncorrected fluid velocity $\Delta U_{l,m,n}^{n+1,k+1}$, yielding

$$f_c^{n+1,k+1} = \frac{\nabla_{n+1,k+1}}{\Delta t} S_h \left[ X^{n+\frac{1}{2},k} \right] \Delta U^{n+1,k+1}$$

$$= \frac{\nabla_{n+1,k+1}}{\Delta t} S_h \left[ X^{n+\frac{1}{2},k} \right] \left( U_b^{n+1,k+1} - J_h \left[ X^{n+\frac{1}{2},k} \right] \tilde{u}^{n+1,k+1} \right),$$

(3.45)

which is nonzero only in the structure domain. A correction of this type ensures that the fluid velocity $u^{n+1,k+1}$ in $V_b(t)$ approximately matches that of the body’s Lagrangian velocity $U_b^{n+1,k+1}$. Combining Eqs. (3.44) and (3.45) yields a succinct update equation for the Eulerian velocity field

$$u^{n+1,k+1} = \tilde{u}^{n+1,k+1} + S_h \left[ X^{n+\frac{1}{2},k} \right] \left( U_b^{n+1,k+1} - J_h \left[ X^{n+\frac{1}{2},k} \right] \tilde{u}^{n+1,k+1} \right),$$

(3.46)

which is identical to the update described by Bhalla et al. [13] for neutrally buoyant (constant density) problems. In fact, we simply reuse an existing implementation [14] of the DLM or constraint immersed boundary method to carry out our multiphase FSI simulations. Note that in general, the corrected velocity field will not satisfy the discrete continuity equation, i.e. $\nabla_h \cdot u^{n+1,k+1} \neq 0$. One could apply an additional velocity projection and pressure correction step to ensure that the final velocity is divergence-free [13], but we have found that is it not necessary to obtain accurate results. As described previously [47], the initial value for pressure at the start of each time step $p^{n+\frac{1}{2},0}$ does not affect the flow dynamics nor the pressure solution at the end of the time step $p^{n+\frac{1}{2}}$; rather it serves as an initial guess to iterative solution of the linear system.

Next, we describe a procedure to determine $U_b^{n+1,k+1}$, which is required to compute $f_c^{n+1,k+1}$. Since the structure is constrained to have a vanishing deformation rate tensor,
the velocity of each Lagrangian marker can be decomposed as the following rigid body motion (dropping the time superscripts for now)

\[(3.47) \quad (U_b)_{l,m,n} = U_r + W_r \times R_{l,m,n},\]

in which \(U_r\) and \(W_r\) represent the linear and angular center of mass velocities, respectively, and \(R_{l,m,n} = X_{l,m,n} - X_{com}\) is the radius vector pointing from the center of mass to the Lagrangian marker position. Two distinct scenarios are considered in the present work:

(1) **Fully prescribed motion:**

For problems in which the motion of the body is specified as a function of time, we can directly set the Lagrangian velocity field at time step \(n + 1\) as

\[(3.48) \quad (U_b)_{l,m,n}^{n+1,k+1} = U_r^{n+1} + W_r^{n+1} \times R_{l,m,n}^{n+1,k},\]

which is then used to update the position of the Lagrangian markers

\[(3.49) \quad X_{l,m,n}^{n+1,k+1} = X_{l,m,n}^{n} + \Delta t (U_b)_{l,m,n}^{n+\frac{1}{2},k+1}.\]

This algorithm can be used to simulate one-way FSI problems such as flows past stationary objects or bodies entering or exiting fluid interfaces with constant velocity.

(2) **Free-body motion:**

For coupled problems in which the body moves as a result of the fluid-structure interaction, we determine the Lagrangian velocity field at time step \(n + 1\) by redistributing the linear and angular momentum \([7, 12, 13]\) in the structure
Here, \( I_b = \sum x_{l,m,n} \in V_b \rho_s \left( R_{l,m,n}^{n+\frac{1}{2},k} \cdot R_{l,m,n}^{n+\frac{1}{2},k} I - R_{l,m,n}^{n+\frac{1}{2},k} \otimes R_{l,m,n}^{n+\frac{1}{2},k} \right) \) is the moment of inertia tensor, in which \( I \) is the \( d \)-dimensional identity tensor, and \( M_b = \sum x_{l,m,n} \in V_b \rho_s \Delta s_1 \Delta s_2 \Delta s_3 \) is the mass of the body. Note that since we assume a uniform density in the solid region, the contribution from \( \rho_s \) cancels out in the actual implementation of Eqs. (3.50) and (3.51). Hence, buoyancy effects due to differences in the fluid and solid densities are implicitly accounted for by the multiphase fluid solver. Once the rigid body velocity components are determined, the structure’s velocity and position are updated via Eqs. (3.48) and (3.49).

We remark that the above formulation assumes that the six rigid degrees of freedom either are all fully prescribed (locked) or all undergoing free-body motion (unlocked). This is not a limitation of the implementation: we are able to mix and match which degrees of freedom are locked and unlocked. Many of the numerical examples considered in the present work make use of this flexibility. Finally, we make two interesting observations in the rigid body projection algorithm: 1) the fluid-structure coupling is implicit, i.e., we are not iterating back-and-forth between a fluid and a rigid body integrator and 2) we do not need to explicitly evaluate the hydrodynamic stress on the immersed structure to displace it or solve the fluid equations with internal velocity boundary conditions. The physical reason behind this implicit coupling can be understood if we consider the
hydrodynamic force as an internal force of the system, which is equal and opposite at the fluid-structure interface. This is the essence of the fast and efficient DLM method of Sharma and Patankar [110].

3.5. Prescription of solid density, viscosity and a well-balanced gravitational force

In the case of a neutrally buoyant structure within a single phase flow, the density and viscosity within $V_b(t)$ is simply taken to be that of the surrounding fluid (i.e. the constant $\rho$ and $\mu$ used in the momentum equation) [13]. However, the choice of the “virtual” fluid that occupies the solid region for multiphase flow problems warrants additional discussion. Specification of $\rho_s$ and $\mu_s$ in this region is required to ensure that the linear system of equations (3.29) and (3.30) is well-posed.

For the “virtual” viscosity, we follow the recommendation of Patel and Natarajan [119] and set $\mu_s$ equal to that of the largest (most viscous) of all fluids in the problem. In our experience, this choice leads to accurate FSI simulations and reasonably fast convergence of the FGMRES solver. In cases where the object is undergoing free-body motion, e.g. a sedimenting sphere, a proper specification of the solid density is vitally important in order to capture inertia and buoyancy effects due to the structure’s weight. Hence, we must set $\rho_s$ based on the physical properties of the body we are trying to simulate.

In cases where the immersed body’s velocity is fully prescribed, we again set $\rho_s$ equal to that of the largest (most dense) of all the fluids in the problem. Note that when this object is in contact with the less dense phase, the gravitational term $\rho g$ in the momentum equation will generate spurious momentum in the solid phase. These spurious
velocities will contaminate the flow field throughout the duration of simulation and lead to inaccurate results. In order to mitigate this erroneous momentum generation, we compute the gravitational body force using only the flow density field $\rho^{\text{flow}}$ (see Eq. (3.22) and Fig. 3.2(a)). Thus for the fully prescribed kinematics case $\rho_s$ enters only in the linear operator but not as a gravitational body force in the solid region. As we showed the previous Chapter, the gravitational force based on $\rho^{\text{flow}}$ is well-balanced by the pressure gradient term. Hence, we in-effect recover a well-balanced gravity force for the coupled three-phase flow problem as well; we will show in Sec. 3.8 that no parasitic currents are generated at the air-water-structure interface in the hydrostatic limit.

3.6. Software implementation

The numerical algorithm described here is implemented in the IBAMR library [14], which is an open-source C++ simulation software focused on immersed boundary methods with adaptive mesh refinement. All of the numerical examples presented here are publicly available via https://github.com/IBAMR/IBAMR. IBAMR relies on SAMRAI [15, 16] for Cartesian grid management and the AMR framework. Linear and nonlinear solver support in IBAMR is provided by the PETSc library [17, 18, 19]. All of the example cases in the present work made use of distributed-memory parallelism using the Message Passing Interface (MPI) library. Between 4 and 512 processors were used in all the cases described here.

3.7. Wave-structure interaction

In this section, we demonstrate that the present numerical method is capable of modeling complex wave-structure interaction problems arising in marine and coastal engineering.
With begin by describing our implementation of a numerical wave tank (NWT). Although NWTs based on VOF methods have been detailed in the literature \cite{105, 123, 124, 125}, studies based on level set methodology are sparse \cite{126, 127, 128}. Wave generation and wave absorption techniques for NWTs is an active area of research, and there are several strategies recommended in the literature (typically in the context of VOF methods) \cite{124, 125}. In this work, we use a combination of Dirichlet wave generation boundary conditions and a relaxation-based wave damping procedure as our preferred choice. More specifically, by imposing inlet velocity boundary conditions based on Stokes wave theory at one end of the domain we are able to generate nonlinear water waves, which coherently propagate throughout the computational domain, and by smoothly damping the traveling wave over a wavelength long region towards the opposite end, we mitigate the wave reflection and wave interference phenomena.

In Sec. \ref{sec:theory}, we describe some background theory required to simulate a NWT within the present computational methodology. In Sec. \ref{sec:validation}, we present a number of validation cases to demonstrate that the solver is able to accurately produce second-order Stokes waves. In Sec. \ref{sec:interaction} we investigate the problem of second-order Stokes wave interaction with a submerged trapezoid. The material properties of the liquid (gas) phase are set to be that of water (air): $\rho_l = 1 \times 10^3$, $\mu_l = 1 \times 10^{-3}$, $\rho_g = 1.2$, and $\mu_g = 1.8 \times 10^{-5}$. The gravitational acceleration of $g = 9.81$ is directed in the negative y-direction for the 2D simulations presented in this section.
3.7.1. Stokes wave theory and numerics

According to second-order Stokes theory \cite{100}, the wave elevation $\eta(x, t)$ from a mean water depth $d$ is given by

$$\eta(x, t) = \frac{H}{2} \cos(2\pi x - \omega t) + \frac{\pi H^2}{8\lambda} \frac{\cosh(kd) [2 + \cosh(2kd)]}{\sinh^3(kd)} \cos(2kx - 2\omega t),$$

in which $H$ is the peak-to-peak height of the wave, $T$ is the time period, $\omega = 2\pi/T$ is the angular frequency, $\lambda$ is the wavelength, and $k = 2\pi/\lambda$ is the wave number. The horizontal and vertical components of velocity that generate this wave profile are written as

$$u_w(x, y, t) = \frac{Hgk}{2\omega} \frac{\cosh[k(d + y)]}{\cosh(kd)} \cos(kx - \omega t) + \frac{3H^2\omega k}{16} \frac{\sinh[2k(d + y)]}{\sinh^4(kd)} \cos(2kx - 2\omega t),$$

$$v_w(x, y, t) = \frac{Hgk}{2\omega} \frac{\sinh[k(d + y)]}{\cosh(kd)} \sin(kx - \omega t) + \frac{3H^2\omega k}{16} \frac{\sinh[2k(d + y)]}{\sinh^4(kd)} \sin(2kx - 2\omega t).$$

Note that in the above expressions for the theory and numerics presented in this section, we are considering a domain with bottom left corner situated at $(0, -d)$, without loss of generality. Since the water phase is represented by negative signed distance values and the free surface is initially located at $y = 0$, the elevation of the wave can be computed from $\phi_{i,j}$ via

$$\eta_{i,j} = -\phi_{i,j} + y_{i,j},$$
in which \( y_{i,j} \) is the \( y \)-coordinate of grid cell \( x_{i,j} \). Since \( \phi_{i,j} \) represents the signed distance function to the interface, it is straightforward to show that the computed elevation \( \eta_{i,j} \) will only be a function of the horizontal grid index \( i \), i.e. \( \eta_{i,m} = \eta_{i,n} \) for all \( m, n = 0, \ldots, N_y - 1 \).

At the inlet (left) boundary, we impose the desired velocities Eqs. (3.53) and (3.54) as boundary conditions acting only in the liquid phase. For the normal velocity component, we compute the face-centered level set value based on the analytical elevation value along the computational boundary, \( \phi_{n - \frac{1}{2},j} = -\eta(0, t^n) + y_{-\frac{1}{2},j} \). The normal velocity boundary condition is then given by \( u_{n - \frac{1}{2},j} = \left( 1 - \tilde{H}_{n - \frac{1}{2},j} \right) u_w \left( 0, y_{-\frac{1}{2},j}, t^n \right) \), where the expression for the numerical Heaviside at the boundary reads

\[
\tilde{H}^{n}_{n - \frac{1}{2},j} = \begin{cases} 
0, & \phi_{n - \frac{1}{2},j} < -n_{\text{cells}} \sqrt{\Delta V}, \\
\frac{1}{2} \left( 1 + \frac{1}{n_{\text{cells}} \sqrt{\Delta V}} \phi_{n - \frac{1}{2},j} + \frac{1}{\pi} \sin \left( \frac{\pi}{n_{\text{cells}} \sqrt{\Delta V}} \phi_{n - \frac{1}{2},j} \right) \right), & |\phi_{n - \frac{1}{2},j}| \leq n_{\text{cells}} \sqrt{\Delta V}, \\
1, & \text{otherwise}.
\end{cases}
\]

In the above expression, \( \sqrt{\Delta V} = \sqrt{\Delta x \Delta y} \) represents a characteristic grid spacing for grids with unequal grid spacing in each direction, e.g. \( \Delta x \neq \Delta y \). Similarly for the tangential velocity component, the desired node-centered level set values can be computed as \( \phi_{n - \frac{1}{2},j - \frac{1}{2}} = -\eta(0, t^n) + y_{-\frac{1}{2},j - \frac{1}{2}} \) with corresponding Heaviside function \( \left( 1 - \tilde{H}_{n - \frac{1}{2},j - \frac{1}{2}} \right) \), which are multiplied by \( u_w \left( 0, y_{-\frac{1}{2},j - \frac{1}{2}}, t^n \right) \) to obtain desired boundary condition. We refer readers to [47] for more details on the imposition of normal and tangential velocity boundary conditions in a staggered flow solver. Note that we are simply imposing homogenous Neumann conditions for the level set value at all domain boundaries. No-slip
boundary conditions are imposed along the bottom and right boundary, while homogeneous tangential velocity and zero pressure boundary conditions are imposed at the top boundary.

In order to mitigate the reflection of waves at the right boundary, we place a damping zone at the downstream end of the computational domain from \( x = x_L \) to \( x = x_U \). We follow the approach described by Jacobsen et al. [125], in which the numerical velocities and level set values are smoothly relaxed at the end of each time step via,

\[
\begin{align*}
    u_{i-\frac{1}{2},j} &= \alpha_{i-\frac{1}{2},j} u_{i-\frac{1}{2},j}^{\text{computed}} + \left(1 - \alpha_{i-\frac{1}{2},j}\right) u_{i-\frac{1}{2},j}^{\text{target}}, \\
    v_{i,j-\frac{1}{2}} &= \alpha_{i,j-\frac{1}{2}} v_{i,j-\frac{1}{2}}^{\text{computed}} + \left(1 - \alpha_{i,j-\frac{1}{2}}\right) v_{i,j-\frac{1}{2}}^{\text{target}}, \\
    \phi_{i,j} &= \alpha_{i,j} \phi_{i,j}^{\text{computed}} + (1 - \alpha_{i,j}) \phi_{i,j}^{\text{target}}.
\end{align*}
\]

In the above expressions, the superscript "computed" indicates the staggered grid velocity and cell-centered level set values computed from the solution methodology described in Sec. 3.4.2 and the superscript "target" indicates the desired analytical values representing still water of depth \( d \). Hence, \( u_{i-\frac{1}{2},j}^{\text{target}} = 0 \), \( v_{i,j-\frac{1}{2}}^{\text{target}} = 0 \), and \( \phi_{i,j}^{\text{target}} = y_{i,j} \). The relaxation parameter \( \alpha \) is smoothly varied from 1, at the interface between the non-relaxed portion of the domain and the damping zone (e.g. \( x_L \)), to 0 at the rightmost computational boundary (e.g. \( x_U \)). For example at cell centers, the functional form of alpha reads,

\[
\alpha_{i,j} = 1 - \frac{\exp\left(\frac{x_{i,j}}{x_{i,j}^{3.5}}\right) - 1}{\exp(1) - 1},
\]

Imposition of pressure or normal traction boundary conditions is possible because of the monolithic velocity-pressure solver.
in which \( \bar{x}_{i,j} = (x_{i,j} - x_L) / (x_U - x_L) \) is the normalized horizontal coordinate varying from 0 to 1 across the length of the damping zone. Analogous expressions are determined for \( \alpha_{i-\frac{1}{2},j} \) and \( \alpha_{i,j-\frac{1}{2}} \). In all of the cases considered in this section, a damping zone of length 10\( d \) is prescribed. Next, we present various numerical examples demonstrating the accuracy of the aforementioned wave generation and damping techniques.

### 3.7.2. Validation of second-order Stokes waves propagating in a NWT

As an initial example we consider a 2D computational domain of size \( \Omega = [0, 68d] \times [-d, 0.3d] \), which is occupied by initially quiescent water of depth \( d = 0.4 \). Air occupies the remainder of the domain from \( y = 0 \) to \( y = 0.3d \). One grid cell of smearing \( n_{\text{cells}} = 1 \) is used on either side of the air-water interface and surface tension forces are neglected. The domain is discretized by a grid of size \( N_x \times N_y \) and a constant time step size of \( \Delta t = 100/(57N_x) \) is used. The wave parameters are chosen to be \( H = 0.05d, T = 9.8995\sqrt{d/g}, \) and \( \lambda = 9.232d \); these are chosen to satisfy the required dispersion relation for (second-order) Stokes waves \[100\],

\[
\omega^2 = gk \tanh(kd).
\]

To quantitatively assess the accuracy of the wave generation boundary conditions, the analytical and simulated elevation computed at a probe situated at \( x = 2.87\lambda \), are plotted against time in Fig. [3.3] for three different grid sizes: 442 \times 66, 884 \times 132, and 1768 \times 264. As the resolution increases, the numerical simulations converge towards the theoretical elevation given by Eq. [3.52]. The errors in maximum elevation attained over the shown time period decrease as the resolution increases, yielding a convergence rate of
Figure 3.3. Convergence study for the temporal evolution of wave elevation at $x = 2.87\lambda$ for a second-order Stokes wave; (—, black) analytical expression given by Eq. 3.52; (…, blue) present simulation for a $442 \times 66$ grid; (−−, green) present simulation for a $884 \times 132$ grid; (----, red) present simulation for a $1768 \times 264$ grid.

1.23 between grid sizes $442 \times 66$ and $884 \times 132$ and a convergence rate of 1.17 between grid sizes $884 \times 132$ and $1768 \times 264$. There are approximately $N_H = 10$ grid cells per wave height and $N_\lambda = 240$ grid cells per wavelength for the finest resolution case considered here, which we hereafter denote as Case A.

For our next example we consider two additional sets of wave parameters, which we denote as Case B and Case C; see Table 3.1 for a full specification of all three cases. These parameters are chosen such that they satisfy the dispersion relation Eq. (3.61) and occupy different locations within the second-order Stokes regime for the wave classification phase space described by Le Méhauté [129] (see Fig. 3.4(a)). Figs. 3.4(b)−3.4(d) show the long-time temporal evolution of elevation for cases A, B, and C, respectively. In all three cases, the numerical wave tank produces elevations that are in excellent agreement with
Table 3.1. Parameter specification for the three second-order Stokes wave cases considered in Sec. 3.7.2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Case A</th>
<th>Case B</th>
<th>Case C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth ((d))</td>
<td>0.4</td>
<td>0.4</td>
<td>2.35</td>
</tr>
<tr>
<td>Wave height ((H))</td>
<td>0.05(d)</td>
<td>0.1(d)</td>
<td>0.05(d)</td>
</tr>
<tr>
<td>Wave period ((T))</td>
<td>(9.8995\sqrt{d/g})</td>
<td>(9.8995\sqrt{d/g})</td>
<td>4.0825\sqrt{d/g})</td>
</tr>
<tr>
<td>Wavelength ((\lambda))</td>
<td>9.232(d)</td>
<td>9.232(d)</td>
<td>2.610(d)</td>
</tr>
<tr>
<td>Domain size</td>
<td>68(d\times1.3d)</td>
<td>68(d\times1.3d)</td>
<td>68(d\times1.3d)</td>
</tr>
<tr>
<td>Cells per wavelength</td>
<td>240</td>
<td>240</td>
<td>68</td>
</tr>
<tr>
<td>Cells per wave height</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Elevation probe location ((x))</td>
<td>2.87(\lambda)</td>
<td>2.87(\lambda)</td>
<td>2.87(\lambda)</td>
</tr>
</tbody>
</table>

These examples show that the present numerical method can be confidently used to simulate second-order Stokes waves across the entire (second-order Stokes) region of applicability.

3.7.3. Wave interaction with a submerged trapezoid

Now we investigate the interaction between second-order Stokes waves and a fully submerged trapezoidal shaped structure. The domain size and numerical parameters are identical to those of Case A in the previous section, and the simulation is carried out on a grid of size 1768 × 264. The top and base of the trapezoid have length 5\(d\) and 27.5\(d\), respectively, and its height is 0.75\(d\); see Fig. 3.6 for a full description of the problem set up. All of the trapezoid’s translational and rotational degrees of freedom are locked and it is fully constrained to remain stationary. Since the structure is fully submerged in one fluid, the issue of parasitic currents due to gravitational force does not arise in this scenario and the solid density is set equal to the water phase density. The primary quantities of interest for this example are the elevation values collected from six stations...
Figure 3.4. (a) Locations of (●, red) Case A, (■, yellow) Case B, and (▲, green) Case C on a phase diagram denoting the applicability of wave theories described by Le Méhauté [129], figure adapted from Holthuijsen [130]. Long-time temporal evolution of wave elevation for second-order Stokes waves with parameters described by (b) Case A, (c) Case B, and (d) Case C; (—, black) analytical expression given by Eq. 3.52; wave elevation for all cases is measured at $x = 2.87\lambda$.

placed along the computational domain. This problem has been studied experimentally by Beji and Battjes [131, 132], and numerically by Kasem and Sasaki [126].
A: Without submerged body

(a) \( t = 0 \)  

(b) \( t = 22 \)  

(c) \( t = 22.5 \)  

(d) \( t = 23 \)  

(e) \( t = 23.5 \)  

(f) \( t = 24 \)  

B: With submerged body

(g) \( t = 0 \)  

(h) \( t = 22 \)  

(i) \( t = 22.5 \)  

(j) \( t = 23 \)  

(k) \( t = 23.5 \)  

(l) \( t = 24 \)  

Figure 3.5. A: Temporal evolution of unobstructed second-order Stokes waves at six different time instances. B: Temporal evolution of second-order Stokes waves interacting with a stationary trapezoidal obstacle at six different time instances.
Visualizations of the evolving unobstructed wavefront from the previous section, and the wave-structure interaction are shown in Fig. 3.5 top half and bottom half panels, respectively. Both simulations show temporally cyclic behavior, although irregular amplitude profiles are exhibited by the WSI case. The irregular wave amplitude corresponds to the wave shoaling effect (reduction of water depth near the shore) caused by the submerged structure. The wave profile results qualitatively agree with those shown in [126].

To quantitatively assess the accuracy of the wave-structure interaction, the temporal evolution of wave elevation at the six stations are shown in Fig. 3.7. The results are in decent agreement with the experimental results described in [131, 132]. Our results are also in excellent agreement with the simulation results of Kasem and Sasaki [126], with minor disagreements being explained by slight differences in the level set discretization, advection, and reinitialization approaches. With the cases described in this section, we have demonstrated that the numerical method described here can be used to accurately
model and solve practical marine engineering problems involving water wave-structure interaction.

### 3.8. Free-surface piercing and floating structure examples

This section investigates several additional 2D and 3D three-phase flow problems to verify the accuracy of the present numerical method. The importance of consistent mass and momentum transport for numerical stability is demonstrated for WSI simulations involving air-water interfaces. We also compare our results to benchmark problems drawn from the multiphase flow literature.

In some of the cases considered in this section the net hydrodynamic force

\[
\mathcal{F}(t) = \oint_{S_b(t)} \mathbf{n} \cdot [-pI + \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)]] \, dS,
\]

is a quantity of interest. Here, \( \mathbf{n} \) is the outward unit normal to the surface of the immersed body. An extrinsic approach to computing these forces is via the Lagrange multiplier method \[13\], which reads as

\[
\mathcal{F}^{n+1} = \sum_{X_{l,m,n} \in V_b} \rho_s \left[ \frac{(U_b)_{l,m,n}^{n+1} - (U_b)_{l,m,n}^n}{\Delta t} - \frac{\Delta U_{l,m,n}^{n+1}}{\Delta t} \right] \Delta s_1 \Delta s_2 \Delta s_3,
\]

in which the discrete approximations of the quantities on the right-hand side are readily available during each time step. In Chapter 5 we also describe an accurate moving control volume approach to computing hydrodynamic forces and torques on immersed bodies using Eulerian grids and without resolving the irregular surface of the immersed structure. Both Lagrangian and Eulerian approaches were shown to be equivalent.
Figure 3.7. Temporal evolution of wave elevation measured at stations (a) S2, (b) S3, (c) S4, (d) S5, (e) S6, and (f) S7, for a 2D second-order Stokes wave interacting with a stationary trapezoidal obstacle (see Fig. 3.6); (●, yellow) experimental data from Beji and Battjes [131, 132]; (---, red) simulation data from Kasem and Sasaki [126]; (—, black) present simulation data.
3.8.1. Cylinder splashing into two fluids

We first consider a cylinder dropping into a fluid-gas interface of modest density ratio of 1.5. The non-conservative flow solver is considered for this problem. A circular cylinder of diameter \( D = 2.5 \times 10^{-3} \) and density \( \rho_s = 1.5 \times 10^3 \) is placed in a two dimensional computational domain of size \( \Omega = [0, 8D] \times [0, 48D] \) with initial center position \((X_0, Y_0) = (4D, 40D)\). The domain is filled halfway from \( y = 0 \) to \( y = 24D \) with a fluid of density \( \rho_l = 1.25 \times 10^3 \); the remainder of the tank, from \( y = 24D \) to \( y = 48D \), is filled with a lighter fluid of density \( \rho_g = 1 \times 10^3 \). The viscosity \( \mu = 1 \times 10^{-3} \) is held constant throughout all three phases. The domain is discretized using a \( N \times 6N \) grid and no-slip boundary conditions are imposed along \( \partial \Omega \). A constant time step size \( \Delta t = 1/(39.0625N) \) is used. This problem has been studied numerically by Ghasemi et al. [50]. Surface tension forces are neglected and one grid cell of smearing (\( n_{\text{cells}} = 1 \)) is used on either side of the interfaces.

Fig. 3.8 shows evolution of the cylinder at various instances in dimensionless time \( T = t\sqrt{g/D} \). At this grid resolution, \( N = 512 \) or 64 cells per diameter (CPD), a cavity is formed in the wake of the cylinder as it penetrates the interface. A symmetric jet forms as the cavity collapses, which shoots upwards and breaks up. Additionally, gas phase entrainment is seen around the cylinder, which is experimentally seen in many three-phase flow problems. To demonstrate the quantitative accuracy of the fluid-structure interaction, the vertical position and vertical velocity are plotted as a function of \( T \) in Fig. 3.9 for both 32 and 64 CPD. Both the interface dynamics and the FSI are in decent agreement with the computational study of Ghasemi et al., with minor disagreements being explained by differences in the interface tracking approaches (a fully-Eulerian VOF.
method is used in [50] to simulate three phase flows). Additionally, the rigid body motion in this work is tracked in a Lagrangian reference frame, while a fully-Eulerian FSI approach is used in [50], which could explain the modest differences in vertical velocities after around $T = 65$ in Fig. 3.9(b) this is when the immersed cylinder approaches the bottom of the computational domain. Even though convection is a dominant process here, this numerical test case demonstrates that the non-conservative and inconsistent mass and
momentum transport scheme can still accurately simulate low density ratio three-phase flows.

3.8.2. Heaving cylinder on an air-water interface

This section investigates the heave decay of a cylinder floating on an air-water interface. A circular cylinder of radius $R = 0.0762$ is placed within a two-dimensional computational domain of length $L = 10$ and height $H = 0.2L$. Water occupies the bottom portion of the domain from $y = -16R$ to $y = 0$, while air occupies the remainder of the tank from $y = 0$ to $y = H - 16R$. The cylinder is partially submerged in the fluid phase with initial center position $(X_0, Y_0) = (L/2, R/3)$ and is half as dense as water with $\rho_s = 5 \times 10^2$. Two grid cells of smearing $n_{\text{cells}} = 2$ are used to transition between different material properties on
Figure 3.10. Temporal evolution of a cylinder heaving on an air-water interface at four different time instances with 46 CPR: (left) density and (right) vorticity generated in the range of $-25$ to $25$.

either side of the interfaces. Only the cylinder’s vertical degrees of freedom are unlocked and surface tension forces are neglected. No-slip boundary conditions are imposed along $\partial \Omega$. The conservative and consistent flow solver is used for this case. This problem has been studied both experimentally by Ito $^{133}$, and numerically by Calderer et al. $^{120}$ and Ghasemi et al. $^{50}$. 
The domain is discretized by grid of size $5N \times N$ and a constant time step size of $\Delta t = 3/(5N)$ is used. To assess convergence, we consider three different grid sizes: $N = 300, 600, 1200$, which correspond to 11, 23, and 46 grid cells per radius (CPR), respectively. Fig. 3.10 shows the evolution of the cylinder and the air-water interface at various instances in dimensionless time $T = t\sqrt{g/R}$ for the 46 CPR simulation. Modest vorticity is generated as the cylinder bobs up and down, and small ripples can be seen traveling outward away from the body along the air-water interface. To quantitatively assess the accuracy and convergence of the fluid-structure interaction, the vertical center of mass position of the cylinder (nondimensionalized by $Y_0$) is plotted against time in Fig. 3.11. As the resolution increases, the numerical simulations converge towards the
experimental results of Itô [133]. As expected, the cylinder’s heave oscillation eventually damps out as it reaches an equilibrium position on the water. This case is representative of real-world applications such as wave energy converter devices, and demonstrates that the present numerical method can be used to accurately simulate floating objects. We also simulated this case with the non-conservative flow solver and obtained similar results (data not shown). Since the flow around the heaving cylinder is relatively moderate and decays over time, the non-conservative solver remains stable even for a high density ratio of $10^3$. This will not be true for some of the cases shown later.

3.8.3. Heaving sphere on an air-water interface

As an extension to the case presented in the previous section, we now consider the heave decay of a three-dimensional sphere floating on an air-water interface. A sphere of radius $R = 0.254$ is placed within a three-dimensional computational domain that is equally long in each direction: $L_x = L_y = L_z = 5$. Water occupies the bottom portion of the domain from $z = -12.6R$ to $z = 0$, while air occupies the remainder of the tank from $z = 0$ to $z = L_z - 12.6R$. The sphere is partially submerged in the fluid phase with initial center position $(X_0, Y_0, Z_0) = (L_x/2, L_y/2, R/3)$ and is half as dense as water with $\rho_s = 5 \times 10^2$. Two grid cells of smearing $n_{\text{cells}} = 2$ are used to transition between different material properties on either side of the interfaces. Only the sphere’s vertical degrees of freedom are unlocked and surface tension forces are neglected. The conservative and consistent flow solver is used for this case. No-slip boundary conditions are imposed along $\partial \Omega$. This case has been studied both experimentally by Beck and Liapis [134], and numerically by Pathak and Raessi [51].
Figure 3.12. Temporal evolution of a sphere heaving on an air-water interface at two different time instances with 40 CPR: (top) density and (bottom) locations of the different refined mesh levels from coarsest to finest: red, green, pink, orange.

In contrast with the 2D case, these simulations make use of adaptive mesh refinement. The domain is discretized by $\ell = 4$ grid levels, each with refinement ratio $n_{\text{ref}} = 2$. Two simulations are carried out: one with $\Delta x_0 = \Delta y_0 = \Delta y_0 = 1/10$ yielding a finest grid spacing of $\Delta x_{\text{min}} = \Delta y_{\text{min}} = \Delta z_{\text{min}} = 1/80$ or 20 grid cells per radius (CPR), and one with $\Delta x_0 = \Delta y_0 = \Delta y_0 = 1/20$ yielding a finest grid spacing of $\Delta x_{\text{min}} = \Delta y_{\text{min}} = \Delta z_{\text{min}} = 1/160$ or 40 CPR. A constant time step size of $\Delta t = 0.16 \Delta x_{\text{min}}$ is used.
Fig. 3.13 shows snapshots of the heaving sphere and the free-surface evolution at two instances in dimensionless time $T = t\sqrt{g/R}$ for the 40 CPR simulation. Over time, the sphere generates ripples in the air-water interface that travel radially outward from the body. The finest mesh level surrounds both the immersed structure and the air-water interface; additional local regions of refinement are not formed because this particular case does not generate significant vorticity in the air phase. To quantitatively assess the accuracy and convergence of the wave-structure interaction, the vertical center of mass position of the sphere (nondimensionalized by $Z_0$) is plotted against time in Fig. 3.13. As the resolution increases, the numerical simulations converge towards the experimental...
results of Beck and Liapis [134]. Similar to the previous case, the sphere’s oscillation eventually damps out as it reaches an equilibrium position on the air-water interface.

3.8.4. Static cylinder on an air-water interface

In the previous cases, the structure underwent free-body motion. Therefore, we employed the full gravitational body force $\rho g$ in the computational domain. In this section, we demonstrate that using this same treatment of gravitational forcing for fully constrained motion produces parasitic currents, whereas using a gravitational forcing of the form $\rho_{\text{flow}} g$ eliminates spurious velocity currents in such cases (see Sec. 3.5 for discussion).

To begin, a circular cylinder of diameter $D = 1$ is placed within a 2D computational domain of size $\Omega = [0, 5D]^2$. Water occupies the bottom half of the domain and air occupies the remainder of the tank. The cylinder is placed at the center of the domain with initial center of mass $(X_0, Y_0) = (2.5D, 2.5D)$. In contrast with the previous cases, all of the cylinder’s translational and rotational degrees of freedom are locked and it is fully constrained to remain stationary, i.e. $U_b = (U_b, V_b) = (0, 0)$. As described in Sec. 3.5, the density and viscosity in the solid region are set to those of the water phase. Two grid cells of smearing $n_{\text{cells}} = 2$ are used to transition between different material properties on either side of the interfaces, and surface tension forces are neglected. No-slip boundary conditions are imposed along $\partial \Omega$. The initial problem set up is shown in Fig. 3.14(a).

We consider two forms of the gravitational body force:

1. the full gravitational forcing $\rho g$, with $\rho$ prescribed using Eq. (3.23).
2. the flow gravitational forcing $\rho_{\text{flow}} g$, with $\rho$ prescribed using Eq. (3.22).
The domain is discretized with a uniform $N \times N$ grid with $N = 200$ and a constant time step size of $\Delta t = 1/(5N)$ is used. For this particular case, no flow dynamics should be generated because the cylinder is held fixed in place and the initially quiescent fluids should maintain hydrostatic equilibrium. Hence, the quantity of interest is the $L^\infty$ norm of velocity $\|u\|_\infty$, which indicates the largest value (in magnitude) of parasitic velocity generated in the domain. Fig. 3.14(b) shows the temporal evolution of $\|u\|_\infty$ as a function of time for both cases. It is seen that the full gravitational forcing leads to nonzero velocity values that do not dissipate over time. This is because spurious momentum is accumulated in the solid phase due to its density value $\rho_s$ while solving the discretized momentum equations (3.29) and (3.30). However, the flow gravitational forcing produces velocities on the order of machine precision, indicating that hydrostatic equilibrium is maintained. Figs 3.14(c) and 3.14(d) show the velocity vectors and gravitational forcing for both cases at $t = 1.0$. Significant velocity is shown to be generated for the $\rho_g$ case, while these velocities are absent for the $\rho_{\text{flow}}g$ case. Based on these test results, for all fully prescribed motion cases considered in this chapter, we make use of the flow gravitational forcing field to ensure accurate and well-balanced results. Finally, we remark that the parasitic current generation due to the gravitational body force occurs for both types of flow solvers.

### 3.8.5. Water entry of a circular cylinder

In this section, we demonstrate the importance of consistent mass and momentum transport to achieve stability for air-water density ratios. A circular cylinder of radius $R =$
Figure 3.14. (a) Initial problem set up for a stationary cylinder (orange) placed on a quiescent air-water interface. (b) Temporal evolution of $\|u\|_{\infty}$ with the (●, red) full gravitational forcing, and the (■, blue) flow gravitational forcing. Velocity vectors and gravitational forcing for (c) $\rho g$, and (d) $\rho_{\text{flow}} g$ at $t = 1.0$. The scale for velocity vectors is identical in both figures.

0.055 is placed within a 2D computational domain of size $\Omega = [0, 40R] \times [0, 24R]$. Water occupies the bottom half of the domain from $y = 0$ to $y = 12R$ and air occupies the remainder of the tank. The cylinder is placed just above the fluid phase with initial center position $(X_0, Y_0) = (20R, 14R)$ and has density equal to that of the water phase.
(i.e. $\rho_s = 1 \times 10^3$). Two grid cells of smearing $n_{\text{cells}} = 2$ are used to transition between different material properties on either side of the interfaces, and surface tension forces are neglected. No-slip boundary conditions are imposed along $\partial\Omega$.

Similar to the previous case, all of the cylinder’s translational and rotational degrees of freedom are locked and its motion is fully constrained to be unity in the vertical direction, i.e. $\mathbf{U}_b = (U_b, V_b) = (0, -1)$. As described in Sec. 3.5, gravitational forces are not evaluated using the density within the solid region for prescribed motion cases. The primary quantity of interest for this example is the dimensionless vertical hydrodynamic force (slamming coefficient) given by

$$C_s = \frac{\mathbf{F} \cdot \mathbf{e}_y}{\rho_l RV_b^2}$$

as a function of the penetration depth $P_d = V_b(t - t_{\text{impact}})/R$, where $t_{\text{impact}} = R/V_b$ denotes the time at which the bottom of the cylinder first touches the air-water interface.

This case has been studied both analytically by von Kàrmàn [135], using potential flow theory, and experimentally by Campbell and Weynberg [136]. This case has also been studied numerically by a number of authors, including Patel and Natarajan [119], Zhang et al. [137], and Kleefsman et al. [138]. The domain is discretized by $\ell = 2$ grid levels with refinement ratio $n_{\text{ref}} = 2$. The grid spacing at the coarsest grid level is $\Delta x_0 = \Delta y_0 = 1/200$ yielding a finest grid spacing of $\Delta x_{\min} = \Delta y_{\min} = 1/400$ or 22 grid cells per radius. A constant time step size of $\Delta t = 0.02\Delta x_{\min}$ is used.

As a first test, we demonstrate the importance of consistent mass and momentum transport for the stability of high density ratio multiphase flows. Fig. 3.15 shows the evolution of the cylinder and the air-water interface when simulated with inconsistent mass
Figure 3.15. Temporal evolution of a cylinder entering an air-water interface at four different time instances. Inconsistent transport of mass and momentum is used for these cases. The simulation becomes unstable shortly after $t = 0.25$. The plotted vorticity is in the range $-50$ to $50$. Locations of the different refined mesh levels from coarsest to finest are shown in purple and green.

and momentum transport, i.e. by using a non-conservative momentum integrator. The simulation quickly becomes unstable, leading to the generation of unphysical interfaces and high regions of vorticity. Lowering the time step even further did not resolve these stability problems. However, the simulation is stable when using consistent mass and momentum transport (Fig. 3.16). As the cylinder enters the water, crashing wave-like structures are generated and air entrainment is seen around the cylinder.
Figure 3.16. Temporal evolution of a cylinder entering an air-water interface at four different time instances. Consistent transport of mass and momentum is used for these cases. The plotted vorticity is in the range $-50$ to $50$. Locations of the different refined mesh levels from coarsest to finest are shown in purple and green.

Fig. 3.17 shows the slamming coefficient as a function of penetration depth. The results are in decent agreement with previous studies, with minor disagreements being explained by differences in the interface tracking approaches and/or difference in the fluid-structure coupling techniques. This test case demonstrates how important consistent mass and momentum transport is for stable simulation of practical multiphase flows, for which air-water density ratios are ubiquitous. Moreover, we demonstrate that the present
Figure 3.17. Dimensionless slamming coefficient as a function of penetration depth for a 2D cylinder entering an air-water interface at constant velocity. (●, yellow) experimental data from Campbell and Weynberg [136]; (—, black) analytical formula from von Kàrmànn [135]; (---, green) simulation data from Patel and Natarajan [119]; (·–·, blue) simulation data from Zhang et al. [137]; (···, purple) simulation data from Kleefsman et al. [138]; (−−−, red) present simulation data.

A numerical method can be used to accurately simulate problems involving fully prescribed body motion.

3.8.6. Rolling barge

In this section, we investigate the roll decay of a rectangular barge floating on an air-water interface. A rectangle of length \( L = 0.3 \) and width \( W = L/3 \) is placed within a 2D computational domain of size \( \Omega = [-25W, 25W] \times [-9W, 16W] \). Water occupies the bottom portion of the domain up until \( y = 0 \) with air occupying the remaining
portion. The barge is initially situated with center of mass \((X_0, Y_0) = (0, 0)\) at a 15° angle of inclination with the horizontal, and has density \(\rho_b = 1.18 \times 10^3\). Two grid cells of smearing \(n_{\text{cells}} = 2\) are used to transition between different material properties on either side of the interfaces. Only the barge’s rotational degrees of freedom are unlocked and surface tension forces are neglected. The conservative and consistent flow solver is used for this case. No-slip boundary conditions are imposed along \(\partial \Omega\). This specific 2D case has been numerically studied by Patel and Natarajan [44].

The domain is discretized by a grid of size \(2N \times N\) with \(N = 1000\), yielding 40 grid cells per width. A constant time step size of \(dt = 1/(4N)\) is used. Fig. 3.18 shows the evolution of the rotating body and the air-water interface at various instances in dimensionless time \(T = t\sqrt{g/L}\). Vortices are shed from the corners of the barge as it rolls back and forth and small disturbances are seen along the water. The temporal evolution of the angle of inclination (Fig. 3.19) are in excellent agreement with the numerical results of Patel and Natarajan [44]. Over time, the roll angle damps out due to the viscous forces of the water phase. This case demonstrates that the present numerical method can be used to accurately simulate floating objects undergoing rigid, rotational motion.

3.8.7. Wedge free-falling into water

In this section, we investigate the problem of a wedge-shaped object impacting a pool of water. A 2D triangular body with top length \(L = 1.2\) is placed within a computational domain of size \(\Omega = [0, 10L] \times [0, 2.5L]\). The wedge is oriented with one of its vertices pointing downwards, making a 25° deadrise angle with the horizontal. Water occupies the bottom third of the domain, while air occupies the remainder of the tank. The bottom
point of the wedge is placed with initial position \((X_0, Y_0) = (5L, 23L/12)\) and the wedge has density \(\rho_s = 466.6\). Two grid cells of smearing \(n_{\text{cells}} = 2\) are used to transition between different material properties on either side of the interfaces, and surface tension forces are neglected. No-slip boundary conditions are imposed along \(\partial \Omega\). Only the wedge’s vertical degrees of freedom are unlocked.

For this 2D case, the domain is discretized by a \(4N \times N\) grid with \(N = 300\), and a constant time step size of \(\Delta t = 3/(160N)\) is used. We again demonstrate the importance of consistent mass and momentum transport by comparing inconsistent and consistent
formulations. Fig. 3.20 shows the evolution of the wedge and the air-water interface when simulated using the non-conservative momentum integrator. The simulation becomes unstable as the wedge impacts the water, leading to the generation of unphysical interfaces and high vorticity regions. Lowering the time step even further did not resolve these stability problems. However, the simulation remains stable when using consistent mass and momentum transport, achieved by the conservative momentum integrator. Fig. 3.21 shows physical interface deformation and reasonable vorticity generation from the vertices of the wedge.

We now turn our attention to the free-fall and water impact of a 3D wedge. The three-dimensional wedge geometry has a square top of size $L \times L$ with $L = 1.2$ and is placed
Figure 3.20. Temporal evolution of a 2D wedge free-falling into an air-water interface at four different time instances. Inconsistent transport of mass and momentum is used for these cases. The simulation becomes unstable shortly after \( t = 0.785 \). The plotted vorticity is in the range \(-300\) to \(300\).

within a computational domain of size \( \Omega = [0, 10L] \times [0, 5L/3] \times [0, 5L/2] \). The wedge is placed such that there is a gap of size \( 5L/12 \) between the vertical sides of the wedge and the lateral walls of the computational domain. The rest of the simulation parameters are analogous to those of the previous 2D case. Consistent mass and momentum transport is used.

In contrast with the two-dimensional case, we make heavy use of adaptive mesh refinement to selectively resolve regions of interest. The domain is discretized by \( \ell = 2 \) grid levels with refinement ratio \( n_{\text{ref}} = 2 \). The grid spacing at the coarsest level is \( \Delta x_0 = \Delta y_0 = \Delta z_0 = 1/25 \), yielding a finest grid spacing of \( \Delta x_{\text{min}} = \Delta y_{\text{min}} = \Delta z_{\text{min}} = 1/50 \).
Figure 3.21. Temporal evolution of a 2D wedge free-falling into an air-water interface at four different time instances. Consistent transport of mass and momentum is used for these cases. The plotted vorticity is in the range −300 to 300.

A constant time step size of $\Delta t = \Delta x_{\text{min}}/160$ is used. This 3D case has been studied both experimentally by Yettou et al. [139], and numerically by Calderer et al. [120] and Pathak and Raessi [51].

Fig 3.22 shows the locations of the different mesh levels at two snapshots in time. At the initial time, (Fig. 3.22(a)) the region surrounding the air-solid and air-water interfaces are covered by the finer mesh. As the wedge impacts the water (Fig. 3.22(b)) additional finer mesh regions are generated to resolve the vorticity generated in the air phase. Fig. 3.23 shows the evolution of the wedge at various instances in time. Upon impact, the wedge generates complex splashing and ripple phenomena along the water as
it bobs up and down. To quantitatively assess the accuracy of the wave-structure interaction, the vertical center of mass position and velocity of the wedge are plotted against time in Fig. 3.24. The results are in excellent agreement with the 3D simulation carried out by Pathak and Raessi [51] and experimental data compiled by Yettou et al. [139]. Finally, we note that while Calderer et al. [120] required \( n_{\text{cells}} = 8 \) smeared transition cells to obtain a stable simulation, we only use \( n_{\text{cells}} = 2 \). In fact in our formulation \( n_{\text{cells}} \) do not affect solver stability at all — only solver convergence properties are affected by this parameter. We get sharp flow structures and interfacial dynamics using \( n_{\text{cells}} = 1 \) or 2; larger smearing leads to diffuse interfaces and vortex dynamics around them. We attribute our simulation robustness to consistent mass and momentum transport, which is not being used in [120]. This case is representative of many real world WSI applications involving the interaction between heavy, buoyant objects interacting with air-water interfaces. The present numerical method can be robustly applied to these types of problems.

3.8.8. 3D water column impacting a stationary obstacle

In this section, we investigate the problem of a water column impacting a stationary box. A rectangular obstacle is placed in a 3D computational domain of size \( \Omega = [0, 3.22] \times [0, 1] \times [0, 1] \), which initially contains a rectangular water column at its far end; see Fig. 3.25 for a full description of the initial problem set up. The structure is held stationary and therefore gravitational forces are not evaluated using the density within the solid region (see Sec. 3.5). Two grid cells of smearing \( n_{\text{cells}} = 2 \) are used to transition between different material properties on either side of the interfaces, and surface tension forces are included.

\[^4\text{We also tried } n_{\text{cells}} = 8 \text{ with the non-conservative solver, but were unable to circumvent numerical instabilities.}\]
with coefficient equal to that of air-water: $\sigma = 0.0728$. No-slip boundary conditions are imposed along $\partial \Omega$. The domain is discretized by a $161 \times 50 \times 50$ grid with constant time step size $\Delta t = 1 \times 10^{-4}$. This problem has been studied experimentally at the Maritime Research Institute Netherlands (MARIN), and numerically by Kleefman et al. [138] and Pathak and Raessi [51].

Fig. 3.26 shows the evolution of the water column at various instances in time. Under the effects of gravity, the column spreads across the domain’s lower boundary, eventually crashing into the rectangular structure. The structure obstructs the fluid flow, causing the water to divert upwards and over the body. Small ripples are also generated in the
Figure 3.23. Temporal evolution of a 3D wedge free-falling into an air-water interface at four different time instances.

...bulk flow. The water eventually hits the left side of the computational domain, causing a flow reversal and secondary crash over the body.
The primary quantities of interest for this example are the pressure values collected from four probes placed on the surface of the obstacle and the water height collected from two probes placed along the domain (Fig. 3.25). The coordinates of the pressure and height probes are shown in Table 3.2. Note that the pressure probes are located at a specific spatial location in the computational domain, whereas the water height probes are lines extending upward from a given 2D coordinate in the \( xy \)-plane. The temporal evolution of pressure and water height at these probes are shown in Figs. 3.27 and 3.28. The results are in decent agreement with the experimental data and the simulations carried out in [51, 138], with minor disagreements being explained by differences in the interface tracking approaches and/or variations in post-processing pressure and water height data.
Figure 3.25. Sketches of the initial problem set up for a water column impacting a stationary rectangular object: (a) 3D view specifying the size of the domain and the locations of the pressure sensors P1, P3, P5, and P7 (×, black) on the surface of the body; (b) top view and (c) side view indicating the dimensions and locations of the water column (blue) and the obstacle (orange), along with the locations of the water height probes H2 and H4.
Figure 3.26. Temporal evolution of a water column impacting a rectangular object at eight different time instances.
Table 3.2. Locations of the pressure and water height probes for the 3D water column impacting a stationary obstacle. Pressure probe locations are given by 3D coordinates while water height probe locations are given by 2D coordinates in the $xy$-plane.

<table>
<thead>
<tr>
<th>Probe</th>
<th>Measurement</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Pressure</td>
<td>(0.82, 0.475, 0.02)</td>
</tr>
<tr>
<td>P3</td>
<td>Pressure</td>
<td>(0.82, 0.475, 0.1)</td>
</tr>
<tr>
<td>P5</td>
<td>Pressure</td>
<td>(0.84, 0.525, 0.16)</td>
</tr>
<tr>
<td>P7</td>
<td>Pressure</td>
<td>(0.92, 0.525, 0.16)</td>
</tr>
<tr>
<td>H2</td>
<td>Water height</td>
<td>(0.992, 0.5)</td>
</tr>
<tr>
<td>H4</td>
<td>Water height</td>
<td>(2.638, 0.5)</td>
</tr>
</tbody>
</table>

With this particular case, we have demonstrated that complex surface tension and gravity driven splashing dynamics are accurately simulated by the present numerical method.

3.9. Conclusions

In this chapter, we coupled the robust multiphase flow solver of Chapter 2 with the DLM-based immersed boundary method of Bhalla et al. [13] to enable fast simulations of high density ratio wave-structure interaction problems. We demonstrated that our method is applicable to a wide range of WSI problems involving air-water interfaces, and can adequately resolve complex wave and splashing dynamics. We were able to achieve substantially reduced computational costs by making use of adaptive mesh refinement to capture important flow features. Various types of rigid body dynamics are modeled, including prescribed, free-translational and free-rotational motion.

For fully-constrained motion, we showed that the “virtual” density within the body domain can produce spurious velocities inside the structure; this erroneous momentum

---

5We relied on the VisIt [140] software to extract the probe data from the parallel HDF5 files.
eventually contaminates the flow field and can yield inaccurate results. To mitigate these parasitic currents, we described a well-balanced formulation of the gravitational body
force based on the density of the “flowing” phases. Additionally, we demonstrated the importance of consistent mass and momentum transport to eliminate numerical instabilities for high density ratio flows, which have long plagued the multiphase flow community.

We also presented a level set method based numerical wave tank implementation. Second-order Stokes waves were generated by using inlet velocity boundary conditions. Wave reflection and wave interference effects were mitigated by the use of a wave damping zone. Although not shown in this chapter, we also generated waves using a relaxation procedure instead of inlet velocity boundary conditions, but found the results to be very similar. Generation of more complex waves including fifth order Stokes \[141\], cnoidal \[142\], focused \[143\], and random waves based upon sea and ocean spectra \[144\] are already underway. Moreover, all of the code development is open-source.

Figure 3.28. Temporal evolution of water height measured at probes (a) H2, and (b) H4 for a 3D water column impacting a stationary obstacle (see Fig. 3.25 and Table 3.2); (●, yellow) experimental data from MARIN; (—·—, red) simulation data from Kleefsman et al. \[138\]; (---, blue) simulation data from Pathak and Raessi \[51\]; (—, black) present simulation data.
Because we presented results for relatively simple geometries, constructive solid geometry concepts to compute the signed distance function to the surface of the immersed body sufficed. Our code also has the ability to compute the signed distance functions from CAD and STL files directly. Moreover, the implementation can also work with a finite element representation of the immersed body instead of unconnected Lagrangian markers. These extensions allow for complex geometries (such as that of WECs) to be represented on Cartesian grids. Further, the solution methodology can be augmented with a RANS or LES turbulence model [96, 97, 98], which would enable simulations of many important industrial and engineering applications such as high inertia vehicles, wave-energy converter devices, and windmills.
CHAPTER 4

Optimal specific wavelength for maximum thrust production in undulatory propulsion

What wavelengths do undulatory swimmers use during propulsion? In this chapter we find that a wide range of body/caudal fin (BCF) swimmers, from larval zebrafish and herring to fully–grown eels, use specific wavelength (ratio of wavelength to tail amplitude of undulation) values that fall within a relatively narrow range. The possible emergence of this constraint is interrogated using numerical simulations of fluid–structure interaction. Based on these, it was found that there is an optimal specific wavelength (OSW) that maximizes the swimming speed and thrust generated by an undulatory swimmer. The observed values of specific wavelength for BCF animals are relatively close to this OSW. The mechanisms underlying the maximum propulsive thrust for BCF swimmers are quantified and are found to be consistent with the mechanisms hypothesized in prior work. The adherence to an optimal value of specific wavelength in most natural hydrodynamic propulsors gives rise to empirical design criteria for man–made propulsors.

4.1. Introduction

Dimensionless quantities are used to directly compare the biomechanics between systems of different scales [145]. The Strouhal number (St = 2fa/U, where f is the tail or wing beating frequency, 2a is the maximum tip–to–tip lateral excursion, and U is the forward speed) is a nondimensional number used to describe the kinematics of flying
and swimming animals [146, 147]. Although it has been shown that some species of body/caudal fin (BCF) swimmers cruise at $0.2 < St < 0.4$, there is evidence to support that the Strouhal number varies over a larger range of $0.6 – 2.2$ for anguilliform and low Reynolds number undulatory swimmers [148, 149]. It is suggested that natural selection drives animals to this range of Strouhal numbers because it maximizes propulsive efficiency (ratio of hydrodynamic power output to mechanical power input) during cruising [146, 150]. Another dimensionless number called the specific wavelength (SW), the ratio of wavelength $\lambda$ to an amplitude length scale, has been used to describe the kinematics of elongated fin (EF) swimmers [151]. Furthermore, SW has been shown through simulations and experiments to maximize net propulsive force for EF swimmers when it achieves a value near 20. This has been referred to as the optimal specific wavelength or OSW [151].

EF swimmers are characterized by a flexible elongated fin that runs lengthwise with the rigid body, allowing greater maneuverability but at slow speeds. The amplitude of the undulations vary along the length and span of the fin. On the other hand, BCF fish undulate their bodies and caudal fin to produce greater thrust and accelerations [145, 152]. The amplitude for BCF swimmers generally varies along the length, but not the span of the swimmer (although the amplitude can vary along the span of the caudal fin [153]). Additionally, the tail amplitude $a$ is crucial in producing propulsive thrust [154], and therefore the relevant specific wavelength for BCF swimmers is $SW = \lambda/a$. It seems likely that BCF swimmers would benefit from adherence to OSW because it ensures that speed can be maximized given a set of kinematic parameters irrespective of whether the animal swims efficiently during cruising or less efficiently during attack or evasive maneuvers. In
this chapter, we show that undulatory BCF swimmers also abide to a relatively narrow range of SW. For this type of propulsion, the observed steady-swimming kinematics are found to optimize net axial force over a wide range of length and velocity scales.

In this work, a meta-analysis was done using data from 27 species of undulatory BCF swimmers to find typical Strouhal numbers and specific wavelengths observed in these animals. Previous analyses of these data showed that the observed Strouhal number depends nonlinearly on the Reynolds number (\( \text{Re} = \rho UL/\mu \), where \( U \) is the fish’s forward swimming speed, \( L \) is the fish’s length, and \( \rho \) and \( \mu \) are the density and viscosity of water, respectively) \[ ^{148} \]. Reanalyzing these data, we found that swimming mode and aspect ratio (AR) play a role in the variation of St. Additionally, a numerical study done in this work is used to establish an optimality condition relating net axial force and specific wavelength of BCF swimmers. The meta-analysis and simulation data were compared to OSW measurements obtained from parametric studies done on robotic undulating sheets \[ ^{155, 156} \]. Based on these analyses, we find reasonable adherence by BCF swimmers to the OSW, and a mechanical reason for the possible emergence of this phenomenon.

4.2. Results

4.2.1. Meta-analysis

We first investigated whether or not BCF animals swim near a constant specific wavelength value. Since the most relevant amplitude length scale for body/caudal fin swimmers is the tail amplitude \[ ^{154} \], the specific wavelength is defined to be \( \lambda/a \). Specific wavelength was calculated for 27 species (28 groups since distinction is made between young
234

and adult axolotl) of steady–swimming BCF swimmers based on data from a meta–analysis study [148]. These observations cover a wide range of lengths (0.3 cm for larval zebrafish to 69.5 cm for Atlantic cod), velocities (0.249 $L/s$ for longnose gar to 54.0 $L/s$ for larval Atlantic herring), and swimming modes (anguilliform to thunniform [152]). These data also include observations from adult and larval Mexican salamanders, which swim with an anguilliform motion [157, 158]. The lateral Reynolds number ($\text{Re}_{\text{lat}} = \rho (fa)a/\mu$, where $a$ and $f$ are the undulation tail amplitude and frequency, respectively) was calculated for all of these observations and ranged from $2.55 \times 10^1$ to $8.55 \times 10^3$. The corresponding range in swimming–speed based Reynolds number was $2.10 \times 10^2$ to $7.71 \times 10^5$.

For the swimming animals considered in this study from [148], the relationship between $\text{Re}$ and $\text{Re}_{\text{lat}}$ is shown in section 4.4.3. We chose to analyze this data using $\text{Re}_{\text{lat}}$ instead of $\text{Re}$ to allow for a direct comparison to our translation–locked fin simulations, for which there is no measured swimming speed (see Parametric Study). Additionally for free–swimming simulations, $\text{Re}_{\text{lat}}$ is a parameter that is known a priori and can be prescribed at the beginning of the simulation. On the other hand, $\text{Re}$ is an output of the simulation, which is not preferable when conducting parametric studies.

For the elongated fin swimmers considered in [151], it was found that the ratio of wavelength to average amplitude $\tilde{a}$ was near 20 for a wide range of aquatic invertebrates and vertebrates. Since the amplitude $a$ at the tail tip was chosen to define specific wavelength in this study and we observed that $a/\tilde{a}$ was between 1.53 and 2.56 for these body/caudal fin swimmers, we chose to interrogate whether or not these animals undulate at SW values near 10. Letting $x_{ij}$ denote observation $j$ from group $i$, we calculated mean quantities for group $i$ with $N_i$ observations as $\bar{x}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{ij}$, and overall averages $\bar{x} =$
\[
\frac{1}{28} \sum_{i=1}^{28} x_i.
\]
We also calculated the overall standard deviations as 
\[
s_x = \sqrt{\frac{1}{27} \sum_{i=1}^{28} (x_i - \bar{x})^2}.
\]
We found that these fish species swim at specific wavelengths between 4.02 and 14.93 with a mean value of \( \bar{SW} = 9.91 \) and standard deviation \( s_{SW} = 3.45 \). We performed a two-sided \( t \)-test on the average specific wavelength \( \bar{SW}_i \) for the \( n = 28 \) groups of swimmers. Although the data are not normally distributed, we chose the \( t \)-test because of its robustness to large deviations from the normality assumption. For an \( \alpha \)-level of 0.05, we found that the average \( \bar{SW} = 9.91 \) did not vary significantly (\( P = 0.888 \)) from 10, with a 95% confidence interval (CI) of 8.57 < \( \bar{SW}_i \) < 11.25. The median \( \bar{SW}_i \) is 10.30.

Fig 4.1 shows the mean St and SW for each of these 28 groups. It is seen that the range of Strouhal numbers depends on the mode of swimming; St is clustered around 0.3 for non-anguilliform (subcarangiform, carangiform, or thunniform) swimmers while St consistently overshoots 0.3 for anguilliform swimmers. The mean Strouhal number for non-anguilliform swimmers is 0.30, while for anguilliform swimmers it is 0.66. For anguilliform swimmers, this average varied significantly from 0.3 (\( n = 11, P = 0.01, 95\% \) CI: 0.41 < \( St_i < 0.90 \)).

It is seen that the observed variation in SW is no more than that in St; SW did not vary as much as St with respect to different swimming modes. Are there traits common to these anguilliform swimmers that correlate to higher Strouhal number locomotion? To answer this, we plotted St vs. Re_{lat} and St vs. AR in Fig 4.2 for both anguilliform and non-anguilliform swimmers. We found that non-anguilliform swimmers were clustered around \( St = 0.3 \) and their aspect ratios tend to be higher (\( AR > 0.133; \) Fig 4.2b). Here the aspect ratio is \( AR = (\text{area of fish body})/(\text{length of fish body})^2 \). Anguilliform swimmers with similar lateral Reynolds numbers as non-anguilliform swimmers have in
Figure 4.1. Observed St and SW ranges from meta–analysis data of body/caudal fin swimmers. The Strouhal number (top) and specific wavelength (bottom) for the species considered in this study. Data points represent average values, $\bar{x}_i$ where $x \in \{St, SW\}$, of individual species and error bars indicate ± one s.d. from the species mean. Dashed lines represent St = 0.3 and SW = 10. Distinction is made between anguilliform and non–anguilliform swimmers. Data labeled as *Ambystoma mexicanum* and *Ambystoma mexicanum young* are from adult and larval axolotl, respectively. All observations of *Clupea harengus* and *Danio rerio* came from anguilliform larvae. For some species, the error bars are not visible at this scale, while for others, only one observation is recorded and no error bars are available; see Table 4.1 for more details.
general higher values of St. This trend may be due to the low aspect ratios of anguilliform swimmers, which tend to be longer and skinnier and swim at higher St values (Fig 4.2b). Additionally, anguilliform swimmers at low Reynolds numbers tend to have even higher values of St as seen in previous studies [148, 159]. For example, notice that the two larval anguilliform species that swim at $Re_{lat} < 10^2$ (Danio rerio and Clupea harengus) also swim at $St > 1$. Thus, we hypothesize that low aspect ratios cause higher St for anguilliform swimmers and low Re swimming tends to further increase the observed values of St in anguilliform swimmers.
Figure 4.3. Variability in SW and OSW as a function of $Re_{\text{lat}}$ and AR. Intraspecies mean specific wavelength vs. (a) lateral Reynolds number, (b) aspect ratio for observed non-anguilliform (●) and anguilliform (▲) swimmers. Of these, orange (green) points represent swimmers with (without) well–defined caudal fins. Red crosses (X) represent the optimal specific wavelength for simulations done in the present study. Blue asterisks (*) represent the optimal specific wavelengths for robotic undulating sheets reported in [155, 156]. Error bars indicate ± one s.d. from the species mean. For some species, the error bars are not visible at this scale, while for others, only one observation is recorded and no error bars are available; see Table 4.1 for more details.

Finally Fig 4.3 shows SW as a function of $Re_{\text{lat}}$ and AR for these species. We did not see a variation pattern in SW due to either of these parameters as was seen with St. Body/caudal fin swimmers were found to adhere to a similar SW constraint as elongated fin swimmers [151]. A possible physical basis of this constraint in BCF swimmers is explored in the next section using numerical simulations.

4.2.2. Numerical simulations
4.2.2.1. Nondimensionalization. Motivation for the present numerical study came from dimensional analysis. Consider a rectangular plate immersed in a fluid with a prescribed traveling wave undulation pattern given by,

\[
y(x, t) = a \sin \left( \frac{2\pi}{\lambda} (x - \lambda ft) \right),
\]

in which \( f \) is the frequency of the traveling wave, \( \lambda \) is the wavelength, and \( a \) is the amplitude of the undulation, which is taken to be constant. Assuming that a steady periodic flow state has been reached and that the translational degrees of freedom of the sheet have been locked, the net static axial propulsive force \( F_x \) depends on the following physical parameters:

\[
F_x = fn(\rho, \mu, f, \lambda, L, h, a),
\]

in which \( fn \) denotes “function of,” \( \rho \) is the fluid density, \( \mu \) is the fluid viscosity, and \( L \) and \( h \) are the length and span of the plate, respectively. Note that since there is no far–field velocity opposing the plate and the plate is locked in place, there is no velocity present in Eq. \( \text{4.2} \).

Using the Buckingham Pi Theorem, Eq. \( \text{4.2} \) can be nondimensionalized as,

\[
\tilde{F} = \frac{F_x}{\frac{1}{2} \rho (fa)^2 Lh} = fn \left( \frac{\lambda}{a}, \frac{\rho (fa)a}{\mu}, \frac{h}{L}, \frac{h}{a} \right) = fn \left( \text{SW}, \text{Re}_{\text{lat}}, \text{AR}, \frac{h}{a} \right).
\]

Here, the dimensionless force \( \tilde{F} \) depends on specific wavelength \( \text{SW} \), lateral Reynolds number \( \text{Re}_{\text{lat}} \), plate aspect ratio \( \text{AR} = Lh/L^2 = h/L \), and a scaled height parameter \( h/a \). Reynolds numbers based on the wave–speed \( \text{Re}_{\text{wave}} = \rho (\lambda f)L/\mu \) or lateral speed
and length $\text{Re}_{\text{lat,}L} = \rho(fa)L/\mu$ would have been equally valid choices in our nondimensionalization. We chose to include the lateral Reynolds number in Eq. 4.3 because there was no swimming speed associated with Eqs. 4.2 and 4.3, and this choice separates the dimensionless quantities depending on frequency and wavelength. For the swimming animals considered in this study, all of these Reynolds numbers are strongly correlated with each other and the swimming–speed Reynolds number $\text{Re}$ (see section 4.4.3). Hence, our analysis is independent of the chosen Reynolds number. In the present numerical study, simulations gave insight on what range of specific wavelength maximizes net propulsive force. We also explored how $\text{AR}$, $\text{Re}_{\text{lat}}$, and $h/a$ affected the optimal specific wavelength (OSW).

4.2.2.2. Parametric study. Using numerical simulations of small sheets and scaled-down eel and mackerel bodies, we interrogated whether an optimization principle exists between net axial force and specific wavelength for highly undulatory bodies.

Two types of simulations were carried out in this study: translation–locked and free–swimming. For translation–locked simulations, let $F = (F_x, F_y, F_z)$ be the transient net force felt on the swimmer in the $x$, $y$, and $z$ directions and for free–swimming simulations, let $U = (U_x, U_y, U_z)$ be the center–of–mass (COM) velocity of the swimmer. In the translation–locked simulations, the undulating body was fixed in place and $F$ was measured while in the free–swimming case, the body’s COM velocity $U$ was solved for by the simulation and the body self–propelled. In both cases, the rotational degrees of freedom of the swimmer were locked. The simulations carried out in this study were “fully–resolved”, meaning that the fluid–solid coupling was not modeled using drag laws; the flow field, net propulsive force, swimming speed, and power generated by the swimmer
were outputs of each simulation. Moreover, all the dynamics of the swimming system were captured in our simulations, including the effect of linear recoil. For translation–locked simulations, the recoil effect led to oscillations in the sway \( F_y \) and heave \( F_z \) forces. For free–swimming simulations, the recoil effect led to an oscillation of the swimmer’s center of mass in the sway \( U_y \) and heave \( U_z \) directions, which were outputs of the simulation. Both axial swimming speed and net axial force oscillate about a mean value in either case.

Fig 4.4a shows the correlation between propulsive thrust \( F_x \) and swimming velocity \( U_x \) by comparing the results from the stationary and free–swimming plate simulations. We indeed see that both swimming speed and thrust are maximized by similar SW values. Additionally, \( F_x \) and \( U_x \) are strongly correlated for SW < 20 and their peaks occur at similar OSW values. For higher SW, their correlation is weaker, although both \( F_x \) and \( U_x \) decrease with increasing SW in this range. We note that all the observed animals in the meta–analysis swam at SW between 4 and 15. The transient behaviors of \( F \) and \( U \) are shown in Figs 4.4b and 4.4c. More details of the numerical simulation technique and parameters are provided in the Sec. 4.4. Hereafter, we focus on analyzing propulsive force from stationary simulations, although we expect that the swimming speed computed from self–propelling simulations would follow similar trends with respect to SW.

We carried out simulations of stationary rectangular small–sheet (0.5 – 8 × 0.1 – 1.2 cm²) and large–sheet (10 – 20 × 1 – 2 cm²) sinusoidally undulating bodies with various wavelengths (0.25 – 40 cm), amplitudes (0.05 – 1.0 cm), and frequencies (1 – 4 Hz). For fixed length, span, frequency, and amplitude, the wavelength was varied and the OSW was taken to be the value of \( \lambda/a \) that maximized the force in the axial direction. The length and lateral speed scales for our simulated sheets were chosen to match those of swimmers.
Figure 4.4. Measured swimming speed and force from undulating sheet simulations. (a) Axial swimming speed and propulsive force computed from free–swimming (green) and translation–locked (black) simulations of rectangular sheets plotted against the specific wavelength. In free–swimming simulations, the forward swimming speed of the undulating plate was an output parameter of the simulation. Simulations were carried out at a lateral Reynolds number $R_{\text{lat}} = 4.49$, with corresponding swimming–speed Reynolds number range $1.8 \times 10^4 < \text{Re} < 1.51 \times 10^2$. (b) Evolution of axial $F_x$ and heave forces $F_z$ over time for a translation–locked, undulating sheet simulation with SW = 13.33. The oscillation in $F_z$ is a signature of the linear recoil effect on the swimmer. The sway force $F_y$ is not shown because the kinematics of the swimmer’s undulation lead to large $F_y$ values, although it also oscillates about a mean value. (c) Evolution of $U$ in each coordinate direction over time for a self–propelled, undulating sheet simulation with SW = 13.33. The oscillation of the heave velocity $U_z$ about 0 is not easily visible at this scale.
with available kinematic data. Small-sheet simulations had $\text{Re}_{\text{lat}} \sim 10^{-1} - 10^2$ similar to species like larval *Clupea harengus* and *Danio rerio*, while large-sheet simulations had $\text{Re}_{\text{lat}} \sim 10^2 - 3 \times 10^2$ similar to species like *Anguilla anguilla* and *Micropterus salmoides*. Since there was no swimming-speed associated with these simulations, $\text{Re} = 0$. The relationship between $\text{Re}$ and $\text{Re}_{\text{lat}}$ was calculated for swimming animals in [148] and is shown in section 4.4.3. We used aspect ratios ranging from 0.05 to 0.6, which is the AR range found for the BCF swimmers considered in this study.

Fig 4.5 shows the results of simulations done on sheets with $\text{Re}_{\text{lat}} = 8.43 \times 10^{-1}$ and $\text{Re}_{\text{lat}} = 4.49$. In Figs 4.5a & 4.5b, the optimal specific wavelength increases from 10 to 20 as the plate aspect ratio increases from 0.1 to 0.4 and 0.05 to 0.6 respectively. Fig 4.6 shows the result from simulations carried out on sheets with $\text{Re}_{\text{lat}} = 3.37 \times 10^2$. For these higher Reynolds number sheets, the OSW is nearly independent of AR. This indicates that there is a parametric dependence of OSW on AR for low $\text{Re}_{\text{lat}}$ regimes that does not persist for moderate $\text{Re}_{\text{lat}}$ swimmers. Due to computational limitations, we were unable to interrogate whether or not this aspect ratio independence continues for even higher Reynolds numbers. However for moderate Reynolds numbers, this is consistent with what is observed in the meta-analysis data: both low AR anguilliform and high AR non-anguilliform swimmers undulated at SW near 10. To our knowledge, there is no experimental observation on BCF fish with both $\text{Re}_{\text{lat}} < 10^2$ and AR > 0.15.

Figs 4.7 & 4.8 show additional small-sheet simulations in which aspect ratio, $\text{Re}_{\text{lat}}$, and amplitude were varied systematically. These show a consistent dependence of OSW on AR for low $\text{Re}_{\text{lat}}$ and that OSW does not vary considerably as a function of frequency or amplitude. In particular, Fig 4.7b shows that the OSW does not vary considerably as
Figure 4.5. Measured propulsive force from low Re\textsubscript{lat} undulating sheet simulations. The axial propulsive force generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate span was varied. These data represent cases where Re\textsubscript{lat} < 1 \times 10^2.

a function of \( h/a \) for fixed AR. The range of \( h/a \) values considered match those of the observed swimmers in this study (1.15 – 8.22).

Fig 4.9 shows small–sheet simulations in which the amplitude profile \( A(x) \) was allowed to vary along its body length. Two different profiles were considered; \( A(x) = ae^{x/L-1} \), which represents a modeled anguilliform swimmer, and \( A(x) = a(0.2 - 0.8x/L + 1.6(x/L)^2) \), which represents a modeled carangiform swimmer [121, 160]. These simulations were carried out at Re\textsubscript{lat} = 4.49. We again see that propulsive force is maximized at
Figure 4.6. Measured propulsive force from high Re_{lat} undulating sheet simulations. The axial propulsive force generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate length was varied. These represent cases where Re_{lat} > 1 \times 10^2.

SW = 10 in both cases, implying that the OSW criterion is valid for variable amplitude sheet swimmers.

Finally, Figs 4.10a and 4.10b show both stationary and free-swimming simulations done on realistic eel and mackerel swimmer geometries, respectively. Although these bodies were realistically shaped, they were scaled-down in size due to the limitations of our computational tool. For the eel body, an anguilliform amplitude profile was prescribed,
while for the mackerel body a carangiform amplitude profile was prescribed. These simulations were carried out at $Re_{lat} = 1.12 \times 10^2$, with corresponding swimming-speed Reynolds number range $1.378 \times 10^3 < Re < 5.95 \times 10^3$. In all cases, there is an optimal specific wavelength that maximizes axial force and swimming speed. This implies that the OSW design principle holds for more realistic swimming bodies. Visualizations of the body geometries and simulations are provided in Sec. 4.4.
Figure 4.8. Measured propulsive force from low Re$_{lat}$ sheets with varying span and length. Axial propulsive force vs. specific wavelength for additional small–sheet simulations with varying aspect ratios. In case (a), plate span was varied while in case (b), plate length was varied.

The axial force simulation data shown in Figs. 4.5, 4.6, 4.7, 4.8, & 4.9 were non–dimensionalized and are presented in Figs. 4.15–4.19.

4.2.3. Robotic sheets

In [155] and [156] the authors performed a parametric study on robotic undulating plates with constant amplitude along their length and span. Although these fins were actuated from a base, these experiments resemble the simulations done on undulating rectangular
sheets in this work insofar as their amplitude did not vary along their span. In these experiments, four identical free-swimming robotic plates were fully submerged in water and undulated at \( f = \frac{6}{9}, \frac{7}{9}, \frac{8}{9}, \) and \( \frac{9}{9} \) Hz; for a given frequency, wavelength was varied in order to find the optimal \( \lambda \) that maximized swimming speed \( U \). The plates had length \( L = 63 \) cm and span \( h = 20 \) cm, corresponding to an aspect ratio of \( \text{AR} = 0.32 \). The sheet had amplitude \( a = 5 \) cm and all of the experiments were carried out at \( \text{Re}_{\text{lat}} \in [1.87 \times 10^3, 2.50 \times 10^3] \), with corresponding swimming-speed Reynolds number range \( \text{Re} \in [5.96 \times 10^4, 1.53 \times 10^5] \).

The OSWs for these four cases are shown in Fig 4.3 alongside the OSWs from the simulations done in this study. The optimal specific wavelengths for these robotic sheets were between 10 and 13, which are consistent with simulation and animal data. Furthermore, the \( \text{Re}_{\text{lat}} \) values for these robotic sheets were higher than those used in our
249

Figure 4.10. Measured swimming speed and force from realistic eel and mackerel simulations. Axial swimming speed and propulsive force computed from free-swimming (green) and translation–locked (black) simulations of undulating (a) eel bodies, and (b) mackerel bodies, plotted against the specific wavelength. In free–swimming simulations, the forward swimming speed of the undulating body was an output parameter of the simulation. These simulations were carried out at Re\textsubscript{lat} = 1.12 \times 10^2, with corresponding swimming–speed Reynolds number range 1.378 \times 10^3 < Re < 5.95 \times 10^3.

numerical simulations. This is evidence that SW maximizes swimming speed for high Reynolds number animals.

4.3. Discussion

Through analysis of undulatory swimmer data, we have shown that BCF swimmers undulate at a relatively narrow range of specific wavelength values. The range of specific wavelength values observed in 27 species of steady–swimming BCF swimmers was 4.02 – 14.93 with SW = 9.91 and s\textsubscript{SW} = 3.45. Additionally we observed, across a large range of Reynolds numbers, that low aspect ratio anguilliform swimmers tend to swim at
greater St than high aspect ratio non–anguilliform swimmers. This is consistent with numerical studies relating the emergence of single and double–row wakes to optimal Strouhal numbers for carangiform and anguilliform swimmers, respectively [121, 160].

Using numerical simulations and data from robotic undulating sheets, we showed that the optimal specific wavelength for maximal thrust generation falls in the range 5–30 for undulating bodies, which closely matches the range of SW values found in observations of BCF fish. However, the morphology of the undulating sheet seemed to cause variation in the location of the OSW. Our undulating sheet simulations were a simplified model problem for actual BCF swimmers. These simulations did not capture the effect of varying span of a swimmer, which is a common trait of BCF swimmers, nor the effect of a distinct caudal fin. However, they provide a physical basis for why these swimmers might undulate within a range of SW values: the larger thrust and swimming speed attained at the OSW are advantageous. Future studies should carry out simulations with more realistic gaits in order to quantify how the OSW changes for more realistic swimming bodies. Similarly, robotic sheet data corroborate the existence of an OSW for undulatory swimmers at high Reynolds numbers, but parametric studies on more realistic BCF robots should be conducted.

To the best of our knowledge, propulsive wavelength of body/caudal fin swimmers is rarely measured in experimental studies. This work establishes the importance of \( \lambda \) as a kinematic parameter for propulsive performance. All of the organisms considered in this study, except for the axolotl, are body/caudal fin fish, and therefore have the same ancestry. The hope is that this work inspires more data on wavelength to be measured.
across a wider variety of swimming animals to determine whether this SW trait is universal among aquatic locomotors.

The optimal specific wavelength rule represents a small subset of a larger and complicated design space with various optimality principles relating the input parameters shown in Eq. 4.2 to other cost of transport or efficiency metrics. We do not claim that the SW constraint is more important than any other optimality principle; rather, the entire landscape needs to be explored in order to determine the most optimal swimmer. However, swimming at the OSW to maximize $U_x$ can be done in both low and high efficiency swimming situations. For example, consider a swimmer of fixed dimensions required to cruise a long distance at a low cost of transport. In this case, slow and steady swimming at a low frequency and amplitude is desirable. However, swimming at the OSW will maximize speed within this high efficiency regime. Conversely in the case where a swimmer needs to escape from a predator, the efficiency at which it swims is less important as it undulates its body with the highest frequency and amplitude possible. Once these maximum $f$ and $a$ values are achieved, swimming at the OSW would further maximize speed within this low efficiency regime. The situation can be reversed as well: for a fixed swimming body required to swim at some desired speed $U$, a SW value can be chosen to achieve this speed while the remaining parameters can be chosen to maintain this speed at a maximum efficiency.

When considering the kinematics of underwater vehicles, the specific wavelength is a quantity that can be prescribed before swimming starts while the Strouhal number is unknown until the vehicle has reached a steady-swimming speed. This is advantageous from a design standpoint as net axial force (or swimming speed) can be predicted and
maximized prior to building a vehicle. However, our numerical simulations establish an
OSW range for maximized thrust without considering the power spent to attain that
propulsive force. Moreover, the Strouhal number has been shown to maximize propulsive
efficiency, the ratio of net thrust times a desired speed to power spent [150]. Given a
set of mechanical constraints on a vehicle (an immersed body of fixed $L, h, \text{ and } a$ that
must swim at $U$), SW and St could provide an optimal choice of $\lambda$ and $f$. Therefore,
the Strouhal number and specific wavelength form a pair of complementary design rules.
This landscape relating propulsive efficiency and thrust to SW and St will be the subject
of future investigation.

Additional constraints on the kinematics of undulatory swimmers must be met when
considering design criteria for underwater vehicles. It has been shown through direct
numerical simulation of turbulent flow over an undulating wall that the wave–speed $V_f =
\lambda f$ must exceed the external flow velocity $U_0$ for thrust to be produced and turbulence
to be reduced [161]. For free–swimming bodies, this forces the swimming speed $U$ to be
less than $V_f$, which is observed for our free–swimming simulation and all the swimmers
considered in this study.

Our simulations also showed that the OSW increases from 10 to 20 as the undulating
plate’s aspect ratio increased from 0.05 to 0.6 when $\text{Re}_{\text{lat}} < 100$. This implies that a
skinny swimmer needs more spatial undulations along its body to maximize thrust than
a wide swimmer in the viscous regime. We hypothesize that the mechanisms described in
out prior work [151] might explain why this occurs. In section 4.4.5 we mathematically
describe and discuss the plausibility of the two competing mechanisms. A skinny plate
with small span $h$ (low aspect ratio) has a disadvantage in transporting fluid efficiently
when compared to a wide, high aspect ratio plate. Therefore, the low AR plate requires more waves along its body (a lower SW) to generate maximal thrust. At higher values of Re_{lat}, we hypothesize that the increased length scale diminishes the importance of varying span. This parametric dependence of the OSW on aspect ratio and Reynolds number has yet to be observed for free–swimming bodies. However, consider a free–swimming fish that is undulating its body at an optimal specific wavelength value that maximizes its swimming speed, keeping all other independent parameters fixed. Now imagine this swimmer’s translational degrees of freedom are locked and the fish is constrained to remain stationary while still undulating its body with the same free–swimming deformation kinematics. We hypothesize that the net axial force generated by this stationary fish would be maximum at similar free–swimming optimal specific wavelength value, keeping all other independent parameters fixed. Simulations carried for an undulating sheets, mackerels, and eels corroborate this hypothesis, although there does seem to be some variation in OSW values between the stationary and free–swimming cases.

The three–dimensional wake structures from undulating bodies has been analyzed as a function of Strouhal number in previous simulation studies [121, 160, 162]. Wake visualizations at three different SW values from the free–swimming, realistic eel and mackerel simulations conducted in this study are shown in Fig. 4.11. The speed at which the top and bottom row wakes push away from each other seems to increase as SW increases. Additionally, it appears that the vortical structures shed from each swimmer are more organized and remain more coherent at the SW value maximizing swimming speed. Therefore, it seems likely that the types of vortices shed at the optimal specific wavelength is
Figure 4.11. Vortical structures shed from free–swimming eel and mackerel. Three-dimensional vortical structures visualized for free–swimming simulations of an eel (a–c) and mackerel (d–f) at three different SW values. The wakes are visualized using isosurfaces of q–criterion, where \( q = \frac{1}{2} (\|A\|^2 - \|S\|^2) \), where \( A \) and \( S \) are the antisymmetric and symmetric parts of the fluid velocity gradient tensor \( \nabla \mathbf{u} \), respectively.

beneficial in some way to propulsive performance when compared to non–optimal SW values. The anatomy of these wakes should be studied in future numerical and experimental work.

Additionally, future parametric studies should further consider the effect of different amplitude and span profiles along the length of the swimming body: would a high AR swimmer with anguilliform amplitude profile maximize thrust or swimming speed at the same SW as a low AR swimmer with carangiform amplitude profile? These types of simulations were carried out in \[162\]; for a given body (with either low or high AR) it was found that anguilliform kinematics produce faster swimming speeds than carangiform
kinematics, which is consistent with the simulations carried out in this work. Additionally, for a given undulation profile \( A(x) \), the higher AR body produced a faster swimming speed or thrust \[ 162 \], which is also shown in our sheet simulations. However, the location of the OSW changes with respect to AR for low Reynolds number sheets; whether this parametric dependence holds for more realistic swimming bodies is yet to be explored.

For the BCF swimmers considered in this study, there is an adherence to a SW \( \sim 10 \) constraint and numerical simulations suggest a possible reason for the emergence of this rule: specific wavelength maximizes the thrust or swimming speed generated by an undulatory swimmer. Similarly, it is hypothesized that the St \( \sim 0.3 \) constraint found in flying and swimming animals emerged because the Strouhal number is a maximizer of propulsive efficiency. Consistency of two more measurements, the flexion ratio (FR) and maximum flexion angle (\( \theta_f \)), has been observed among propulsors of multiple taxonomic groups, length scales, and fluid media \[ 163 \]. These quantities describe the extent of bending in non–anguilliform fish that primarily undulate their caudal peduncle and fin. These bending characteristics are important to consider because they encode information about the amplitude profile along the length of a swimming body in a dimensionless way. Although we did conduct variable amplitude profile simulations in this study, we did not study the parametric effect of FR or \( \theta_f \) on swimming speed or efficiency. We hypothesize that these parameters also induce an optimality principle. This is evidenced by studies showing the effect of variable stiffness profiles and bending ratios on propulsive performance \[ 164, 165 \]. Future work should explore these universal bending patterns for all BCF swimmers and their relationship to propulsive performance, St, and SW in order to unify various design rules for human–engineered, underwater propulsors. Doing
so would specify the $\lambda$, $f$ and $A(x)$ needed to build a hydrodynamically optimal swimming machine.

4.4. Materials and methods

4.4.1. Experimental data

Analysis was done on kinematic data for 26 species of steady–swimming undulatory fish, and 1 species of salamander from a meta–analysis study [148]. Table 4.1 contains relevant data for the swimmers considered in the present study. These data were selected for analysis because a well–defined wavelength, frequency, amplitude, and swimming speed of the animals were reported.

Data on aspect ratio (AR) for each of the species were also measured from schematics provided by the sources listed in Table 4.1 when available, or side–view images from [http://www.fishbase.org/]. Data on $Re = \rho UL/\mu$ were provided for each specimen in [148]. However, for a more direct comparison between observations and simulations, the data were reinterpreted in terms of $Re_{lat}$. We found that these two dimensionless quantities are strongly monotonically correlated with Spearman’s rank correlation coefficient $r_s = 0.940$ (see section 4.4.3).

4.4.2. Numerical simulations

The three-dimensional numerical simulations of the undulating sheet were carried out using the constraint–based immersed boundary method (cIB) developed within the IBAMR software [13, 183]. IBAMR is an immersed boundary (IB) method implementation with support for adaptive mesh refinement (AMR) and distributed memory parallelism [14].
Table 4.1. Mean specific wavelength, lateral Reynolds number, and aspect ratio for the organisms studied in this work. *Anguilliform swimmers. †Swimmer has a distinct caudal fin. All Clupea harengus and Danio rerio specimens were larval. Aspect ratio data are measured from side-view images or schematics of swimmer and not related to the listed $N_t$ value.

<table>
<thead>
<tr>
<th>Species</th>
<th>$N_t$</th>
<th>$SW_x \pm$ 1 s.d.</th>
<th>$Re_{st} \pm$ 1 s.d.</th>
<th>AR</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abramis brama (Cyprinidae)†</td>
<td>1</td>
<td>7.09</td>
<td>$1.75 \times 10^4$</td>
<td>0.242</td>
<td>148</td>
</tr>
<tr>
<td>Ambystoma mexicanum (Ambystomatidae)†</td>
<td>8</td>
<td>5.87 ± 1.06</td>
<td>$1.41 \times 10^4 \pm 6.86 \times 10^3$</td>
<td>0.126</td>
<td>148, 157</td>
</tr>
<tr>
<td>Ambystoma mexicanum young*</td>
<td>3</td>
<td>4.34 ± 1.19</td>
<td>$3.62 \times 10^4 \pm 3.05 \times 10^3$</td>
<td>0.102</td>
<td>148, 158</td>
</tr>
<tr>
<td>Ammodytes marinus (Ammodiptidae)†</td>
<td>4</td>
<td>8.16 ± 0.40</td>
<td>$4.48 \times 10^4 \pm 1.78 \times 10^3$</td>
<td>0.056</td>
<td>148</td>
</tr>
<tr>
<td>Anguilla anguilla (Anguillidae)</td>
<td>4</td>
<td>8.26 ± 1.16</td>
<td>$3.33 \times 10^4 \pm 2.34 \times 10^3$</td>
<td>0.065</td>
<td>148, 166</td>
</tr>
<tr>
<td>Anguilla rostrata (Anguillidae)</td>
<td>1</td>
<td>8.72</td>
<td>$6.42 \times 10^4$</td>
<td>0.065</td>
<td>148, 167</td>
</tr>
<tr>
<td>Carassius auratus (Cyprinidae)†</td>
<td>1</td>
<td>8.58</td>
<td>$1.38 \times 10^4$</td>
<td>0.254</td>
<td>148</td>
</tr>
<tr>
<td>Chelon labrosus risso (Mugilidae)†</td>
<td>2</td>
<td>13.96 ± 2.31</td>
<td>$3.75 \times 10^4 \pm 1.67 \times 10^3$</td>
<td>0.186</td>
<td>148, 168</td>
</tr>
<tr>
<td>Clupea harengus (Clupeidae)†</td>
<td>6</td>
<td>4.92 ± 0.06</td>
<td>$2.55 \times 10^4 \pm 1.69 \times 10^3$</td>
<td>0.114</td>
<td>148, 169, 170</td>
</tr>
<tr>
<td>Danio rerio (Cyprinidae)</td>
<td>11</td>
<td>5.36 ± 1.29</td>
<td>$5.34 \times 10^4 \pm 3.43 \times 10^3$</td>
<td>0.115</td>
<td>148, 171, 149</td>
</tr>
<tr>
<td>Esox (hybrid) (Esocidae)†</td>
<td>1</td>
<td>11.92</td>
<td>$4.43 \times 10^4$</td>
<td>0.140</td>
<td>148, 172</td>
</tr>
<tr>
<td>Euthynus affinis (Scombridae)†</td>
<td>4</td>
<td>14.82 ± 1.11</td>
<td>$9.31 \times 10^4 \pm 3.82 \times 10^3$</td>
<td>0.175</td>
<td>148, 173</td>
</tr>
<tr>
<td>Gadus morhua (Gadidae)†</td>
<td>6</td>
<td>11.59 ± 2.28</td>
<td>$2.39 \times 10^4 \pm 1.74 \times 10^3$</td>
<td>0.169</td>
<td>148, 174</td>
</tr>
<tr>
<td>Gambusia affinis (Poeciliidae)†</td>
<td>6</td>
<td>7.50 ± 1.94</td>
<td>$1.24 \times 10^4 \pm 3.95 \times 10^3$</td>
<td>0.200</td>
<td>148, 175</td>
</tr>
<tr>
<td>Hyperoplus lanceolatus (Ammodytidae)†</td>
<td>5</td>
<td>10.09 ± 2.08</td>
<td>$1.11 \times 10^4 \pm 9.22 \times 10^3$</td>
<td>0.077</td>
<td>148</td>
</tr>
<tr>
<td>Lepisosteus osseus (Lepisosteidae)†</td>
<td>7</td>
<td>11.89 ± 0.90</td>
<td>$1.99 \times 10^4 \pm 6.50 \times 10^3$</td>
<td>0.064</td>
<td>148, 176</td>
</tr>
<tr>
<td>Leuciscus leuciscus (Cyprinidae)†</td>
<td>1</td>
<td>6.03</td>
<td>$3.08 \times 10^4$</td>
<td>0.176</td>
<td>148</td>
</tr>
<tr>
<td>Liza ramada (Mugilidae)†</td>
<td>1</td>
<td>11.40</td>
<td>$3.94 \times 10^4$</td>
<td>0.168</td>
<td>148</td>
</tr>
<tr>
<td>Micropterus salmoides (Centrarchidae)†</td>
<td>5</td>
<td>12.06 ± 0.87</td>
<td>$7.34 \times 10^4 \pm 2.85 \times 10^3$</td>
<td>0.222</td>
<td>148, 177</td>
</tr>
<tr>
<td>Oncorhynchus mykiss (Salmonidae)†</td>
<td>10</td>
<td>10.52 ± 1.19</td>
<td>$8.18 \times 10^4 \pm 3.24 \times 10^3$</td>
<td>0.200</td>
<td>148, 172</td>
</tr>
<tr>
<td>Pelamis platyrhincus (Hydrophidae)†</td>
<td>2</td>
<td>4.14 ± 0.34</td>
<td>$5.57 \times 10^4 \pm 2.26 \times 10^3$</td>
<td>0.070</td>
<td>148, 178, 179</td>
</tr>
<tr>
<td>Plectrocheilus platessa (Pleurostomidae)†</td>
<td>1</td>
<td>13.12</td>
<td>$5.00 \times 10^8$</td>
<td>0.347</td>
<td>148, 174</td>
</tr>
<tr>
<td>Pollachius virens (Gadidae)†</td>
<td>9</td>
<td>11.16 ± 1.25</td>
<td>$3.17 \times 10^4 \pm 1.25 \times 10^3$</td>
<td>0.179</td>
<td>148, 180</td>
</tr>
<tr>
<td>Salmo salar (Salmonidae)†</td>
<td>3</td>
<td>14.86 ± 5.02</td>
<td>$4.35 \times 10^4 \pm 2.65 \times 10^3$</td>
<td>0.138</td>
<td>148</td>
</tr>
<tr>
<td>Sarda chiliensis (Scombridae)†</td>
<td>2</td>
<td>13.43 ± 2.35</td>
<td>$5.00 \times 10^4 \pm 3.57 \times 10^3$</td>
<td>0.154</td>
<td>148, 181</td>
</tr>
<tr>
<td>Scomber japonicus (Scombridae)†</td>
<td>12</td>
<td>12.50 ± 2.21</td>
<td>$1.50 \times 10^4 \pm 9.38 \times 10^3$</td>
<td>0.143</td>
<td>148, 175</td>
</tr>
<tr>
<td>Scomberomorus (Scombridae)†</td>
<td>9</td>
<td>9.46 ± 0.83</td>
<td>$8.55 \times 10^4 \pm 4.45 \times 10^3$</td>
<td>0.134</td>
<td>148, 180</td>
</tr>
<tr>
<td>Strongylura marina (Belonidae)†</td>
<td>3</td>
<td>13.50 ± 3.02</td>
<td>$5.26 \times 10^4 \pm 5.48 \times 10^3$</td>
<td>0.050</td>
<td>148, 182</td>
</tr>
</tbody>
</table>

The cIB method involves solving the Navier–Stokes momentum and mass conservation equations for a combined fluid and solid domain. The sheet kinematics and motion are represented in a Lagrangian frame and are treated as a constraint force in the momentum equation. The fluid motion is solved for on an Eulerian grid with no-slip or periodic boundary conditions used on all faces of the computational domain. We found that the choice of boundary conditions on the faces of the computational domain did not affect the thrust computation on the body of the swimmer given that the swimmer was far enough away from simulation boundaries. The body’s lateral undulations was
given by a traveling wave $y(x, t) = A(x) \sin(2\pi(x/\lambda - ft))$, where the amplitude profile is either constant ($A(x) = a$) or variable ($A(x) = ae^{x/L-1}$ for anguilliform swimmers and $A(x) = a(0.2 - 0.8x/L + 1.6(x/L)^2)$ for carangiform swimmers. No-slip boundary conditions were enforced on the surface of the swimmer.

Three different undulating bodies were considered in this study: sheets, eels, and mackerels. Top- and side-view visualizations from free-swimming simulations are shown in Fig 4.12. Simulations were carried out for a wide range of morphological and kinematic parameters as described in the text. The fluid had physical properties corresponding to water at 25°C with density $\rho = 1 \text{ g/cm}^3$ and viscosity $\mu = 0.89 \times 10^{-2} \text{ g/(cm} \cdot \text{s)}$.

A grid-convergence test was conducted for every set of simulations in order to validate the numerical technique. Let $\Delta X = (\Delta x, \Delta y, \Delta z)$ be the grid-spacing at the finest mesh level and $q$ be the desired measured quantity (i.e. axial force or swimming speed). Simulations were conducted at an initial grid-spacing $\Delta X_0$ with measured quantity $q_0$, and a refined grid-spacing $\Delta X_1 = \Delta X_0/2$ with measured quantity $q_1$. If the percent change $\delta = |q_1 - q_0|/q_0 \times 100\%$ between these two measurements was less than 10%, then the rest of the simulations in that set were conducted at the coarser grid-spacing $\Delta X_0$. Otherwise, we ran a more refined case at $\Delta X_2 = \Delta X_0/4$ and compared $q_1$ and $q_2$. This refinement continued until a $\delta < 10\%$ was achieved and a grid-spacing was decided. This process was done every time $\text{Re}_{\text{lat}}$ changed by an order of magnitude. All simulations were carried out at a finest grid-spacing $(5.86, 4.81, 4.46) \times 10^{-3} \text{ cm} \leq \Delta X \leq (6.25, 6.25, 6.25) \times 10^{-2} \text{ cm}$ and time-step $1 \times 10^{-4} \text{ s} \leq \Delta t \leq 5 \times 10^{-4} \text{ s}$. A similar grid-refinement study was conducted in [121] to validate the drag measurements on a mackerel swimmer at $\text{Re} = 4000$. 
Figure 4.12. Wake and body visualizations from free–swimming simulations. Top figures show the midline kinematics (black) over time for the three different types of undulating bodies considered in the present numerical study. Dashed red line denote the amplitude function $\pm a(x)$. Middle and bottom figures show contours of vorticity magnitude for the three bodies. Middle figures show the top–view with undulations present in the lateral direction, while bottom figures show the side–view of each body. (a), (b), & (c) An undulating flat plate with SW = 10 and $Re_{\text{lat}} = 4.49$; the low Reynolds number causes the wake to remain large and mostly attached. (d), (e) & (f) An undulating eel body with SW = 10 and $Re_{\text{lat}} = 1.12 \times 10^2$. (g), (h), & (i) An undulating mackerel body with SW = 10 and $Re_{\text{lat}} = 1.12 \times 10^2$.

For free–swimming simulations, the various Reynolds numbers ranges were as follows: $Re \in [2.25 \times 10^2, 5.95 \times 10^3]$, $Re_{\text{lat}} \in [4.49 \times 10^0, 1.12 \times 10^2]$, $Re_{\text{lat,L}} \in [4.49 \times 10^1, 1.12 \times 10^2]$, and $Re_{\text{wave}} \in [2.25 \times 10^2, 4.49 \times 10^4]$. For translation–locked simulations, the various Reynolds numbers ranges were as follows: $Re = 0$, $Re_{\text{lat}} \in [2.81 \times 10^{-1}, 3.37 \times 10^2]$, $Re_{\text{lat,L}} \in [1.12 \times 10^1, 6.74 \times 10^3]$, and $Re_{\text{wave}} \in [4.21 \times 10^1, 2.70 \times 10^5]$. Since the Reynolds numbers considered were relatively modest (except for $Re_{\text{wave}}$, which is generally much larger than Re during free–swimming; see section 4.4.3) and the high velocities were confined to the region close to the plate, no turbulence model was used in the present
study. All the simulations were conducted with zero incoming velocity. Visualizations from the simulations were checked to ensure that the domain sizes were sufficiently large to minimize the interaction between the walls of the computational domain and the immersed body.

Fig 4.13 shows an example of the computational setup and grid–refinement validation for a translation–locked sheet undulating at Re_{lat} = 3.37 \times 10^2 and SW = 20. For the coarser grid simulation, \( \Delta X_0 = (3.125, 3.125, 3.125) \times 10^{-3}L \), while for the refined grid simulation, \( \Delta X_1 = (1.5625, 1.5625, 1.5625) \times 10^{-3}L \). The mean axial force measurements for each case are \( q_0 = 33.6 \text{ mN} \) and \( q_1 = 34.3 \text{ mN} \), which represents a percent change of \( \delta = 2.08\% \). This gives us confidence that the high Re_{lat} simulations carried out in this work are relatively insensitive to grid spacing up to the prescribed \( \delta \) tolerance.
4.4.3. Correlation between various Reynolds numbers

The Reynolds number \( \text{Re} = \frac{\rho U_x L}{\mu} \), where \( U_x \) is the forward swimming speed of the animal, is generally used to categorize the ratio of inertial to viscous forces in free-swimming \([148]\). However for the translation–locked simulations carried out in this study, this conventional Reynolds number is always zero and does not provide any useful information about the flow regime. In Sec.\([4.2.2.1]\) we presented a nondimensionalization of the axial force generated by a translation–locked undulating fin and a dimensionless quantity called the lateral Reynolds number \( \text{Re}_{\text{lat}} = \frac{\rho (f a) a}{\mu} \) is obtained. We note in the main text that a variety of Reynolds numbers could replace \( \text{Re}_{\text{lat}} \) in the nondimensionalization, including the wave–speed Reynolds number \( \text{Re}_{\text{wave}} = \frac{\rho (\lambda f) L}{\mu} \) or \( \text{Re}_{\text{lat},L} = \frac{\rho (f a) L}{\mu} \).

The Spearman’s rank correlation coefficient \( r_s(X,Y) \) measures how well the relationship between two quantities can be described by a monotonic function. A \( r_s \) value close to 1 indicates that \( X \) and \( Y \) increase nearly in tandem. In Fig \([4.14]\) we show that all these dimensionless quantities correlate strongly with \( \text{Re} \) for the swimming animals considered in \([148]\). For these data, \( r_s(\text{Re}_{\text{lat}},\text{Re}) = 0.940 \), \( r_s(\text{Re}_{\text{lat},L},\text{Re}) = 0.974 \), and \( r_s(\text{Re}_{\text{wave}},\text{Re}) = 0.988 \). Hence, our analysis is independent of the chosen dimensionless Reynolds number. We chose to analyze the data with respect to \( \text{Re}_{\text{lat}} \) because it is independent of wavelength.

4.4.4. Parametric study: Dimensionless force results

For a translation–locked sheet of length \( L \) and span \( h \), the force generated by the undulations depends on the physical input parameters of the system Eq.\([4.2]\) with nondimensionalization Eq.\([4.3]\).
Figure 4.14. Correlation between different Reynolds numbers. The swimming-speed Reynolds number $Re$ vs. $Re_{lat}$ (●; blue), $Re_{lat,L}$ (♦; purple), and $Re_{wave}$ (■; red) for the swimmers considered in [148], along with the lines of best fit. The best fit lines are given by $\log Re = 1.803 + 1.340 \log Re_{lat}$, $\log Re = -0.4058 + 1.2306 \log Re_{lat,L}$, and $\log Re = -1.450 + 1.083 \log Re_{wave}$.

Earlier, we presented axial force data vs. specific wavelength from a numerical parametric study of sinusoidally undulating sheets (see Figs 5−9). In Figs 4.15−4.19 we show the dimensionless force $\hat{F}$ data from these same simulations. In all of the simulations considered in this study, the OSW that maximizes $F_x$ and $\hat{F}$ are the same.

4.4.5. Quantifying the friction and velocity mechanisms

Let an undulating sheet (or fish body) of length $L$, span $h$, constant amplitude $a$, frequency $f$ and wavelength $\lambda$ be immersed in a fluid with density $\rho$ and velocity $\mathbf{u}(x, y, z, t) =$
Figure 4.15. Dimensionless axial force data from Fig 4.5. Dimensionless axial propulsive force $\hat{F}$ generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate span was varied. These data represent cases where (a) $Re_{\text{lat}} = 8.43 \times 10^{-1}$ and (b) $Re_{\text{lat}} = 4.49$.

Let $\Omega$ be the domain containing the sheet. Consider that the sheet is undulating but not translating and that there is no externally imposed flow on the sheet.

A stationary undulating sheet sucks stationary fluid from the front end and accelerates it to some momentum at the downstream end. This is applicable at all finite Reynolds number flows in this work. The fluid momentum ejected at the downstream end is usually manifested in the form of a wake, which is prominent at high Reynolds numbers. The acceleration of the fluid from front to back is caused by a backwards force from the sheet on the fluid, which by Newton’s third law is equal to the forward thrust force from the fluid on the sheet (Fig. 4.20). The momentum in the wake is eventually dissipated/decelerated.
Figure 4.16. Dimensionless axial force data from Fig 4.6. Dimensionless axial propulsive force $\hat{F}$ generated by a stationary undulating sheet plotted against specific wavelength. In both cases (a) & (b), plate length was varied. These data represent cases where $\text{Re}_{\text{lat}} = 3.37 \times 10^2$.

Further downstream (Fig 4.20). This dissipation corresponds to a net force on the walls of the domain, which, under “steady” conditions, is equal and opposite to the thrust force on the sheet. This can be formally shown by applying momentum conservation to appropriate control volumes. Consequently, the forward (thrust) force $F_x$ on a stationary undulatory body can be estimated by \cite{184, 185, 186, 187, 159}

\begin{equation}
F_x \sim \mathcal{M}_{\text{wake}}(\lambda f)f,
\end{equation}
in which $M_{\text{wake}}$ is a measure of the fluid mass mobilized by the sheet (also added mass in reactive theory for thrust generation [187]) or ejected into the wake, $\lambda f$ is the wave velocity of undulation, and $f$ is the frequency of undulation.

In an earlier work [151] and in this work we find that $F_x$ is maximized at some optimal specific wavelength (OSW) of undulation. It was hypothesized in the earlier work [151] that two competing mechanisms cause thrust to be maximized at the optimal specific wavelength. A high wavelength $\lambda$ leads to a higher wave velocity ($= \lambda f$) and fluid is pushed backward faster at a given frequency; this was called the velocity mechanism. However, since the waves are “shallower” at high wavelength, fluid is transported less
Figure 4.18. Dimensionless axial force data from Fig 4.8. Dimensionless axial propulsive force $\hat{F}$ vs. specific wavelength for additional small-sheet simulations with varying aspect ratios. In case (a), plate span was varied while in case (b), plate length was varied. These data represent cases where (a) $Re_{lat} = 8.43 \times 10^{-2}$ and (b) $Re_{lat} = 4.49$.

efficiently. Conversely, a low $\lambda$ leads to “deeper” waves and thus fluid is transported backwards more efficiently, but it is done so at a slower speed; this was called the friction mechanism. In the context of Eq. 4.4, $\mathcal{M}_{wake}$ would quantify the friction mechanism and $\lambda f$ would quantify the velocity mechanism. To interrogate the plausibility of the hypothesis for OSW proposed in [151], each of the quantities $F_x$, $\mathcal{M}_{wake}$, and $\lambda f$ need to be quantified independently.

The thrust $F_x$ is computed directly by integrating forces on the sheet and $\lambda f$ is known based on chosen parameters. In Fig. 4.21, we show that the maximum fluid axial velocity $u_{max}$ generated by the sheet over a single swimming cycle is directly correlated to the
Figure 4.19. Dimensionless axial force data from Fig 4.9. Dimensionless axial propulsive force $\hat{F}$ generated by a stationary undulating sheet with prescribed anguilliform (▲) and carangiform (●) amplitude profiles plotted against specific wavelength. These data represent cases where $Re_{int} = 4.49$.

traveling wave velocity $\lambda f$. Therefore, the velocity mechanism is quantified by wave speed.

It is not straightforward to compute $M_{wake}$, which quantifies how much fluid is mobilized and accelerated by the force $F_x$ exerted by the sheet on the fluid. Note that downstream from the wake the fluid is in fact decelerated, which corresponds largely to forces on the wall. In general, the accelerating and decelerating domains of the fluid surrounding the sheet are not clearly separated.

In order to overcome the above issue, we choose to consider a different but related problem. We perform undulating sheet simulations in a fully periodic domain – no wall boundaries are included. In this case, the force from the sheet on the fluid causes the fluid
in the entire domain to be accelerated through one period of undulation – the fluid does not decelerate downstream from the sheet because there is no resistance from the walls.

The total momentum of the fluid in the computational domain is given component-wise as

\[(m_x, m_y, m_z) = \int_{\Omega} \rho u \, dV.\]

We choose to analyze a sheet with \(L = 1.0\) cm, \(h = 0.1\) cm, \(f = 3\) Hz, \(a = 0.05\) cm with three different SW values: 5, 10, and 13.33. For this sheet, axial thrust is maximized at SW = 10. The sheet occupies the area \([0, 1] \times [-0.1, 0]\) in the \(xz\)-plane with minimum and maximum extents in the undulation direction at \(y = -0.05\) cm and \(y = 0.05\) cm. For these simulations, \(\Omega = [-1.7, 4.3] \times [-1.1, 0.9] \times [-1.2, 0.8]\) with periodic boundary conditions on all faces, unless otherwise stated.

In our simulated sheet cases, we compute the average thrust force over a full swimming cycle \([t_0, t_0 + T]\), starting at some time \(t_0\) after an oscillatory steady-state has been reached in the axial force. Here, \(T = 1/f\) is the sheet period. In Fig 4.22, we show \(F_x\) directly computed by integrating forces on the sheet. We show \(F_x\) computed on the sheet in a fully periodic domain and for another case where there are walls present. All other parameters, including domain size are the same. Both forces are identical. As desired, the periodic boundary condition did not affect thrust production.

The average axial force should satisfy the following equation,

\[(4.6) \quad F_x = f \cdot [m_x(t_0 + T) - m_x(t_0)] = f \cdot \Delta m_x,\]
where \( m_x(t) \) is in general nonzero at \( t_0 \). Fig 4.22 also plots \( f \Delta m_x \) in a periodic domain with \( f = 3 \) Hz for all cases. The comparison between \( F_x \) and \( f \Delta m_x \) is consistent with Eq. 4.6; the error is attributed to numerical integration.

Now, we argue that the two aforementioned mechanisms (velocity and friction) can be defined by analyzing the change in fluid momentum in Eq. 4.6. Note that although a large \( \lambda \) sheet may generate high velocities, the volume over which these high velocities are realized also contributes to the momentum (through integration over volume elements \( dV \) in Eq. 4.5). There are no clearly separated domains of mobile or immobile fluid. Therefore, we choose to quantify the amount of fluid being transported by using a weight factor \( u_x/\lambda f \) over the entire volume. Mathematically, this is equivalent to splitting \( m_x \) into

\[
(4.7) \quad m_x = \lambda f \int_{\Omega} \rho \frac{u_x}{\lambda f} dV = \lambda f \cdot \mathcal{M},
\]

where \( \mathcal{M} \) is measured in unit of mass. The integral term equal to \( \mathcal{M}(t) \) represents the amount of mobilized fluid weighted by \( u_x/\lambda f \); this quantifies how much fluid is transported at axial velocities close to the theoretical maximum. Using Eq. 4.6, the average thrust force on the sheet is given by

\[
(4.8) \quad F_x = f \cdot \lambda f \left( \mathcal{M}(t_0 + T) - \mathcal{M}(t_0) \right) = f \cdot \lambda f \cdot \Delta \mathcal{M},
\]

where \( \lambda f \) and \( \Delta \mathcal{M} \) are, by definition, quantitative measures of the velocity and friction mechanisms, respectively. Eq. 4.8 is analogous to Eq. 4.4. \( \Delta \mathcal{M} \) quantifies how much
additional fluid is mobilized by force $F_x$ over one periodic interval of duration $T$. Quantitatively, the competing mechanisms are shown in Fig. 4.22. As SW increases, $\lambda f$ increases, while $\Delta M$ decreases, as hypothesized.

Figs. 4.23 and 4.24 show velocity contours for the cases considered. In Fig. 4.23 contours of velocity in the immediate vicinity of the sheet are plotted for three different SW values for two snapshots in time. Notice that as SW decreases, the maximum velocity also decreases (also plotted in Fig. 4.21). However in Fig. 4.24 we show the same instances in time, but instead plot contours of $u_x/u_{\text{max}}$. We choose to normalize by the maximum fluid axial velocity $u_{\text{max}}$ generated by the sheet over $\Omega$ and over one swimming cycle $[t_0, t_0 + T]$ to illustrate the differences between these sheets. Now as SW decreases, the area of influence of the maximum velocities increases i.e. there are more areas of dark red in the SW = 5 case than in the SW = 13.33 case.
Figure 4.20. Momentum transfer by an undulatory swimmer. At the front of the swimmer, the body sucks stationary fluid with negligible momentum $m_{in}$ and accelerates it downstream. This ejected fluid is often manifested as a wake with momentum $m_{wake} \sim M_{wave}(\lambda f)$. Finally, this wake eventually dissipates further downstream as it loses momentum.

Figure 4.21. Correlation between maximum axial fluid velocity and traveling wave velocity. The maximum fluid axial velocity $u_{max}$ generated by the sheet over $\Omega$ and over one swimming cycle $[t_0, t_0 + T]$ vs. the traveling wave speed $\lambda f$. 
Figure 4.22. Quantitative analysis of the velocity and friction mechanisms. (a) Net axial force generated by simulations of an undulating plate with $L = 1.0$ cm, $h = 0.1$ cm, $f = 3$ Hz, $a = 0.05$ with (red) periodic boundary conditions, and (black) wall boundary conditions. Average axial fluid momentum multiplied by undulation frequency over one swimming cycle with periodic boundary conditions (green) as functions of SW. (b) Velocity of the traveling wave $\lambda f$ (pink) and change in weighted mass (blue).

Figure 4.23. Qualitative analysis of the velocity mechanism. Mid–sheet ($z = −0.05$ cm) contours of axial fluid velocity for a stationary sheet simulation for various snapshots in time. Black dots represent the Lagrangian points of the undulating body. The plate has $L = 1.0$ cm, $h = 0.1$ cm, $f = 3$ Hz, $a = 0.05$ cm with (a) & (b) SW = 13.33, (c) & (d) SW = 10, (e) & (f) SW = 5.
Figure 4.24. Qualitative analysis of the friction mechanism. Mid-sheet ($z = -0.05$ cm) contours of normalized axial fluid velocity ($u_x/u_{\text{max}}$) for a stationary sheet simulation for various snapshots in time. Black dots represent the Lagrangian points of the undulating body. The plate has $L = 1.0$ cm, $h = 0.1$ cm, $f = 3$ Hz, $a = 0.05$ cm with (a) & (b) SW = 13.33, (c) & (d) SW = 10, (e) & (f) SW = 5. For each difference SW, $u_{\text{max}}$ is taken to be the maximum axial fluid velocity over $\Omega$ and over one swimming cycle.
CHAPTER 5

A moving control volume approach to computing hydrodynamic forces and torques on immersed bodies

In this chapter, we present a moving control volume (CV) approach to computing hydrodynamic forces and torques on complex geometries. The method requires surface and volumetric integrals over a simple and regular Cartesian box that moves with an arbitrary velocity to enclose the body at all times. The moving box is aligned with Cartesian grid faces, which makes the integral evaluation straightforward in an immersed boundary (IB) framework. Discontinuous and noisy derivatives of velocity and pressure at the fluid-structure interface are avoided and far-field (smooth) velocity and pressure information is used. We re-visit the approach to compute hydrodynamic forces and torques through force/torque balance equation in a Lagrangian frame from a prior work [13]. We prove the equivalence of the two approaches for IB methods, thanks to the use of Peskin’s delta functions. Both approaches are able to suppress spurious force oscillations and are in excellent agreement, as expected theoretically. Test cases ranging from Stokes to high Reynolds number regimes are considered. We discuss regridding issues for the moving CV method in an adaptive mesh refinement (AMR) context. The proposed moving CV method is not limited to a specific IB method and can also be used, for example, with embedded boundary methods.
5.1. Introduction

Fluid-structure interaction (FSI) problems involving moving bodies is a challenging area in the computational fluid dynamics field that has vested the interest of researchers for several decades. FSI modeling has traditionally been carried out in two ways: the body-fitted mesh approach using unstructured grids [107, 109] and the Cartesian grid approach based on the fictitious domain method [6, 7]. Although the body-fitted mesh approach to FSI resolves the fluid-structure interface sharply, it requires complex mesh management infrastructure along with high computational costs for solving linear equations. The fictitious domain method on the other hand extends the fluid equations inside the structure along with some additional non-zero body forcing term. As a result, regular Cartesian grids and fast linear solvers such as fast Fourier transform can be used to solve the common momentum equation of the continua. However, fictitious domain methods tend to smear the fluid-structure interface and hence reduce the solution accuracy near the interface.

One widely used fictitious domain approach to FSI is the immersed boundary (IB) method [5] which was originally proposed by Peskin in the context of cardiac flows [4]. The main advantage to IB methods is that they do not require a body-fitted mesh to model the structure. The immersed body is allowed to freely cut the background Cartesian mesh, making the IB method easy to implement within an existing incompressible flow solver. A Lagrangian force density is computed on structure nodes, which is then transferred to the background grid via regularized delta functions. The use of regularized delta functions diffuses the interface and smears it over a number of grid cells proportional to the width
of the delta function. This often leads to discontinuous or noisy derivatives of the velocity and pressure field required to compute surface traction.

The original IB method uses an explicit time stepping scheme and fiber elasticity to compute the additional body forcing inside the region occupied by the structure. The method works fairly well for soft elastic structures, but incurs severe time step restrictions if the stiffness of the material is substantially increased to model rigid bodies. Specialized versions of the original IB method have been developed to model rigid and stiff bodies in an efficient manner like the implicit IB method [188, 189, 190, 191, 192], the direct forcing method [193, 13, 183], and the fully constrained IB method [194, 111, 112]. The implicit IB method requires special solvers like algebraic [195] and geometric multigrid [188, 191] to treat the stiff elastic forces implicitly. The direct forcing and fully constrained IB methods that are primarily used for modeling rigid bodies impose rigidity constraint through Lagrange multipliers. The direct forcing method approximates the Lagrange multiplier with suitable penalty term and solves the fluid-structure equations in a fractional time stepping scheme. This is in contrast to fully constrained IB method where the fluid velocity, pressure, and Lagrange multipliers are solved together. Apart from the obvious advantage of imposing the rigidity constraint exactly rather approximately, the fully constrained IB method can be used for Stokes flow where fractional time stepping schemes do not work. However, special preconditioners are required to solve for the Lagrange multipliers exactly [111, 112], which makes solving the system costly for big three dimensional volumetric bodies.
A common feature of IB methods that are based on Peskin’s IB approach [5] is that they do not require rich geometric information like surface elements and normals to compute the Lagrangian force density. Structure node position (and possibly the node connectivity) information suffices to compute Lagrangian forces. However, there are many other versions of the IB method and Cartesian grid based methods that require additional geometric information to sharply resolve the fluid-structure interface. Examples include the immersed finite element method [196, 197, 198, 199], the immersed interface method [200, 201], the ghost-fluid method [116], and the cut-cell embedded boundary method [114, 115, 202]. For such approaches, the availability of surface elements and normals along with sharp interface resolution makes the surface traction computation easier and smooth (for at least smooth problems).

Often times, the net hydrodynamic forces and torques on immersed bodies are desired rather than point-wise traction values. Force/torque balance equations can be used to compute the hydrodynamic force/torque contribution instead of directly integrating surface traction. In the context of “Peskin-like” IB methods, this implies that one can essentially eliminate surface mesh generation as a post-processing step if only net hydrodynamic forces and torques are desired. This has another added advantage of not using noisy derivatives of velocity and pressure at the interface for evaluating hydrodynamic forces and torques.

In this chapter we analyze and compare two approaches to computing net hydrodynamic forces and torques on an immersed body. In the first approach we use the Reynolds transport theorem (RTT) to convert the traction integral over an irregular body surface to a traction integral over a regular and simple Cartesian box (that is aligned with grid
faces). The RTT is proposed for a moving control volume that translates with an arbitrary velocity to enclose the immersed body at all times. We refer to this approach as the moving CV approach. In the context of locally refined grids, the moving control volume can span a hierarchy of grid levels. In the second approach, hydrodynamic forces and torques are computed using inertia and Lagrange multipliers (approximate or exact) defined in the body region. We refer to this approach as the LM approach and has been used before in [13]. For IB methods, we show that both approaches are equivalent. This is due to a special property of Peskin’s delta functions that makes Lagrangian and Eulerian force density equivalent [5]. We show that both these approaches give smooth forces and suppress spurious oscillations that arise by directly integrating spatial pressure and velocity gradients over the immersed body as reported in the literature [117].

Although application of the RTT on a stationary control volume to evaluate hydrodynamic force is a well known result [203, 204]; its extension to moving control volumes was first proposed by Flavio Noca in 1997 [205]. They were motivated by the task of evaluating net hydrodynamic force on a moving bluff body using DPIV data from experiments. Flavio has also proposed force expressions that eliminate the pressure variable; a quantity not available in DPIV experiments [205, 206]. We do not analyze such expressions in this work, however. In the context of IB method, Bergmann and co-workers [207, 208] have used Flavio’s moving control volume force expressions (involving both velocity and pressure) to compute hydrodynamic forces. They observed spurious force oscillations with a moving control volume approach [207]. We show that by manipulating time derivatives in the original expressions, one can eliminate such spurious oscillations. We also present strategies to mitigate jumps in velocity derivatives in an AMR framework. Such jumps
arise when the Cartesian grid hierarchy is regridded and velocity in the new grid hierarchy is reconstructed from the old hierarchy \cite{209}. Lai and Peskin \cite{210} used stationary control volume analysis on a uniform grid to compute steady state hydrodynamic forces on a stationary cylinder for Reynolds numbers between $100 - 200$ using the IB method. They did not consider time derivative terms in their analysis and temporal hydrodynamic force profiles were reported at steady state. In this work we include time derivative terms for finite Reynolds number flows (but not for steady Stokes flow) in our (moving) control volume analysis.

If point-wise traction values are desired for IB-like methods, there are several recommendations proposed in the literature for smoothing them. Here we list a few of them. Verma et al. \cite{211} have recommended using a “lifted” surface: a surface two grid cell distance away from the actual interface to avoid choppy velocity gradients. This recommendation, based upon their empirical tests using a Brinkman penalization method, can change depending on the smoothness of the problem and the discrete delta function used in IB methods. Goza et al. \cite{212} obtain smooth point-wise force measurements by using a force filtering post-processing step that penalizes inaccurate high frequency stress components. Martins et al. \cite{118} enforce a continuity constraint in the velocity interpolation stencil to reconstruct a second-order velocity field at IB surface points that is discretely divergence-free. They also impose a normal gradient constraint in the pressure interpolation stencil to reconstruct a second-order accurate pressure field at IB points. They have successfully eliminated spurious force oscillations using constrained least-squares stencils. The idea of unconstrained moving least-squares velocity interpolation and force spreading in the context of direct forcing IB method was first proposed by Vanella and Balaras \cite{213}. 
Lee et al.\cite{117} attribute sources of spurious force oscillations to spatial pressure discontinuities across the fluid-solid interface and temporal velocity discontinuities for moving bodies. They recommend using fine grid resolutions to alleviate spurious oscillations. Their analyses and tests\cite{117} show that grid spacing has a more pronounced effect on spurious force oscillations than computational time step size.

For a range of test cases varying from free-swimming at high Reynolds number to steady Stokes flow, we show that both LM and moving CV methods are in excellent agreement and are able to suppress spurious force oscillations. They do not require any additional treatment such as least-squares stencil (velocity and pressure) interpolation or force filtering beyond simple integration of force balance laws. For moderate to high Reynolds number test cases, we use a direct forcing IB method to estimate Lagrange multipliers, and for Stokes flow we use a fully constrained IB method to compute Lagrange multipliers exactly.

### 5.2. Equations of motion

#### 5.2.1. Immersed boundary method

The immersed boundary (IB) formulation uses an Eulerian description for the momentum equation and divergence-free condition for both the fluid and the structure. A Lagrangian description is employed for the structural position and forces. Let \( \mathbf{x} = (x_1, \ldots, x_d) \in \Omega \) denote fixed Cartesian coordinates, in which \( \Omega \subset \mathbb{R}^d \) is the fixed domain occupied by the entire fluid-structure system in \( d \) spatial dimensions. Let \( \mathbf{s} = (s_1, \ldots s_d) \in U \) denote the fixed material coordinate system attached to the structure, in which \( U \subset \mathbb{R}^d \) is the Lagrangian curvilinear coordinate domain. The position of the immersed structure
occupying a volumetric region $V_b(t) \subset \Omega$ at time $t$ is denoted by $X(s, t)$. We consider only neutrally buoyant bodies to simplify the implementation; this assumption implies that the fluid and structure share the same uniform mass density $\rho$. The deviatoric stress tensor of fluid, characterized by dynamic viscosity $\mu$, is extended inside the structure to make the momentum equation of both media appear similar. The combined equations of motion for fluid-structure system are [5]

\begin{align}
(5.1) \quad & \rho \left( \frac{\partial u(x, t)}{\partial t} + u(x, t) \cdot \nabla u(x, t) \right) = -\nabla p(x, t) + \mu \nabla^2 u(x, t) + f(x, t), \\
(5.2) \quad & \nabla \cdot u(x, t) = 0, \\
(5.3) \quad & f(x, t) = \int_U F(s, t) \delta(x - X(s, t)) \, ds, \\
(5.4) \quad & U(s, t) = \int_\Omega u(x, t) \delta(x - X(s, t)) \, dx, \\
(5.5) \quad & \frac{\partial X}{\partial t}(s, t) = U(s, t).
\end{align}

Eqs. (5.1) and (5.2) are the incompressible Navier-Stokes equations written in Eulerian form, in which $u(x, t)$ is the velocity, $p(x, t)$ is the pressure, and $f(x, t)$ is the Eulerian force density, which is non-zero only in the structure region. Interactions between Lagrangian and Eulerian quantities in Eqs. (5.3) and (5.4) are mediated by integral equations with Dirac delta function kernels, in which the $d$-dimensional delta function is $\delta(x) = \Pi_{i=1}^d \delta(x_i)$. Eq. (5.3) converts the Lagrangian force density $F(s, t)$ into an equivalent Eulerian density $f(x, t)$. In the IB literature, the discretized version of this operation is called force spreading. Using short-hand notation, we denote force spreading operation by $f = S[X] F$, in which $S[X]$ is the force-spreading operator. Eq. (5.4) determines the physical velocity
of each Lagrangian material point from the Eulerian velocity field, so that the immersed
structure moves according to the local value of the velocity field \( \mathbf{u}(\mathbf{x}, t) \) (Eq. (5.5)). This
velocity interpolation operation is expressed as \( \frac{\partial \mathbf{X}}{\partial t} = \mathbf{U} = \mathbf{J}[X] \mathbf{u} \), in which \( \mathbf{J}[X] \) is the
velocity-interpolation operator. It can be shown that if \( \mathbf{S} \) and \( \mathbf{J} \) are taken to be adjoint
operators, i.e. \( \mathbf{S} = \mathbf{J}^* \), then Lagrangian-Eulerian coupling conserves energy [5].

5.2.2. Discrete equations of motion

We employ a staggered grid discretization for the momentum and continuity equations (see
Fig. 5.2). More specifically, Eulerian velocity and force variables are defined at face centers
while the pressure variable is defined at cell centers. Second-order finite difference stencils
are used to spatially discretize the Eulerian equations on locally refined grids [13, 209].
The spatial discretization of various operators are denoted with \( h \) subscripts. To discretize
equations in time, we take \( \Delta t \) as the time step size, and \( n \) as the time step number. We
use the direct forcing method of Bhalla et al. [13] for moderate to high Reynolds number
cases and the fully constrained IB method of Kallemov et al. [111] for Stokes flow cases.
The two methods differ in how the Lagrangian force density or Lagrange multipliers are
computed. The time integrators are also different for the two methods. Here we briefly
describe the discretized equations for both methods. We refer readers to [13, 111, 112]
for more details.
For the direct forcing method, the time stepping scheme reads as \[ T3 \]

\begin{align}
\rho \left( \frac{\tilde{u}^{n+1} - u^n}{\Delta t} + [u \cdot \nabla_h u]^{(n+\frac{1}{2})} \right) &= -\nabla_h p^{n+\frac{1}{2}} + \frac{\mu}{2} \nabla^2_h (\tilde{u}^{n+1} + u^n), \\
\nabla_h \cdot \tilde{u}^{n+1} &= 0, \\
F^{n+\frac{1}{2}} &= \rho \Delta t \left( U^{n+1}_b - J_h[\mathbf{X}^{n+\frac{1}{2}}] \tilde{u}^{n+1} \right), \\
\rho \left( \frac{u^{n+1} - \tilde{u}^{n+1}}{\Delta t} \right) &= S_h[\mathbf{X}^{n+\frac{1}{2}}] F^{n+\frac{1}{2}}.
\end{align}

Succinctly, we first solve for a velocity field $\tilde{u}^{n+1}$ and a pressure field $p^{n+\frac{1}{2}}$ as a coupled system by solving Eqs. (5.6) and (5.7) simultaneously. The velocity $\tilde{u}^{n+1}$, which is correct in the fluid region but not in the structure region $V_b(t)$, is then corrected by estimating the Lagrange multiplier $F^{n+\frac{1}{2}}$ via Eq. (5.8). Here $U^{n+1}_b$ is the desired rigid body velocity of the Lagrangian nodes, and $\mathbf{X}^{n+\frac{1}{2}}$ is the midstep estimate of Lagrangian node position. Finally, the Lagrange multiplier $F^{n+\frac{1}{2}}$ is spread on the background grid to correct the momentum in the structure region to $u^{n+1}$. We use Adams-Bashforth to approximate the midstep value of nonlinear convection term $u \cdot \nabla u$ via

\begin{equation}
[u \cdot \nabla_h u]^{(n+\frac{1}{2})} = \frac{3}{2} u^n \cdot \nabla_h u^n - \frac{1}{2} u^{n-1} \cdot \nabla_h u^{n-1}.
\end{equation}

For the fully constrained method, we simultaneously solve for the updated Eulerian velocity $u^{n+1}$ and pressure $p^{n+1}$ at time $t^{n+1}$ along with the Lagrange multiplier $F^{n+1}$. 
The time stepping scheme reads as \[ \nabla_h p^{n+1} + \mu \nabla_h^2 u^{n+1} + S_h[X^n] F^{n+1} = 0, \] (5.11)
\[ \nabla_h \cdot u^{n+1} = 0, \] (5.12)
\[ J_h[X^n] u^{n+1} = U_b^{n+1}. \] (5.13)

The coupled system of Eqs. (5.11)-(5.13) is solved simultaneously using a preconditioned FGMRES solver. For both methods, we update the Lagrangian node positions \( X^{n+1} \) using rigid body translation and rotation. We use Peskin’s 4-point regularized delta functions for the \( S \) and \( J \) operators in all our numerical experiments, unless stated otherwise.

Our finite Reynolds number fluid solver has support for adaptive mesh refinement, and some cases presented in Sec. 5.5 make use of multiple grid levels (also known as a grid hierarchy). A grid with \( \ell \) refinement levels with grid spacing \( \Delta x_0, \Delta y_0, \) and \( \Delta z_0 \) on the coarsest grid level has minimum grid spacing \( \Delta x_{\min} = \Delta x_0/n_{\text{ref}}^{\ell-1}, \Delta y_{\min} = \Delta y_0/n_{\text{ref}}^{\ell-1}, \) and \( \Delta z_{\min} = \Delta z_0/n_{\text{ref}}^{\ell-1} \) on the finest grid level. Here, \( n_{\text{ref}} \in \mathbb{N} \) is the refinement ratio. In the present work, the refinement ratio is taken to be the same in each direction, although this is not a limitation of the numerical method. The immersed structure is always placed on the finest grid level. For all of the cases considered in this work a constant time step size \( \Delta t = \min(\Delta t^{\ell}) \) is chosen, in which the time step size \( \Delta t^{\ell} \) on grid level \( \ell \) satisfies the convective CFL condition \( \Delta t^{\ell} \leq C \min \left( \frac{\Delta x}{\|u_x\|_\infty}, \frac{\Delta y}{\|u_y\|_\infty}, \frac{\Delta z}{\|u_z\|_\infty} \right)^{\ell} \). In this work, the convective CFL number is set to \( C = 0.3 \) unless otherwise stated.
5.3. Hydrodynamic force and torque

5.3.1. Moving control volume method

5.3.1.1. Hydrodynamic force. Letting \( T = \mu (\nabla u + \nabla u^T) \) denote the viscous stress tensor, the net hydrodynamic force is defined to be the force of the fluid on the body:

\[
F(t) = -\oint_{S_b(t)} \mathbf{n} \cdot [-p\mathbf{I} + \mathbf{T}] \, dS,
\]

in which the integral is taken over the surface of the body \( S_b(t) = \partial V_b(t) \), and \( \mathbf{n} \) is the unit inward normal to the surface. In practice, evaluating Eq. (5.14) is inconvenient in numerical experiments because it is often difficult to obtain accurate surface velocity gradients and pressure values. Moreover, evaluating Eq. (5.14) also requires computational geometry to obtain surface normals and area. Instead we use a control volume approach to compute \( F(t) \) which avoids these requirements.

Let an arbitrary (possibly time dependent) domain \( V_{CV}(t) \) completely surround \( V_b(t) \), i.e. \( V_b(t) \subset V_{CV}(t) \), as shown in Fig. 5.1. By considering the change in momentum within the control volume \( V_{CV}(t) \) and the net momentum flux at its surface \( S_{CV}(t) = \partial V_{CV}(t) \), a general expression for the hydrodynamic force on the body can be obtained

\[
F(t) = -\frac{d}{dt} \int_{V(t)} \rho \mathbf{u} \, dV + \oint_{S_{CV}(t)} \mathbf{n} \cdot [-p\mathbf{I} - (\mathbf{u} - \mathbf{u}_S)\rho \mathbf{u} + \mathbf{T}] \, dS - \oint_{S_b(t)} \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_S)\rho \mathbf{u} \, dS,
\]

in which \( V(t) = V_{CV}(t) \setminus V_b(t) \) is the volume outside the immersed body but inside the CV, and \( \mathbf{u}_S \) is the velocity of the surface over which the (surface) integral is evaluated. In general \( \mathbf{u}_S \neq \mathbf{u} \), and \( \mathbf{u}_S \) can be arbitrarily chosen so that the moving CV always encloses the immersed body. The unit normal \( \mathbf{n} \) in Eq. (5.15) points outward on \( S_{CV}(t) \).
Figure 5.1. Sketch of the immersed structure (solid line) surrounded by an arbitrary control volume (dashed line).

and into $S_b(t)$. The above force equation was first derived by Noca and has been used in various experimental and numerical studies. For Cartesian grid based methods, the CV can be chosen as a simple rectangular domain, for which the unit normals on $S_{CV}(t)$ are aligned with the Cartesian axes. Finally, the integral over $S_b(t)$ vanishes for many applications where no-slip ($u_S = u$) boundary condition can be chosen for $u_S$. Henceforth, we will analyze cases for no-slip boundary conditions.

When Eq. (5.15) is discretized, the first term requires a discrete approximation of the integral at two separate time instances or at two different locations of the moving control volume. Bergmann et al. observed spurious force oscillations as a result of this time derivative term. We show that by manipulating the first term using the Reynolds transport theorem, an expression for hydrodynamic force can be obtained that does not
require contributions from control volumes at two different spatial locations. The modified equation reads as

\begin{equation}
F(t) = - \int_{V_{CV}(t)} \rho \frac{\partial u}{\partial t} \, dV + \frac{d}{dt} \int_{V_{b}(t)} \rho u \, dV + \oint_{S_{CV}(t)} n \cdot [-pI - u \rho u + T] \, dS.
\end{equation}

A detailed derivation of Eq. (5.16) is provided in Sec. 5.3.1.3. Note that although Eqs. (5.15) and (5.16) are equivalent formulas to obtain the hydrodynamic force on an immersed body, their physical interpretations are different. In Eq. (5.15), the control volume is moving with some prescribed velocity, usually chosen to follow the structure to ensure it is contained within the CV at all times. Hence, one must keep track of the velocity \( u_S \) of the control surface \( S_{CV}(t) \). Moreover since the time derivative appears outside of the integral over \( V(t) \), it requires a discrete approximation of momentum on two time-lagged control volumes. On the other hand in Eq. (5.16), all integrals over the control volume are evaluated at a single time instance and therefore a discrete evaluation on two separate CVs is never needed. Discretely, the control volume is placed at a new location at time step \((n+1)\) and no information from its previous location at time step \( n \) is used. Hence, \( u_S \) never appears in the calculation. This has the numerical benefit of suppressing force oscillations, which will be shown in Sec. 5.5.

**5.3.1.2. Hydrodynamic torque.** The net hydrodynamic torque on an immersed body is defined to be the net moment of hydrodynamic force exerted by fluid on the body about a given reference point:

\begin{equation}
M(t) = - \oint_{S_b(t)} r \wedge (n \cdot [-pI + T]) \, dS,
\end{equation}
in which \( \mathbf{r} = \mathbf{x} - \mathbf{x}_0 \). The torque is computed with respect to some reference point \( \mathbf{x}_0 \), which can be fixed at a location or move with time (e.g. the center of mass of a swimmer). Following Noca’s derivation for the force expression Eq. (5.15), one can measure the change in angular momentum within a moving control volume to obtain an expression for torque which reads as

\[
\mathcal{M}(t) = -\frac{d}{dt}\int_{V(t)} (\mathbf{r} \wedge \rho \mathbf{u}) \, dV + \int_{S_{CV(t)}} [\mathbf{r} \wedge (\mathbf{n} + \mathbf{n} \cdot \mathbf{T} - \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_S)(\mathbf{r} \wedge \rho \mathbf{u}))] \, dS
\]

(5.18)

\[
- \oint_{S_{b(t)}} \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_S)(\mathbf{r} \wedge \rho \mathbf{u}) \, dS.
\]

Note that the torque expression in Eq. (5.18) is slightly different from Eq. 15b given in Bergmann and Iolla [207]. Once again by applying the Reynolds transport theorem to the first term on left-hand side of Eq. (5.18), we obtain a torque expression involving a control volume contribution at a single spatial location and without any \( \mathbf{u}_S \) terms

\[
\mathcal{M}(t) = -\int_{V(t)} \rho \mathbf{r} \wedge \frac{\partial \mathbf{u}}{\partial t} \, dV + \frac{d}{dt}\int_{V(t)} \rho (\mathbf{r} \wedge \mathbf{u}) \, dV
\]

(5.19)

\[
+ \oint_{S_{CV(t)}} [\mathbf{r} \wedge (\mathbf{n} + \mathbf{n} \cdot \mathbf{T}) - (\mathbf{n} \cdot \mathbf{u})\rho (\mathbf{r} \wedge \mathbf{u})] \, dS.
\]

For the derivation of torque expressions (5.18) and (5.19), see Sec. 5.3.1.4

5.3.1.3. Derivation of the new hydrodynamic force expression. Here we present a detailed derivation of Eq. (5.16) from Eq. (5.15). The ultimate goal is to obtain an expression for hydrodynamic force which involves integral contributions from a single CV rather than two time-lagged CVs. The Reynolds transport theorem (RTT) [203, 204] gives an expression for the time derivative of an arbitrary quantity \( q(x, t) \) on a time
dependent region $\Omega(t)$

\begin{equation}
\frac{d}{dt} \int_{\Omega(t)} q \, dV = \int_{\Omega(t)} \frac{\partial q}{\partial t} \, dV + \oint_{\partial \Omega(t)} (n \cdot u_s) q \, dS,
\end{equation}

in which $u_s$ is the velocity and $n$ is the outward pointing unit normal vector of the boundary $\partial \Omega(t)$. Applying the RTT to Eq. (5.15) yields the expression

\begin{equation}
\mathcal{F}(t) = -\int_{V(t)} \rho \frac{\partial u}{\partial t} \, dV - \oint_{\partial V(t)} (n \cdot u_s) \rho u \, dS
+ \oint_{S_{CV}(t)} n \cdot [-p I - (u - u_s) \rho u + T] \, dS - \oint_{S_b(t)} n \cdot (u - u_s) \rho u \, dS,
\end{equation}

Recall that $V(t) = V_{CV}(t) \setminus V_b(t)$, where $V_{CV}(t)$ is the entire control volume and contains the body domain $V_b(t)$; $S_{CV}(t) = \partial V_{CV}(t)$ is the boundary of the CV, and $S_b(t) = \partial V_b(t)$. Hence, $\partial V(t) = S_{CV}(t) \cup S_b(t)$, and the second integral can be split into two boundary integrals

\begin{equation}
\mathcal{F}(t) = -\int_{V(t)} \rho \frac{\partial u}{\partial t} \, dV
- \oint_{S_{CV}(t)} (n \cdot u_s) \rho u \, dS + \oint_{S_{CV}(t)} n \cdot [-p I - (u - u_s) \rho u + T] \, dS
- \oint_{S_b(t)} (n \cdot u_s) \rho u \, dS - \oint_{S_b(t)} n \cdot (u - u_s) \rho u \, dS,
\end{equation}

which can be simplified to obtain

\begin{equation}
\mathcal{F}(t) = -\int_{V(t)} \rho \frac{\partial u}{\partial t} \, dV + \oint_{S_{CV}(t)} n \cdot [-p I - u \rho u + T] \, dS - \oint_{S_b(t)} n \cdot (u \rho u) \, dS.
\end{equation}

Recall that in Eq. (5.23), the unit normal vector points outward on $S_{CV}(t)$ and inward on $S_b(t)$ since the integral is considered with respect to the boundary of $V(t)$. Next,
notice that by definition \( V_{CV}(t) = V(t) \cup V_b(t) \) is the union of disjoint regions. Hence, the integral over \( V(t) \) can be split into \( \int_{V(t)} q \, dV = \left( \int_{V_{CV}(t)} - \int_{V_b(t)} \right) q \, dV \). Applying the split to Eq. (5.23) yields

\[
F(t) = -\int_{V_{CV}(t)} \rho \frac{\partial u}{\partial t} \, dV + \int_{V_b(t)} \rho \frac{\partial u}{\partial t} \, dV + \oint_{S_{CV}(t)} n \cdot [-pI - u \rho u + T] \, dS - \oint_{S_b(t)} n \cdot (u \rho u) \, dS.
\]  

(5.24)

Finally, we can apply the Reynolds transport theorem (Eq. (5.20)) to the integral over \( V_b(t) \) above. Letting \( \mathbf{N} \) be the outward pointing unit normal vector to \( S_b(t) \), we obtain

\[
F(t) = -\int_{V_{CV}(t)} \rho \frac{\partial u}{\partial t} \, dV + \frac{d}{dt} \int_{V_b(t)} \rho u \, dV + \oint_{S_{CV}(t)} n \cdot [-pI - u \rho u + T] \, dS - \oint_{S_b(t)} n \cdot (u \rho u) \, dS
\]

(5.25)

Substituting the fact that \( \mathbf{N} = -\mathbf{n} \) yields a general expression for the hydrodynamic force on an immersed body

\[
F(t) = -\int_{V_{CV}(t)} \rho \frac{\partial u}{\partial t} \, dV + \frac{d}{dt} \int_{V_b(t)} \rho u \, dV + \oint_{S_{CV}(t)} n \cdot [-pI - u \rho u + T] \, dS - \oint_{S_b(t)} n \cdot (u \rho u - u_S \rho u) \, dS.
\]  

(5.26)

Eq. (5.26) can be further simplified if we assume no-slip boundary conditions at the fluid-structure interface by setting \( u_S = u \), which gives

\[
F(t) = -\int_{V_{CV}(t)} \rho \frac{\partial u}{\partial t} \, dV + \frac{d}{dt} \int_{V_b(t)} \rho u \, dV + \oint_{S_{CV}(t)} n \cdot [-pI - u \rho u + T] \, dS
\]

(5.27)
This is the net hydrodynamic force expression as written in Eq. (5.16).

5.3.1.4. Derivation of the new hydrodynamic torque expression. Conservation of angular momentum for a material volume \( V_m(t) \) (a volume that moves with the local fluid velocity) that shares its boundary with an arbitrary moving control volume \( V(t) \) at time \( t \) can be written as [203, 204]

\[
\frac{d}{dt} \int_{V_m(t)} (r \wedge \rho u) \, dV = \oint_{\partial V_m(t) = \partial V(t)} r \wedge [n \cdot \Sigma] \, dS
\]

\[
(5.28) \quad = \oint_{S_{CV(t)}} r \wedge [n \cdot \Sigma] \, dS + \oint_{S_b(t)} r \wedge [n \cdot \Sigma] \, dS,
\]

in which \( r = x - x_0 \), with \( x_0 \) as a reference point for computing torques, and \( \Sigma = -p I + T \). Letting \( \mathcal{M}(t) \) be the torque exerted by the fluid on the body and noticing that \( \oint_{S_b(t)} r \wedge [n \cdot \Sigma] \, dS = -\mathcal{M}(t) \), we have

\[
(5.29) \quad \mathcal{M}(t) = -\frac{d}{dt} \int_{V_m(t)} (r \wedge \rho u) \, dV + \oint_{S_{CV(t)}} r \wedge [n \cdot \Sigma] \, dS.
\]

Using the RTT, the integral of an arbitrary quantity \( \Phi \) over the material volume \( V_m(t) \) can be related to integral over arbitrary volume \( V(t) \) with surface velocity moving with \( u_S \) as

\[
(5.30) \quad \frac{d}{dt} \int_{V_m(t)} \Phi \, dV = \frac{d}{dt} \int_{V(t)} \Phi \, dV + \oint_{\partial V_m(t) = \partial V(t)} n \cdot (u - u_S) \Phi \, dS.
\]
Using Eq. (5.30) with $\Phi = \mathbf{r} \wedge \rho \mathbf{u}$, the expression for torque becomes

$$
\mathcal{M}(t) = -\frac{d}{dt} \int_{V(t)} (\mathbf{r} \wedge \rho \mathbf{u}) \, dV + \oint_{S_{CV(t)}} \left[ \mathbf{r} \wedge (-p \mathbf{n} + \mathbf{n} \cdot \mathbf{T}) - \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_S)(\mathbf{r} \wedge \rho \mathbf{u}) \right] \, dS
$$

(5.31) $- \oint_{S_{b(t)}} \mathbf{n} \cdot (\mathbf{u} - \mathbf{u}_S)(\mathbf{r} \wedge \rho \mathbf{u}) \, dS.$

Finally, by manipulating the term derivative term in Eq. (5.31) using the RTT we get an expression for torque on an immersed body as

$$
\mathcal{M}(t) = -\int_{V_{CV(t)}} \rho \mathbf{r} \wedge \frac{\partial \mathbf{u}}{\partial t} \, dV + \frac{d}{dt} \int_{V_b(t)} \rho (\mathbf{r} \wedge \mathbf{u}) \, dV
$$

$$
+ \oint_{S_{CV(t)}} \left[ \mathbf{r} \wedge (-p \mathbf{n} + \mathbf{n} \cdot \mathbf{T}) - \mathbf{n} \cdot \mathbf{u} \rho (\mathbf{r} \wedge \mathbf{u}) \right] \, dS.
$$

5.3.2. Numerical integration

We use Riemann summation to evaluate the various integrals in Eqs. (5.16) and (5.19) over a moving rectangular control volume. Fig. 5.2 shows the rectangular control volume marked by its lower and upper coordinates $(x_L, y_L)$ and $(x_U, y_U)$, respectively.

The arbitrary surface velocity $\mathbf{u}_S$ is chosen such that the moving CV is forced to align with Cartesian grid faces. This greatly simplifies the evaluation of various terms inside the force and torque integrals. The linear and angular momentum integrals over $V_b(t)$ are evaluated in the Lagrangian frame, whereas the rest are computed in the Eulerian frame. The details of these computations are given below.

For notational simplicity, we present the discretized equations in two spatial dimensions. An extension to three spatial dimensions is straightforward. A discrete grid covers the physical domain $\Omega$ with mesh spacing $\Delta x$ and $\Delta y$ in each direction. The position
Figure 5.2. A staggered grid spatial discretization. The $x$-velocity component $u$ is solved for at locations depicted with horizontal arrows. The $y$-velocity component $v$ is solved for at locations depicted with vertical arrows. The pressure $p$ is solved for at locations depicted with a solid black dots. The corners $(x_L, y_L)$ and $(x_U, y_U)$ define the control volume $V_{CV}(t)$, which is shaded in red. The dashed line represents $S_{CV}(t) = \partial V_{CV}(t) = L \cup T \cup R \cup B$.

of each grid cell center is given by $x_{i,j} = (x_{i,j}, y_{i,j})$. For a given cell center, $x_{i-\frac{1}{2},j}$ denotes the physical location of the cell face that is half a grid space away from $x_{i,j}$ in the negative $x$-direction, i.e. $x_{i-\frac{1}{2},j} = (x_{i,j} - \frac{\Delta x}{2}, y_{i,j})$. Similarly $x_{i,j-\frac{1}{2}}$ denotes the physical location of the cell face that is half a grid cell away from $x_{i,j}$ in the negative $y$-direction, i.e. $x_{i,j-\frac{1}{2}} = (x_{i,j}, y_{i,j} - \frac{\Delta y}{2})$. The discrete approximations described here are also valid when adaptive mesh refinement is used, although volume weights need to be appropriately modified for different velocity components.

Let $t^n$ be the time at time step $n$. After stepping forward from time $t^n$ to $t^{n+1} = t^n + \Delta t$, a pressure solution is obtained at cell centers $p_{i,j}^{n+1} = p(x_{i,j}, t^{n+1})$, while velocity components are obtained at cell faces: $u_{i-\frac{1}{2},j}^{n+1} = u(x_{i-\frac{1}{2},j}, t^{n+1})$ and $v_{i,j-\frac{1}{2}}^{n+1} = v(x_{i,j-\frac{1}{2}}, t^{n+1})$. 
These are the only Eulerian quantities needed to evaluate the discrete approximations to Eqs. (5.16) and (5.19). Only rectangular control volumes are considered in the present work. A CV is described by its lower left and upper right corners: letting \((x_L, y_L)\) and \((x_U, y_U)\) denote the lower and upper corners respectively, the control volume is defined to be the Cartesian product of intervals \(V_{CV}(t) = \{ x \in \Omega \mid x \in [x_L, x_U] \times [y_L, y_U] \}\). Moreover, \(S_{CV}(t)\) is forced to remain on grid lines and it not allowed to cross into the interior of grid cells. This greatly simplifies the required numerical approximations. Refer to Fig. 5.2 for a sketch of the control volume configuration over a staggered mesh discretization. Let the control volume and surface at a time instance \(t^{n+1}\) be denoted by \(V_{CV}^{n+1} = V_{CV}(t^{n+1})\) and \(S_{CV}^{n+1} = S_{CV}(t^{n+1})\), respectively.

5.3.2.1. Discrete approximation to surface integrals. The control surface \(S_{CV}^{n+1}\) is composed of four segments (eight faces in 3D) denoted by \(B, L, T,\) and \(R\) in Fig. 5.2. Consequently, computing surface normals on each of these segments is simple, e.g for the bottom segment \(B^{n+1}, n = -e_y\). The discretized surface integral of a quantity \(\Phi\) over \(S_{CV}^{n+1}\) is simply the sum over these four segments

\[
(5.33) \quad \oint_{S_{CV}^{n+1}} n \cdot \Phi \, dS = \oint_{R^{n+1}} e_x \cdot \Phi \, dS - \oint_{L^{n+1}} e_x \cdot \Phi \, dS + \oint_{T^{n+1}} e_y \cdot \Phi \, dS - \oint_{B^{n+1}} e_y \cdot \Phi \, dS.
\]

Moreover, it is sufficient to show the discrete approximation to the surface integral over a single segment since the contribution from the other three segments are computed
analogously. Over the bottom surface $B$, the discretization of each term is given by

$$\oint_B \mathbf{n} \cdot (-p \mathbf{I}) \, dS = \oint_B -e_y (-p) \, dS$$

$$\approx \sum_{(i,j-\frac{1}{2}) \in B} -e_y \frac{(p_{i,j} + p_{i,j-1})}{2} \Delta x,$$  

(5.34)

$$\oint_B \mathbf{n} \cdot (-u \rho u) \, dS = -\rho \oint_B -v (u e_x + v e_y) \, dS$$

$$\approx -\rho \sum_{(i,j-\frac{1}{2}) \in B} -v_{i,j-\frac{1}{2}} \left[ \frac{u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j-1} + u_{i+\frac{1}{2},j-1}}{4} e_x + v_{i,j-\frac{1}{2}} e_y \right] \Delta x,$$  

(5.35)

$$\oint_B \mathbf{n} \cdot (\nabla u + (u u^T)) \, dS = \mu \oint_B \left[ \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) e_x + 2 \frac{\partial v}{\partial y} e_y \right] \, dS$$

$$\approx \mu \sum_{(i,j-\frac{1}{2}) \in B} \left[ \left( \frac{u_{i+\frac{1}{2},j-1} - u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j-1} - u_{i-\frac{1}{2},j}}{2\Delta y} + \frac{v_{i+1,j-\frac{1}{2}} - v_{i-1,j-\frac{1}{2}}}{2\Delta x} \right) e_x \right] \Delta x$$

$$+ \mu \sum_{(i,j-\frac{1}{2}) \in B} \left[ 2 \frac{v_{i+\frac{1}{2},j} - v_{i,j-\frac{3}{2}}}{2\Delta y} e_y \right] \Delta x.$$  

(5.36)

Fig. 5.3 shows a schematic of the pressure and velocity values required to evaluate Eqs. (5.34), (5.35), and (5.36). Evaluating the surface integral on $B$ in the torque calculation about a point $x_0$ is done by computing $r_{i,j-\frac{1}{2}} = x_{i,j-\frac{1}{2}} - x_0$ and evaluating the cross product between $r_{i,j-\frac{1}{2}}$ and the integrand.
5.3.2.2. Change in control volume momentum. A discretization of the the time derivative term in Eq. (5.16) for the integral over $V_{CV}(t)$ at time step $n + 1$ is given by

$$ - \int_{V_{CV}^{n+1}} \rho \frac{\partial \mathbf{u}}{\partial t} \, dV \approx - \int_{V_{CV}^{n+1}} \rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \, dV = - \frac{1}{\Delta t} \int_{V_{CV}^{n+1}} \rho \mathbf{u}^{n+1} \, dV + \frac{1}{\Delta t} \int_{V_{CV}^{n+1}} \rho \mathbf{u}^n \, dV, $$

where a discrete approximation to the total linear momentum within $V_{CV}^{n+1}$ can be written as

$$ \int_{V_{CV}^{n+1}} \rho \mathbf{u} \, dV \approx e_x \sum_{(i - \frac{1}{2}, j) \in V_{CV}^{n+1}} \rho u_{i - \frac{1}{2}, j} \Delta V_{i - \frac{1}{2}, j} + e_y \sum_{(i, j - \frac{1}{2}) \in V_{CV}^{n+1}} \rho v_{i, j - \frac{1}{2}} \Delta V_{i, j - \frac{1}{2}}. $$

Here, $\Delta V = \frac{\Delta x \Delta y}{2}$ when either $(i - \frac{1}{2}, j) \in S_{CV}^{n+1}$ or $(i, j - \frac{1}{2}) \in S_{CV}^{n+1}$, and $\Delta V = \Delta x \Delta y$ otherwise, to ensure that $\sum_{(i - \frac{1}{2}, j)} \Delta V_{i - \frac{1}{2}, j} = \sum_{(i, j - \frac{1}{2})} \Delta V_{i, j - \frac{1}{2}} = |V_{CV}^{n+1}|$, the volume of the CV.
In the original hydrodynamic force formula Eq. (5.15) introduced by Noca [205], a
discretization of the momentum term is given by

\[
\frac{\partial}{\partial t} \int_{V_{CV}^{n+1}} \rho \mathbf{u} \, dV \approx -\frac{1}{\Delta t} \int_{V_{CV}^{n+1}} \rho \mathbf{u}^{n+1} \, dV + \frac{1}{\Delta t} \int_{V_{CV}^{n}} \rho \mathbf{u}^{n} \, dV.
\]

Notice that Eqs. (5.37) and (5.39) are nearly identical, although the former only requires
an evaluation over a single CV, while the latter requires an evaluation over two time-lagged
CVs.

The analogous term in the torque calculation Eq. (5.19) is discretized differently. Each
velocity location \( \mathbf{x}_{i-\frac{1}{2},j} \) is looped over and an approximation to \( \nu \) is computed. Then
\( \mathbf{r}_{i-\frac{1}{2},j} = \mathbf{x}_{i-\frac{1}{2}} - \mathbf{x}_0 \) is computed and used in the cross product. Mathematically, this is
realized as

\[
-\int_{V_{CV}^{n+1}} \rho \mathbf{r} \wedge \frac{\partial \mathbf{u}}{\partial t} \, dV \approx -\frac{1}{\Delta t} \int_{V_{CV}^{n+1}} \rho \mathbf{r} \wedge \mathbf{u}^{n+1} \, dV + \frac{1}{\Delta t} \int_{V_{CV}^{n+1}} \rho \mathbf{r} \wedge \mathbf{u}^{n} \, dV,
\]

in which

\[
\int_{V_{CV}^{n+1}} \rho \mathbf{r} \wedge \mathbf{u} \, dV \approx \sum_{(i-\frac{1}{2},j) \in V_{CV}^{n+1}} \mathbf{r}_{i-\frac{1}{2},j} \wedge \left[ \mathbf{u}_{i-\frac{1}{2},j} \right] dV \\
\sum_{(i-\frac{1}{2},j) \in V_{CV}^{n+1}} \mathbf{r}_{i-\frac{1}{2},j} \wedge \left[ \mathbf{u}_{i-\frac{1}{2},j} \right] dV \approx \sum_{(i-\frac{1}{2},j) \in V_{CV}^{n+1}} \left[ \mathbf{v}_{i-1,j-\frac{1}{2}} + \mathbf{v}_{i,j-\frac{1}{2}} + \mathbf{v}_{i-1,j+\frac{1}{2}} + \mathbf{v}_{i,j+\frac{1}{2}} \right] dV \\
5.3.2.3. Change in body momentum. The final term that needs to be discretely
approximated is the change in momentum of the immersed body. This is presented as an
integral of the body’s velocity over the region \( V_b(t) \subset \Omega \) in an Eulerian reference frame.
Since the body’s position and velocity are described in a Lagrangian reference frame over
a region \( I_b \subset U \), it is generally much easier to evaluate the Lagrangian form of this integral
instead. Using the definition of $\delta(x)$, it can be shown that the momentum over these two different reference frames are equivalent:

\begin{equation}
\int_{V_b} \rho u(x, t) dx = \int_{I_b} \rho U(s, t) ds.
\end{equation}

Letting $\mathcal{G}^n$ denote the collection of discrete IB points corresponding to the region $I_b$ at time step $n$, the object’s momentum is obtained by

\begin{equation}
P^n_b = \sum_{(l,m) \in \mathcal{G}^n} \rho U^n_{l,m} \Delta s_{l,m},
\end{equation}

in which $U^n_{l,m}$ denotes the velocity of IB node $(l, m)$ at time step $n$, and $\Delta s_{l,m}$ denotes the discrete volume occupied by the node. The change in momentum required for the evaluation of hydrodynamic forces is then given by

\begin{equation}
\frac{d}{dt} \int_{I_b} \rho U(s, t) ds \approx \frac{P^n_{b+1} - P^n_b}{\Delta t}.
\end{equation}

The change in the body’s angular momentum for the torque calculation is done similarly.

### 5.3.3. Lagrange multiplier method

The hydrodynamic force and torque on an immersed body can also be computed in an extrinsic manner from force/torque balance laws. Specifically, for an immersed body occupying volume $V_b(t)$, the force and torque balance laws read as

\begin{equation}
\frac{d}{dt} \int_{V_b(t)} \rho U \, dV = \mathcal{F} + \int_{V_b(t)} F \, dV,
\end{equation}

\begin{equation}
\frac{d}{dt} \int_{V_b(t)} \rho \mathbf{R} \wedge U \, dV = \mathcal{M} + \int_{V_b(t)} \mathbf{R} \wedge F \, dV,
\end{equation}
in which \( \mathbf{R} = \mathbf{X} - \mathbf{X}_0 \) is the radius vector from reference point \( \mathbf{X}_0 \) to Lagrangian node position \( \mathbf{X} \), and \( \mathbf{F} \) is the Lagrange multiplier imposing rigidity constraint as defined in Eq. (5.3). Note that for the IB method, the net hydrodynamic force \( \mathbf{F} \) and torque \( \mathbf{M} \) on the body can be readily evaluated in the Lagrangian frame as a part of the solution process without computing any extra terms. We have used this approach in a previous work [13]. We will compare results obtained from both moving CV and LM method in Sec. 5.5.

### 5.3.4. Equivalence of the two methods

Although Eqs. (5.16) and (5.45) look different, we now show that the use of (regularized) delta functions in the IB method make them equivalent expressions. To prove this, consider a single body in the domain \( \Omega \), occupying a region of space \( V_b(t) \subset \Omega \). Let \( \mathbf{F}(s, t) \) denote the Lagrange multiplier field defined on Lagrangian nodes, and \( f(x, t) \) denote its Eulerian counterpart. Here, \( s \in U \) and \( U \subset \mathbb{R}^d \). With continuous and Peskin’s discrete delta functions, the following identity holds

\[
\int_\Omega f(x, t) \, dx = \int_\Omega \left[ \int_{U \subset \Omega} \mathbf{F}(s, t) \delta(x - X(s, t)) \, ds \right] \, dx \\
= \int_{U \subset \Omega} \left[ \int_\Omega \mathbf{F}(s, t) \delta(x - X(s, t)) \, dx \right] \, ds \\
= \int_{U \subset \Omega} \mathbf{F}(s, t) \left[ \int_\Omega \delta(x - X(s, t)) \, dx \right] \, ds \\
= \int_U \mathbf{F}(s, t) \, ds.
\]

(5.47)
The above expression is the equivalence of Lagrangian and Eulerian force densities and is a well known result \([5, 209]\). Starting with the momentum Eq. (5.1) in Eulerian form

\[ \rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \cdot (\mathbf{u} \mathbf{u}) = \nabla \cdot (-p \mathbf{I} + \mathbf{T}) + \mathbf{f}, \]

and integrating it over \(V_{CV}(t)\), we obtain

\[ \int_{V_{CV}(t)} \rho \frac{\partial \mathbf{u}}{\partial t} \, dV + \int_{V_{CV}(t)} \rho \nabla \cdot (\mathbf{u} \mathbf{u}) \, dV = \int_{V_{CV}(t)} \nabla \cdot [-p \mathbf{I} + \mathbf{T}] \, dV + \int_{V_{CV}(t)} \mathbf{f} \, dV. \]

Applying the divergence theorem to terms with \(\nabla \cdot\), and using Eq. (5.47) we obtain

\[ \int_{V_{CV}(t)} \mathbf{f} \, dV = \int_{V_{b}(t)} \mathbf{F} \, dV = \int_{V_{CV}(t)} \rho \frac{\partial \mathbf{u}}{\partial t} \, dV - \int_{S_{CV}(t)} \mathbf{n} \cdot [-p \mathbf{I} - \rho \mathbf{u} \mathbf{u} + \mathbf{T}] \, dS. \]

Using the above expression for \(\int_{V_{b}(t)} \mathbf{F} \, dV\) with Eq. (5.45) we get force expression (5.16).

Similarly, one can prove the equivalence of torque expressions (5.19) and (5.46) by noting that \(\mathbf{r} \wedge \nabla \cdot \mathbf{S} = \nabla \cdot (\mathbf{r} \wedge \mathbf{S})\) for any symmetric tensor satisfying \(\mathbf{S} = \mathbf{S}^T\).

When multiple bodies exist in the domain, however, there is a subtle difference in the LM and moving CV expressions. The LM expressions in the form of Eqs. (5.45) and (5.46) are restricted to individual bodies; therefore in the presence of multiple bodies, \(\mathbf{F}\) and \(\mathbf{M}\) can be computed separately for each body. For the moving CV method, care must be taken to restrict the control volume to a particular body, i.e. it should not enclose other bodies in its vicinity. Otherwise, \(\mathbf{F}\) and \(\mathbf{M}\) will contain contribution from multiple bodies.

We explore this subtlety with the moving CV method by taking flow past two stationary cylinders in Sec. 5.5.1.3 and drafting-kissing-tumbling of two sedimenting cylinders in Sec. 5.5.5. For an AMR framework, the boundary of a control volume can span multiple
refinement levels. When velocity is reconstructed from old hierarchy to new, there is generally no guarantee that momentum is conserved because of the inter-level velocity interpolation. This can lead to jumps in $F$ and $M$ because of the time derivative terms. Eqs. (5.45) and (5.46) are restricted to $V_b(t)$, which generally lies on the finest grid level. Therefore, one can expect force and torque calculations to be relatively insensitive to velocity reconstruction operations, which mostly requires intra-level interpolation. We explore this issue with translating plate example in Sec. 5.5.3.2.

5.4. Software implementation

We use the IBAMR library [14] to implement the moving control volume method in this work for our numerical tests. IBAMR has built-in support for direct forcing and fully constrained IB methods, among other variants of the IB method. IBAMR relies on SAMRAI [15, 16] for Cartesian grid management and the AMR framework. Solver support in IBAMR is provided by PETSc library [17, 18, 19].

5.5. Results

5.5.1. Flow past cylinder

In this section we validate our moving control volume method and Lagrange multiplier method for computing hydrodynamic forces and torques on immersed bodies.

5.5.1.1. Stationary cylinder. We first consider the flow past a stationary circular cylinder. The cylinder has diameter $D = 1$ and is placed in a flow with far-field velocity $U_\infty = (U_\infty, V_\infty) = (1, 0)$. The computational domain is a rectangular channel taken to be of size $18D \times 12D$, with the center of the cylinder placed at $(x, y) = (0, 0)$. The domain
Figure 5.4. Comparison of the drag coefficient for flow past a cylinder at \( Re = 550 \) measured in two different ways. (—, red): Control volume using Eq. (5.16); (---, black): Lagrange multiplier using Eq. (5.45).

is discretized by a uniform Cartesian mesh of size 900 \( \times \) 600. The aerodynamic drag coefficient \( C_D = \mathbf{F} \cdot \mathbf{e}_x / (\rho D \| U_\infty \|^2 / 2) \) is calculated numerically in two different ways: via Eq. (5.16) and via integrating Lagrange multipliers enforcing the rigidity constraint on the cylinder (Eq. (5.45)). The control volume is taken to be \([-D, 1.5D] \times [-D, D]\) and does not move from its initial location. Note that in the case where both the body and the control volume are stationary, Eqs. (5.15) and (5.16) are equivalent and give the same numerical solution. The density is set to \( \rho = 1 \) and the Reynolds number of the flow is \( Re = \rho U_\infty D / \mu = 550 \). This problem has been studied numerically by Bergmann and Iollo [207] and by Ploumhans and Winckelmans [219]. The temporal behavior of \( C_D \) matches well with the previous studies [207, 219].
5.5.1.2. Translating cylinder. Next, we consider the case of a circular cylinder translating with prescribed motion. The parameters used in this case are identical to those of the flow past a stationary cylinder case, except now \( \mathbf{U}_\infty = (U_\infty, V_\infty) = (0, 0) \). The cylinder is dragged with speed \( \mathbf{U}_b = (U_b, V_b) = (-1, 0) \). The aerodynamic drag coefficient is computed as \( C_D = \mathcal{F} \cdot \mathbf{e}_x / (\rho D \| \mathbf{U}_b \|^2 / 2) \). The control volume is initially set to \([-D, 1.5D] \times [-D, D]\) and translates to the left every few time steps to ensure that it always contains the cylinder. The density is set to \( \rho = 1 \) and the Reynolds number is \( \text{Re} = \rho U_b D / \mu = 550 \). Periodic boundary conditions are used on all faces of the computational domain. This problem was also studied numerically by Bergmann and Iollo [207].

Fig. 5.5 shows the time evolution of \( C_D \) calculated in three different ways, via Eqs. (5.15) and (5.16), and by integrating Lagrange multipliers. The present control volume method matches well with the Lagrange multiplier approach. However, spurious jumps in drag coefficient are seen for the original control volume method outlined by Noca. These spurious oscillations are also present in the computation done by Bergmann and Iollo [207]. We remark that the oscillations seen by Bergmann and Iollo [207] are quantitatively different than the ones presented here in Fig. 5.5 for comparison, although both use Eq. (5.15) to compute hydrodynamic force. This can be attributed to differences in the numerical method used to impose constraint: we use a constraint-based immersed boundary method whereas Bergmann and Iollo use a Brinkman penalization method.

5.5.1.3. Two stationary cylinders. To study the effect of control volume size in the presence of multiple bodies, we consider the case of two stationary circular cylinders each with \( D = 1 \) and placed in a flow with far-field velocity \( \mathbf{U}_\infty = (U_\infty, V_\infty) = (1, 0) \). The
Figure 5.5. Comparison of the drag coefficient $C_D$ a translating cylinder at $Re = 550$ measured in three different ways.

- Control volume using Eq. (5.15);
- Control volume using Eq. (5.16);
- Lagrange multiplier.

Note that LM and present moving CV curves are on top of each other.

Bottom and top cylinders are centered about position $(x, y) = (0, -2D)$ and $(x, y) = (0, 2D)$, respectively within a rectangular channel of size $18D \times 12D$. The domain is discretized by a uniform Cartesian mesh of size $900 \times 600$ and the Reynolds number is $Re = \rho U_\infty D/\mu = 550$ for each cylinder. The density of the fluid is set to $\rho = 1$.

Four different (but symmetric) control volume configurations are considered:

1. Two disjoint CVs located at $[-D, D] \times [-3D, -D]$ and $[-D, D] \times [D, 3D]$.
2. Two CVs located at $[-D, D] \times [-3D, D]$ and $[-D, D] \times [-D, 3D]$ that slightly overlap, but do not intersect the other cylinder.
3. Two CVs located at $[-D, D] \times [-3D, 2D]$ and $[-D, D] \times [-2D, 3D]$, where each CV holds one full cylinder and half of the second cylinder.
Two CVs located at $[-D, D] \times [-3D, 2.7D]$ and $[-D, D] \times [-2.7D, 3D]$, where each CV contains both cylinders.

Fig. 5.6 shows flow visualizations for the four different CV configurations. Fig. 5.7 shows the drag coefficient over time for each of the four configurations. In the case where the CVs do not overlap (Figs. 5.6(a) and 5.7(a)), or when they overlap but do not enclose multiple bodies either partially or fully (Figs. 5.6(b) and 5.7(b)) the drag coefficients calculated for the top and bottom cylinders are close to the drag coefficient calculated for a single cylinder considered in Section 5.5.1.1.

The effect of resizing the control volumes to partially or fully contain other objects is seen in the hydrodynamic drag force measurement. In the case where each CV contains one and a half cylinders (Figs. 5.6(c) and 5.7(c)), the drag coefficient deviates significantly from the drag coefficient calculated for a single cylinder. Rather, the computed force is the drag on a combined full and a half cylinder contained within the CV, which is approximately 1.5 times the drag on a single cylinder. In the case where each CV contains both cylinders (Figs. 5.6(d) and 5.7(d)), the measured $C_D$ in each control volume is approximately twice the $C_D$ measured on a single cylinder. Therefore, in presence of multiple bodies in the domain, care must be taken to restrict the CV to an individual body.

5.5.2. Oscillating cylinder

In this section we consider cases from previous studies, some of which have reported spurious force oscillations in hydrodynamic drag and lift forces with IB methods. We
Figure 5.6. Vorticity generated by a circular cylinder at $t = 5$ for $Re = 550$: (a) two disjoint control volumes; (b) two slightly overlapping control volumes; (c) each control volume contains one and a half cylinders; (d) each control volume contains both cylinders. All figures are plotted for vorticity between $-20$ and $20$.

do not observe such spurious force oscillations using LM and CV methods within an immersed boundary framework.

5.5.2.1. **In-line oscillation.** We consider an in-line oscillation of a circular cylinder in a quiescent flow as done in Dütsch et al. [220] and Lee et al. [117]. The cylinder has a diameter $D = 1$ and is placed in a domain of size $[-16D, 16D] \times [-8D, 8D]$ with zero
velocity prescribed on all boundaries of the domain. The initial center of mass of the cylinder is placed at (0, 0) and its velocity is set to \( U_b = (-U_0 \cos(2\pi ft), 0) \), in which \( f \) is the frequency of oscillation. The Reynolds number of the flow is \( \text{Re} = \rho U_0 D/\mu = 100 \), and the Keulegan-Carpenter number is \( KC = U_0/(fD) = 5 \). The time period of oscillation of the cylinder is given by \( T = 1/f \). The density of the fluid is set to be \( \rho = 1 \). These parameters are chosen to match those reported in [117, 220].

Figure 5.7. Temporal evolution of drag coefficient for \( \text{Re} = 550 \): (a) two disjoint control volumes; (b) two slightly overlapping control volumes; (c) one and a half cylinders; (d) two full cylinders. —: Bottom CV; --- (green): Top CV.
Three levels of mesh refinement are used, with \( n_{\text{ref}} = 4 \) between each level. The cylinder is embedded in the finest mesh level at all time instances. At the coarsest level, three different mesh sizes are used: \( 50 \times 25, 100 \times 50, \) and \( 200 \times 100 \), which corresponds to finest grid spacings of \( \Delta x_{\min} = \Delta y_{\min} = 0.04D, \Delta x_{\min} = \Delta y_{\min} = 0.02D, \) and \( \Delta x_{\min} = \Delta y_{\min} = 0.01D \), respectively. The computational time step size is chosen to be \( \Delta t = 0.005D/U_0 \), matching that of Lee et al. \[117\]. This time step size satisfies the convective CFL condition with \( C = 0.7 \), which is found to be stable for all the mesh sizes considered here. A stationary control volume is placed at \([-4D, 4D] \times [-2D, 2D]\) in order to contain the entire cylinder at all time instances. The drag coefficient is computed as \( C_D = \mathcal{F} \cdot e_x/(0.5\rho U_0^2 D) \).

Fig. 5.8 shows the time evolution of drag coefficient for the oscillating cylinder. Both CV and LM approaches yield identical results. Even our coarse resolution results are in good agreement with the high resolution results of Lee et al. \[117\]. Moreover, Lee et al. conducted this case within an immersed boundary framework and observed large spurious force oscillations (see Fig. 14 in \[117\]) at coarse grid resolutions. They compute drag on the immersed body by evaluating pressure and velocity gradients within the body region \( V_b(t) \). Only at fine grid resolutions, where the spatial gradients are more accurate, were they able to suppress the spurious force oscillations. Since we do not require spatial gradients of velocity and pressure within the body region \( V_b(t) \) with our approach, we do not observe such spurious oscillations.

5.5.2.2. Cross-flow oscillation. Here we consider a cylinder oscillating in the transverse direction with an imposed cross-flow. The cylinder has a diameter \( D = 1 \) and its initial center of mass is placed at \((x, y) = (0, 0.2D)\). The cylinder is placed in a domain
Figure 5.8. Temporal evolution of the drag coefficient of an oscillating cylinder at $Re = 100$ measured by the (a) present control volume approach and (b) Lagrange multiplier approach. Here, (---, black): $\Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.04D$; (---, red): $\Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.02D$; (---, blue): $\Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.01D$. of size $[-5D, 27D] \times [-8D, 8D]$ and oscillates in the transverse direction with velocity $U_b = (0, -V_0 \sin(2\pi f_e t))$, where $f_e$ is the frequency of oscillation. An axial free-stream velocity $U_\infty$ is set at the inlet, top, and bottom faces of the computational domain. The transverse traction components are set to zero on the top and bottom boundaries, and the axial and transverse tractions are set to zero at the outflow boundary.

The density of the fluid is set to $\rho = 1$ and the Reynolds number based on the free-stream velocity is $Re = \rho U_\infty D/\mu = 185$. Letting $f_0 = 0.193 U_\infty / D$ be the natural shedding frequency for a stationary cylinder, we set $f_e / f_0 = 1$. The maximum oscillation velocity of the cylinder is taken to be $0.159 V_0 / (f_e D) = 0.2$. This case is considered in Lee et al. [117] and Guilmineau and Queutey [221]. A stationary CV is placed at $[-D, D] \times [-2D, 2D]$, which contains the cylinder at all time instances (see Fig. 5.9).

The domain is discretized with three different uniform meshes of sizes $400 \times 192$, $800 \times 384$, and $1600 \times 768$, which corresponds to grid spacings of $\Delta x = \Delta y = 0.08D$, 
\[ \Delta x = \Delta y = 0.04D, \text{ and } \Delta x = \Delta y = 0.02D, \text{ respectively.} \]

A constant time step size of \( \Delta t = 0.005D/U_\infty \) is used for all the computations, matching that of Lee at al. \[117\]. The time step size satisfies the convective CFL condition with \( C = 0.7 \), which is found to be stable for all the mesh sizes considered here. Fig. 5.10 shows the time evolution of drag coefficient \( C_D = \mathbf{F} \cdot \mathbf{e}_x/(0.5\rho U_\infty^2 D) \). Again, both the CV and LM approach produce identical results and do not incur spurious force oscillations even at coarse resolutions as observed by Lee et al. (see Fig. 16 in \[117\]).

5.5.2.3. Rotational oscillation. As a last example of this section, we consider the case of a cylinder undergoing a rotational oscillation about its center of mass in a quiescent flow. The diameter of the cylinder is taken to be \( D = 1 \) and is placed in a domain of size \([-20D, 20D] \times [-20D, 20D]\), with zero velocity prescribed on all computational boundaries. The initial center of mass of the cylinder is placed at \((x, y) = (0, 0)\) and it rotates about its center with a velocity \( \omega_b = A_m \sin(2\pi ft) \), in which \( f \) is the frequency of oscillation and \( T = 1/f \) is the time period of oscillation. The density of the fluid is set to be \( \rho = 1 \). The Reynolds number of the flow is \( \text{Re} = \rho U_m D/\mu = 300 \), in which
$U_m = A_m D/2$. The cylinder rotates with frequency $f = 0.1$ and has maximum angular velocity $A_m = 10 f D$. These parameters are chosen to match Borazjani et al. [113].

The time step size is chosen to be $\Delta t = 1 \times 10^{-4} T$ and the grid is discretized by a two level mesh, which consists of a coarse mesh of size $512 \times 512$, and an embedded fine mesh with refinement ratio $n_{\text{ref}} = 4$. The minimum grid spacing at the finest level is $\Delta x_{\text{min}} = \Delta y_{\text{min}} = 0.0195 D$. The structure remains on the finest grid level for all time instances. A stationary control volume is placed at $[-1.01562 D, 1.01562 D]^2$ and the torque coefficient is computed as $C_T = \mathbf{M} \cdot \mathbf{e}_z/(0.5 \rho U_m^2 D)$. Fig. 5.11 shows the temporal evolution of $C_T$, which is in excellent agreement with sharp-interface CURVIB method results of Borazjani et al. [113].
5.5.3. Moving plate

In this section we consider the effect of regridding on hydrodynamic force calculations using the control volume approach. In the context of an immersed body AMR framework, regridding occurs when the body has moved some distance, or when new flow features of interest have appeared in the computational domain that require additional mesh refinement to resolve them adequately. We consider a moving plate example to understand the jumps in the drag coefficient due to regridding, and to provide some strategies to mitigate them.

5.5.3.1. Moving plate on a uniform mesh. First we consider a single level domain case in two spatial dimensions. A finite plate of height $b = 1$ is dragged perpendicular to
itself with constant velocity $\mathbf{U}_b = (U_b, V_b) = (-1, 0)$ in an infinite fluid at rest. The plate is modeled as a thin line of points, separated by grid cell size distance in the transverse direction. The physical domain is a periodic box of dimension $32b \times 22b$ and the domain is discretized by a uniform Cartesian grid of size $1024 \times 1024$. The initial location of the control volume is $[-2b, 2b] \times [-b, b]$, and it moves with an arbitrary speed to enclose the plate at all time instances. The density of the fluid is set to be $\rho = 1$. The Reynolds number of the flow is $\text{Re} = \rho U_b b / \mu = 20$. The drag coefficient is calculated as $C_D = \mathbf{F} \cdot \mathbf{e}_x / (\rho U_b^2 b / 2)$. An asymptotic solution $C_D \approx 2.09$ was derived by Dennis et al. and this problem was also studied numerically by Bhalla et al.  

Fig. 5.12 shows the time evolution of drag coefficient for the moving plate. We see that the numerical solution obtained by the moving CV computation matches well with the asymptotic value derived in and does not contains any spurious force oscillations or jumps.

5.5.3.2. Moving plate on an adaptive mesh. Next, we consider the same moving plate example but with locally refined grids. The domain is discretized by a coarse grid of size $256 \times 256$. A more refined mesh immediately surrounds the plate with refinement ratio $n_{\text{ref}} = 4$, giving the finest level an equivalent grid size of $1024 \times 1024$. The mesh is adaptive in the sense that it selectively refines in areas with large velocity gradients and where the immersed body is located in the domain. Apart from the locally refined grids, we use the same parameters of Sec. 5.5.3.1 for this case.

Three different control volumes are used. First, the CV is set initially to $[-2b, 2b] \times [-1.03125b, 1.03125b]$ and it translates to the right along with the plate. In this configuration, the CV spans multiple levels of the grid hierarchy. Fig. 5.13(b) shows the
Figure 5.12. Comparison of the drag coefficient for a translating plate at $\text{Re} = 20$ measured by the present moving control volume approach (—, blue); ---: asymptotic value from Dennis et al. [222].

measured drag coefficient over time. Again, $C_D$ evolves towards the asymptotic value derived in [222]. However, there are small jumps in $C_D$ over time, which are absent from the uniform mesh case of Sec. 5.5.3.1. These jumps can be attributed to the mesh hierarchy regridding to follow the moving plate or due to the moving CV itself. To rule out the possibility of jumps due to the motion of control volume, we consider a second CV configuration in which the CV is held stationary all times. The stationary CV again spans multiple grid levels and is big enough to contain the moving plate at all time instances. Fig. 5.14 shows the CV configuration and measured drag coefficient over time for this CV $= [-2b, -1.03125b] \times [20b, 1.03125b]$. The jumps in $C_D$ are again observed. In our third configuration, we limit the moving CV on the finest grid level. The CV configuration and
Figure 5.13. (a) Vorticity generated by a moving flat plate at $t = 7.5$ for $Re = 20$. The moving control volume spanning both the coarsest and finest grid level is shown in blue. The plotted vorticity is between $-7.5$ and $7.5$. (b) Temporal evolution of drag coefficient measured by the present moving control volume approach (–, blue); ---: asymptotic value from Dennis et al. [222].

$C_D$ temporal profile is shown in Fig. 5.15. The initial location of this control volume is $[-b, 0.625b] \times [-1.03125b, 1.03125b]$.

From Fig. 5.15(b), we note that the jumps due to regridding can be substantially mitigated if the moving CV is restricted to the finest mesh level. The CV translates to the right along with the plate, but remains on the finest mesh level throughout the simulation. There are no longer jumps in the computed drag values for $t \leq 10$.

The jumps in the hydrodynamic forces due to regridding occur due to velocity reconstruction following a regridding operation. A common velocity reconstruction strategy employed in an AMR framework is to first interpolate velocities from the coarser level to the new fine level, and then directly copy velocities from the old fine level in the spatial regions where old and new fine levels intersect. We refer readers to Griffith et al. [209] for details. When we restrict the moving CV to the finest grid level, we reduce errors.
Figure 5.14. (a) Vorticity generated by a moving flat plate at $t = 7.5$ for $Re = 20$. The stationary control volume spanning both the coarsest and finest grid level is shown in red. The plotted vorticity is between $-7.5$ and $7.5$. (b) Temporal evolution of drag coefficient measured by the present control volume approach (—, red); --: asymptotic value from Dennis et al. [222]

in momentum change due to velocity reconstruction from coarse to fine level interpolation. The contribution of $u^n$ in the time derivative term in Eqs. (5.16) and (5.19) can be evaluated either before or after regridding. In our empirical tests, we have observed that evaluating the time derivative term after regridding helps in mitigating the jumps even further (comparison data not shown). In all our results shown above, we evaluate the contribution from $u^n$ after regridding, i.e, using the old velocity at the new hierarchy configuration.
5.5.4. Swimming eel

In this section we demonstrate that the moving control volume approach can be used to determine the hydrodynamic forces and torques on a free-swimming body. We consider a two-dimensional undulating eel geometry, which is adapted from [109, 13]. The eel’s reference frame is aligned with the x-axis and in this reference frame the lateral displacement along $0 \leq x \leq L$ over its projected length $L$ is given by

$$y(x, t) = 0.125 \frac{x + 0.03125}{1.03125} \sin \left[2\pi(x - t/T)\right].$$

A backwards-traveling wave of the above form having a time period $T$ causes the eel to self-propel. The swimmer is taken to have a projected length $L = 1$, and time period $T = 1$. The Reynolds number based on $V_{\text{max}} = 0.785L/T$, the maximum undulation velocity at the tail tip, is $\text{Re} = \rho V_{\text{max}}L/\mu = 5609$. The undulations travel in the positive
$x$-direction, thereby propelling the eel in the negative $x$-direction. The density of the fluid is taken to be $\rho = 1$.

The eel’s total velocity $\mathbf{U}_b = \mathbf{U}_r + \mathbf{W}_r \times \mathbf{R} + \mathbf{U}_k$ in the Lagrangian frame is given by the following components: its rigid linear $\mathbf{U}_r$ and angular $\mathbf{W}_r$ center of mass velocities and its deformational velocity $\mathbf{U}_k$, which we assume to have zero net linear and angular momentum. The self-propulsion velocities are obtained using conservation of linear and angular momentum in the body domain $V_b(t)$

\begin{align}
M_b \mathbf{U}_r^{n+1} &= \int_{V_b(t)} \rho \left( \mathbf{J}_h [\mathbf{X}^{n+\frac{1}{2}}] \tilde{\mathbf{u}}^{n+1} \right) dV, \\
I_b \mathbf{W}_r^{n+1} &= \int_{V_b(t)} \rho \mathbf{R}^{n+\frac{1}{2}} \times \left( \mathbf{J}_h [\mathbf{X}^{n+\frac{1}{2}}] \tilde{\mathbf{u}}^{n+1} \right) dV,
\end{align}

in which $M_b$ and $I_b$ are the mass and moment of inertia tensor of the body, respectively. Having obtained these rigid body velocities, the body velocity at time $t^{n+1}$ required in Eq. (5.8) is obtained as $\mathbf{U}_b^{n+1} = \mathbf{U}_r^{n+1} + \mathbf{W}_r^{n+1} \times \mathbf{R}^{n+\frac{1}{2}} + \mathbf{U}_k^{n+1}$, in which $\mathbf{R}^{n+\frac{1}{2}} = \mathbf{X}^{n+\frac{1}{2}} - \mathbf{X}_0^{n+\frac{1}{2}}$ is the radius vector from (an estimated) midstep center of mass $\mathbf{X}_0^{n+\frac{1}{2}}$ to (an estimated) midstep Lagrangian node position $\mathbf{X}^{n+\frac{1}{2}}$. We refer readers to [13] for more details.

The fully periodic domain is taken to be of size $8L \times 4L$ and is discretized with a three-level hierarchy of Cartesian grids. The size of the coarsest grid is $128 \times 64$ grid cells and $n_{\text{ref}} = 4$ is taken for subsequent finer grids. Hence, the finest grid, with spacing equivalent to that of a uniform mesh of size $2048 \times 1024$, embeds the undulatory swimmer at all times. A time step size of $\Delta t = 1 \times 10^{-4} \, T$ is employed. The head of the swimmer is initially centered at $(x, y) = (0, 0)$ and its body extends in the positive $x$-direction. The
CV is initially located at $[-1.02L, 1.0425L] \times [-0.7075L, 0.73L]$, which encompasses the entire swimmer. The CV is allowed to span multiple grid levels. Whenever the eel’s center of mass translates a distance $\Delta x$, the CV moves with velocity $\mathbf{u}_S = (-\Delta x/\Delta t, 0)$ in order to remain aligned with the grid lines. Fig. 5.16 shows the vortical structures generated by the eel at four separate time instances, along with the locations of the moving control volume.

Fig. 5.17(a) shows the time evolution of axial ($U_r = U_r \cdot \mathbf{e}_x$) and lateral ($V_r = U_r \cdot \mathbf{e}_y$) swimming velocities, along with the rotational velocity $W^z_r = W_r \cdot \mathbf{e}_z$. The eel is shown to travel in the $-\mathbf{e}_x$ direction, eventually reaching a steady state speed. The angular velocity oscillates about a zero mean value, while the lateral velocity has small non-zero mean due
to initial transients. Fig. 5.17(b) shows the time evolution of net axial \((F_x = \mathbf{F} \cdot \mathbf{e}_x)\) and lateral \((F_y = \mathbf{F} \cdot \mathbf{e}_y)\) forces acting on the eel’s body, along with the net torque \(M_z = \mathbf{M} \cdot \mathbf{e}_z\), which is measured from the eel’s center of mass. The forces and torque are computed using the moving CV approach, although identical estimates are obtained from the LM approach (data not shown). Both \(\mathbf{F}\) and \(\mathbf{M}\) oscillate about a mean value of zero, which is expected during free-swimming as there are no external forces and torques applied on the swimmer.

For an object initially at rest in a periodic and quiescent fluid, the net linear momentum over the entire computational domain should remain zero \cite{13}, i.e., \(\mathbf{P}(t) = \int_\Omega \rho \mathbf{u} \, dV = 0\). Similarly, the net angular momentum of the system should also remain zero at all times, i.e., \(\mathbf{L}(t) = \int_\Omega \rho \mathbf{r} \wedge \mathbf{u} \, dV = 0\). In other words, all of the momentum generated due to the eel’s vortex shedding should be redistributed to the eel’s translational and rotational motion. Moreover, the change in linear momentum of the body should be equal to the net force on the body during free-swimming. Hence, the net force on the body should be given by \(\mathbf{F}(t) = \frac{d}{dt} \int_{V_b} \rho \mathbf{u} \, dV\), implying that

\[
(5.53) \quad \mathbf{I}(t) = -\int_{V_{CV}(t)} \rho \frac{\partial \mathbf{u}}{\partial t} \, dV + \oint_{S_{CV}(t)} \mathbf{n} \cdot [\mathbf{I} - \rho \mathbf{u} + \mathbf{T}] \, dS = 0.
\]

The above statement also implies the conservation of linear momentum for the control volume. Eq. (5.53) also implies that the sum of the Langrange multipliers enforcing the rigidity constraint within \(V_b(t)\) is zero during free-swimming (see Eq. (5.49)). Fig. 5.17(c) shows the temporal evolution of \(\mathbf{P}_x = \mathbf{P} \cdot \mathbf{e}_x\), \(\mathbf{P}_y = \mathbf{P} \cdot \mathbf{e}_y\), \(\mathbf{I}_x = \mathbf{I} \cdot \mathbf{e}_x\), and \(\mathbf{I}_y = \mathbf{I} \cdot \mathbf{e}_y\); it is indeed seen that these quantities are nearly zero for all time instances. The slight increase in \(\mathbf{P}_x\) is attributed to spatial and temporal discretization errors, whereas the jumps in \(\mathbf{I}_x\)
Figure 5.17. (a) Temporal evolution of (—, black) axial $U_r$ & (---, blue) lateral $V_r$ swimming velocity, and (---, red) rotational $W_z$ velocity of the eel calculated by Eqs. (5.51) and (5.52). (b) Temporal evolution of net (—, black) axial $F_x$ & (---, blue) lateral $F_y$ forces, and (---, red) net torque $M_z$ on the body of the eel measured by the present moving control volume approach. (c) Temporal evolution of (—, black) axial $P_x$ and (---, blue) lateral $P_y$ momentum of the entire fluid domain. Temporal evolution of change in linear momentum within the control volume: (—, gray) $I_x$ and (---, red) $I_y$.

correspond to time steps at which regridding occurs. Similar observations are made for angular momentum conservation for the entire system (data not shown here).
Drafting, kissing, and tumbling

In this section we simulate the dynamic interactions between two sedimenting cylindrical particles and use the moving control volume and Lagrange multiplier approaches to determine the hydrodynamic forces. The cylinders are identically shaped with diameter $D = 0.2$ cm and are placed in a domain of size $[-5D, 5D] \times [0, 40D]$, with zero velocity prescribed on the left and right boundaries, and with axial and transverse tractions set to zero at the top and bottom boundaries. The density and viscosity of the fluid are set to $\rho = 1.0$ g/cm$^3$ and $\mu = 0.01$ g/(cm $\cdot$ s), respectively. Each particle is subject to a gravitational body force $F^g = -(\rho_s - \rho)gV_pe_y$, where $g = 980$ cm/s$^2$ is the gravitational constant, $\rho_s = 1.01\rho$ is the density of the solid, and $V_p = \pi(D/2)^2$ is the volume of each particle. This is realized through an Eulerian body force $f^g$ added to the right-hand side the momentum Eq. (5.1), which is nonzero only in the particle domains. Similar to the free-swimming eel case, the particles’ translational and rotational velocities are obtained via Eqs. (5.51) and (5.52).

The domain is discretized with a two-level hierarchy of Cartesian grids. The size of the coarsest grid is $64 \times 256$ grid cells and $n_{ref} = 4$ is taken for subsequent finer grids. Hence, the finest grid, with spacing equivalent to that of a uniform mesh of size $256 \times 1024$, embeds each particle at all times. The minimum grid spacing on the finest level is $\Delta x_{min} = \Delta y_{min} = 0.0390625D$. A time step size of $\Delta t = 5 \times 10^{-4}$ s is used. Particle 1 is placed with initial center of mass $x_1 = (X_p, Y_p) = (-0.005D, 36D)$, while particle 2 is placed below particle 1 with initial center of mass location at $x_2 = (X_p, Y_p) = (0, 34D)$. Under these conditions, the two particles start to accelerate downwards due to gravity. Particle 1 travels through a low pressure wake created by the leading particle 2, which
causes particle 1 to fall faster; this stage is called drafting. Eventually, particle 1 catches up to and nearly contacts particle 2, a process termed as kissing in literature. This kissing stage is unstable and eventually the particles are left to tumble separately. The parameters here are chosen to match with previous numerical studies on drafting, kissing, and tumbling done by Feng et al. [223], Jafari et al. [224], and Wang et al. [225].

Artificial repulsive forces are added to avoid numerical issues due to overlapping particles. The functional form of this force on particle $i$ due to particle $j$ is given by

$$
F_{ij}^P = \begin{cases} 
0, & \|\mathbf{x}_i - \mathbf{x}_j\| > R_i + R_j + \zeta \text{ or } i = j \\
c_{ij} \epsilon_P \left( \frac{\|\mathbf{x}_i - \mathbf{x}_j\| - R_i - R_j - \zeta}{\zeta} \right)^2 \left( \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|} \right), & \|\mathbf{x}_i - \mathbf{x}_j\| \leq R_i + R_j + \zeta 
\end{cases}
$$

in which $R_i = R_j = R$ is the radius of both particles, $c_{ij} = \rho \pi R^2 g$ is a force scale parameter, $\epsilon_P = 2.0 \, \text{g cm/s}^2$ is a stiffness parameters for collisions, and $\zeta = \Delta y_{\text{min}}$ is a mesh threshold parameter indicating how far away the two particles need to be in order to feel particle-particle interaction force. This particular repulsive force was used by Feng et al. [223]. No repulsive force between the particle and the wall are used for the case considered here. Similar to the gravitational force, the particle interaction force is realized through an Eulerian body force $f_P$ added to the right-hand side of the momentum Eq. (5.1), which is nonzero only in the particle domains.

The strategy for placing the control volume for this example is different than the previous examples. Rather than setting the CV in motion with some prescribed velocity to enclose the body, the CVs are chosen to surround each cylinder based on its center of mass location: $[X_p - 5\Delta x_{\text{min}}, X_p + 5\Delta x_{\text{min}}] \times [Y_p - 5\Delta y_{\text{min}}, Y_p + 5\Delta y_{\text{min}}]$. This means that when the particles are close to each other during the kissing stage, each CV contains the
second body partially, leading to inaccurate force measurements. This is a limitation of using simple rectangular control volumes. Fig. 5.18 shows the vortical structures generated by the two particles at four separate time instances, along with the locations of the moving control volumes.

The expression for hydrodynamic force based on the Lagrange multiplier method needs to be modified to account for the presence of additional body forces in the solid region (additional to the Lagrange multiplier constraint forces). These forces need to be added
to the right-hand side of Eq. (5.45), yielding

\[ F_i = \frac{d}{dt} \int_{V_i(t)} \rho \ \mathbf{U}_i \, dV - \int_{V_i(t)} (F_i + F^g_i + F^p_{ij}) \, dV, \]

in which \( F_i, U_i, F^g_i \) and \( V_i(t) \) are the net hydrodynamic force, center of mass velocity, Lagrange multiplier force, gravitational force, and domain of particle \( i \), respectively. The expression for hydrodynamic force based on control volume analysis remains unchanged in the presence of the additional body forces in the solid region, and Eq. (5.16) remains valid in this scenario.\(^1\)

Figs. 5.19(a) and 5.19(b) show the time evolution of the centers of mass \((X_p, Y_p)\) of both particles. Particle 1 gradually approaches particle 2 up until near \( t = 2.5 \) s when the particles kiss and eventually separate. Upon separation, both particles over time reach a terminal velocity. This temporal behavior matches well with the results of Jafari et al.\(^{224}\). Figs. 5.19(c) and 5.19(d) show the time evolution of net axial \((F_x = F \cdot e_x)\) and lateral \((F_y = F \cdot e_y)\) forces acting on each particle, calculated using both the control volume and Lagrange multiplier approaches. Between \( t = 1 \) s and \( t = 2.5 \) s, the particles are close to each other and it is not possible to create rectangular CVs that contain only a single particle (see insets in Figs. 5.18(b) and 5.18(c)). Hence, the CV force calculations during this time period are inaccurate. However, the forces calculated by the LM method remain accurate at all times. Outside of this time period, both the CV and LM approaches are in excellent agreement. Eventually, the net hydrodynamic force on each particle is \(-F^g\), indicating a terminal velocity has been achieved.

\(^1\)This is because \( \int_{V_{CV}(t)} f \, dV \) that contains contribution of all body forces in Lagrangian domain is evaluated via the right-hand side of Eq. (5.49)
Figure 5.19. (a) Temporal evolution of the $x$ coordinate of center of mass $X_p$ for (—, black) particle 1 & (—, green) particle 2. (b) Temporal evolution of the $y$ coordinate of center of mass $Y_p$ for (—, black) particle 1 & (—, green) particle 2. Center of mass data from Jafari et al. [224] for (×, black) particle 1 & (■, green) particle 2. (c) Temporal evolution of net $F_x$ via the moving control volume approach on (—, black) particle 1 & (—, green) particle 2. Temporal evolution of net $F_x$ via the Lagrange multiplier approach approach on (---, blue) particle 1 & (---, yellow) particle 2. (d) Temporal evolution of net $F_y$ via the moving control volume approach on (—, black) particle 1 & (—, green) particle 2. Temporal evolution of net $F_y$ via the Lagrange multiplier approach approach on (---, blue) particle 1 & (---, yellow) particle 2. The time period between the two red lines indicates inaccurate CV force measurements since each rectangular CV contains multiple bodies.
5.5.6. Stokes flow

In the previous sections, we considered finite Reynolds number cases simulated using a direct forcing IB method in which Lagrange multipliers were approximated in the body domain. Here we consider a fully constrained IB method in which we compute Lagrange multipliers exactly. We consider Stokes flow examples here, although the fully constrained method also works equally well at finite Reynolds numbers as shown in [111].

For steady Stokes flow in the absence of inertia ($\rho = 0$), the momentum equation reads as

$$ - \nabla p(x) + \mu \nabla^2 u(x) + f(x) = 0. $$

(5.56)

Since Eq. (5.56) is a steady state problem, it cannot be solved numerically with the split fluid-structure solver described in [13]. Rather, the discretized system of Eqs. (5.11)-(5.13) are solved by the monolithic fluid-structure solver described in [111] to obtain a numerical solution to the constrained Stokes system. The rigidity constraint is enforced on the surface of the body, and the body is discretized only by surface nodes and not by a volumetric mesh. This is because enforcing the rigidity constraint on the surface also imposes rigid body motion inside the body for Stokes flow. Setting the inertial terms to zero in Eqs. (5.16) and (5.19) yield

$$ \mathcal{F} = \int_{Scv} n \cdot [-pI + T] \, dS, $$

(5.57)

$$ \mathcal{M} = \int_{Scv} [r \wedge (-p n + n \cdot T)] \, dS. $$

(5.58)
Hence, the net hydrodynamic force and torque on an object in Stokes flow is simply the pressure and viscous fluxes through the control surface. For the LM method, the Lagrange multiplier force density $F(X)$ is computed (exactly) on body surface $X \in S_b$, and the net force and torque is given by

$$\mathcal{F} = - \oint_{S_b} F(X) \, dS,$$

(5.59)$$\mathcal{M} = - \oint_{S_b} \mathbf{R} \wedge F(X) \, dS.$$ (5.60)

### 5.5.6.1. Flow between two concentric shells.
We first consider the case of two concentric shells, which was studied numerically by Kallemov et al. [111]. The inner and outer shell have geometric radii $R_1^g = 1.807885$ and $R_2^g = 4R_1^g$, respectively. The computational domain is a cube of size $[0, L]^3 = [0, 4.15R_2^g]^3$, which is discretized by a uniform grid of size $60^3$. The center of both shells is placed at $(x, y, z) = (L/2, L/2, L/2)$. Uniform velocity $u = (1, 0, 0)$ is prescribed on each wall of the computational domain, and the inner and outer shell are set to have rigid–body velocity $(U_1, V_1, W_1) = (0, 0, 0)$ and $(U_2, V_2, W_2) = (1, 0, 0)$, respectively. The inner shell is discretized with 42 surface markers while the outer shell is discretized with 642 surface markers to ensure that the markers are about 2 grid cells apart. The viscosity is set to $\mu = 1$.

Each spherical shell has an effective hydrodynamic radius $R^h$ due to the immersed boundary kernel used to discretize the delta-function [226]. For the 6-point kernel considered by Kallemov et al., it was found that $R^h_1 = 1.22R_1^g$ and $R^h_2 = 0.96R_2^g$ for the numerical parameters chosen here [111]. It was also found that as both the Eulerian and Lagrangian meshes are refined, $R^h/R^g \to 1$. These hydrodynamic radii can be used in
the analytical expression for the drag on the inner sphere \([227]\), given by

\[
\mathbf{F}_{\text{exact}} \cdot \mathbf{e}_x = -6\pi \mu R^h_1 U_2 K,
\]

in which \(K = (1 - \lambda^5)/\alpha\), \(\alpha = 1 - 9\lambda/4 + 5\lambda^3/2 - 9\lambda^5/4 + \lambda^6\) and \(\lambda = R^h_1/R^h_2\).

Depending on the control volume size, the drag on either the inner shell, or on both shells can be obtained. First, a CV of dimension \([L/2 - 1.659R^g_1, L/2 + 1.659R^g_1]^3\) is chosen to surround the inner shell, but to exclude the outer shell. Next, a CV of dimension \([L/2 - 1.383R^g_2, L/2 + 1.383R^g_2]^3\) is chosen to include both inner and outer shells. Fig. 5.20 shows the configuration of the concentric shells/spheres and the two control volumes.

Table 5.1 shows the drag measurements from the analytical expression, from integrating surface Lagrange multipliers, and from the control volume analysis. The middle column shows that all three methods are in agreement for the drag on the inner shell. Moreover, the last column shows that the combined drag on both inner and outer shells is the same when computed from Lagrange multipliers and control volume analysis.

### 5.5.6.2. Single rotating shell

Next, we consider a single shell rotating in a bounded domain with no exterior flow. The shell is taken to be the same as the inner sphere in the previous example, with geometric radius \(R^g_1 = 1.807885\) in a computational domain of size \([0, L]^3 = [0, 16.6R^g_1]^3\). There is no outer shell in this example. The center of the shell

| Method                           | \(|\mathbf{F} \cdot \mathbf{e}_x| (\text{Inner})\) | \(|\mathbf{F} \cdot \mathbf{e}_x| (\text{Inner & Outer})\) |
|----------------------------------|-----------------------------------------------|--------------------------------------------------|
| Analytical Eq. (5.61)            | 115.409                                       | N/A                                              |
| Lagrange multiplier Eq. (5.59)   | 114.297                                       | 17.350                                           |
| Control volume Eq. (5.57)        | 114.298                                       | 17.350                                           |
is placed at the centroid of the cube \((x, y, z) = (L/2, L/2, L/2)\) and \(u = (0, 0, 0)\) is set at all computational boundaries. Periodic boundary conditions were also used and yielded nearly identical torque measurements (data not shown). The viscosity is set to \(\mu = 1\). The shell rotates about a diameter with angular velocity \(\omega = (0, 1, 0)\). In an unbounded flow at rest, Faxén’s law states that the torque on the sphere by the fluid is

\[
\mathcal{M} = -8\pi \mu (R^h_1)^3 \omega,
\]

in which the hydrodynamic radius of the sphere \(R^h_1\) is used [227]. Although the domain in the numerical method is bounded, we still get decent agreement between our numerical results and Eq. (5.62).
Table 5.2. Comparison of torque measurements $\mathbf{M} \cdot \mathbf{e}_y$ for Stokes flow around single rotating sphere.

<table>
<thead>
<tr>
<th>Grid size</th>
<th>Number of markers</th>
<th>$R_1^3/R_1^3$</th>
<th>Analytical Eq. (5.62)</th>
<th>LM Eq. (5.60)</th>
<th>CV Eq. (5.58)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60$^3$</td>
<td>42</td>
<td>1.22</td>
<td>-269.669</td>
<td>-236.474</td>
<td>-236.474</td>
</tr>
<tr>
<td>120$^3$</td>
<td>162</td>
<td>1.09</td>
<td>-192.323</td>
<td>-183.662</td>
<td>-183.662</td>
</tr>
<tr>
<td>240$^3$</td>
<td>642</td>
<td>1.04</td>
<td>-167.052</td>
<td>-164.446</td>
<td>-164.446</td>
</tr>
</tbody>
</table>

A CV of dimension $[L/2 - 1.659R_1^3, L/2 + 1.659R_1^3]^3_3$ is chosen to surround the shell. Three different grid sizes are used to discretize the domain: 60$^3$, 120$^3$, and 240$^3$, which corresponds to 42, 162, and 642 surface markers on the shell respectively (to ensure that the markers are approximately 2 grid cells apart).

Table 5.2 shows the torque measurements $\mathbf{M} \cdot \mathbf{e}_y$ from the analytical expression, from integrating the moments of surface Lagrange multipliers, and from the control volume analysis, for the three different grid resolutions. As expected, the LM and CV measured torque do not match exactly with the analytical expression (presumably because of finite domain effects). However, the LM and CV torque values are in excellent agreement with each other.

Conclusions

In this chapter we presented a moving control volume (CV) approach to compute the net hydrodynamic forces and torques on a moving body immersed in a fluid. This approach does not require evaluation of (possibly) discontinuous spatial velocity or pressure gradients within or on the surface of the immersed body. The analytical expressions for forces and torques were modified from those initially presented in [205], and this modification has been shown to eliminate spurious jumps in drag [207]. Our implementation
treats the control volume as a rectangular box whose boundary is forced to remain on grid lines, which greatly simplifies the evaluation of surface integrals.

The approach is shown to accurately compute the forces and torques on a wide array of fluid-structure interaction problems, including flow past stationary and moving objects, Stokes flow, and high Reynolds number free-swimming. Spurious momentum gain or loss due to adaptive mesh refinement can produce jumps in the computed forces in the CV approach, although forcing the CV to remain on the finest grid level can ameliorate this issue.

We also show the equivalence between the Lagrange multiplier (LM) approach and the CV approach. The main advantage of the CV approach over the LM approach is that it is applicable to situations where explicit Lagrange multipliers are not available, for example, in the embedded boundary/cut-cell approach to FSI.

The control volume approach implemented here assumes a no-slip boundary condition on the fluid-structure interface. However, a generalization for transpiration boundary conditions can be derived as well (see Eq. (5.26)). Use of such a boundary condition would required richer geometric information and data structures to evaluate surface quantities of the immersed body. Finally, our approach can be easily extended to cases where additional body forces are present in the momentum equation.
CHAPTER 6

Ongoing work and future scope

Thus far in this thesis, we have described a robust numerical method for simulating high density ratio, multiphase fluid-structure interaction. We have also investigated hydrodynamic optimality in aquatic locomotion for body/caudal fin swimmers and developed a moving control volume approach to smoothly compute moments on immersed bodies. In this chapter, we present ongoing and future work that build upon the ideas presented in the previous chapters. Two major research directions are currently being pursued: 1) unification of the aquatic locomotion parameter space and 2) application of the multiphase constraint immersed boundary method to simulation of industrial fluid flow problems.

6.1. Unified landscape of optimal aquatic locomotion

6.1.1. Introduction

Given an undulatory swimmer of length $L$, span $h$ with deformational kinematics prescribed by,

\[
y(x, t) = a(x) f(x, t) = a(x) \sin \left( \frac{2\pi}{\lambda} (x - \lambda f t) \right),
\]

how should its swimming parameters be chosen to generate the greatest speed or efficiency? Aquatic species in nature have efficient swimming kinematics that are fine-tuned
by natural selection. One must quantitatively describe not only the efficiency of these swimming animals, but also the morphological patterns found in these species in order to develop design criteria for underwater vehicles. This is generally done using dimensionless quantities that enable comparison across length scales [146, 228, 151, 163]. In Chapter 4, we investigated one such parameter: the wavelength to tail amplitude ratio (called the specific wavelength SW), which was shown to maximize swimming speed or thrust for BCF animals. In this chapter, we review a few more nondimensional swimming parameters and provide preliminary results for their optimality principles using fully-resolved simulations and observational data from swimming fish. In future work, we will compare these different measures to each other in order to develop a unified and wholistic understanding of the aquatic locomotion parameter space. Doing so will allow us to determine optimal design criteria for underwater vehicle.

6.1.2. Nondimensionalization

We begin our analysis by nondimensionalizing the swimming body system and separating the input parameters, which are known \textit{a priori}, from the output parameters, which are taken to be averages over time after an oscillatory steady-state has been reached. When studying the hydrodynamics of swimming fish or underwater vehicles, two scenarios are generally considered: static and free-swimming.

First, we would consider an undulating swimmer whose translational degrees of freedom are locked; we call this the \textit{static} case. In this scenario, the net axial static force $F_{st}$ is measured as a function of the input parameters in Eq. \ref{6.1}, whereas the swimming speed is not considered, i.e. $U_s = 0$. The other output parameter is the net static power
$P_{st}$ spent by a swimmer due to undulations in the lateral direction. For a rectangular fin with constant amplitude profile $a(x) \equiv a$, the static force and power depend on the following physical parameters:

\begin{align}
F_{st} &= fn(\rho, \mu, f, \lambda, L, h, a)|_{U_s=0}, \\
P_{st} &= fn(\rho, \mu, f, \lambda, L, h, a)|_{U_s=0},
\end{align}

in which $\rho$ and $\mu$ are the density and viscosity of the fluid respectively. Here, $fn$ denotes “function of”. For a suitable nondimensionalization, we obtain

\begin{align}
\hat{F}_{st} &= \frac{F_{st}}{\frac{1}{2} \rho (fa)^2 L h} = fn \left( \frac{\lambda}{a}, \frac{h}{L}, \frac{h}{a}, \frac{\rho (fa) a}{\mu} \right) \bigg|_{U_s=0} = fn \left( SW, AR, \hat{h}, \text{Re}_{lat} \right) \bigg|_{U_s=0}, \\
\hat{P}_{st} &= \frac{P_{st}}{\frac{1}{2} \rho (fa)^3 L h} = fn \left( \frac{\lambda}{a}, \frac{h}{L}, \frac{h}{a}, \frac{\rho (fa) a}{\mu} \right) \bigg|_{U_s=0} = fn \left( SW, AR, \hat{h}, \text{Re}_{lat} \right) \bigg|_{U_s=0},
\end{align}

in which the right-hand side of each equation contains what we call the fundamental parameters that describe aquatic locomotion: the specific wavelength $SW$ [151], the fin aspect ratio $AR$ [121, 160], the scaled span $\hat{h}$, and the Reynolds number based on lateral speed $Re_{lat}$ (Chapter 4).

Next we consider the case in which the translation-locked swimmer has an axial fluid field of strength $U_\infty$ opposing it. The net axial force is then given as a function of fin parameters and this inlet velocity: $F_{U_\infty} = fn(\rho, \mu, f, \lambda, L, h, a, U_\infty)$. The inlet velocity value that produces $F_{U_\infty} = 0$ is exactly the free-swimming speed $U_s$ with an associated
power expenditure $P_{Us}$, both of which are functions of the input parameters:

\begin{align}
U_s &= fn(\rho, \mu, f, \lambda, L, h, a)\big|_{F_{U\infty}=0}, \\
P_{Us} &= fn(\rho, \mu, f, \lambda, L, h, a)\big|_{F_{U\infty}=0}.
\end{align}

We can nondimensionalize Eq. (6.6) and (6.7) in the following way,

\begin{align}
\frac{1}{St} = \frac{U_s}{2fa} &= fn\left(\frac{\lambda}{a}, \frac{h}{L}, \frac{h}{a}, \frac{\rho(fa)a}{\mu}\right)\bigg|_{F_{U\infty}=0} = fn\left(SW, AR, \hat{h}, Re_{lat}\right)\bigg|_{F_{U\infty}=0}, \\
\hat{P}_{Us} &= \frac{P_{Us}}{\frac{1}{2}\rho(fa)^3Lh} = fn\left(\frac{\lambda}{a}, \frac{h}{L}, \frac{h}{a}, \frac{\rho(fa)a}{\mu}\right)\bigg|_{F_{U\infty}=0} = fn\left(SW, AR, \hat{h}, Re_{lat}\right)\bigg|_{F_{U\infty}=0},
\end{align}

in which the left-hand side of Eq. (6.8) is the inverse of the Strouhal number, which has been associated with describing efficient swimmers when constrained to a range between $0.2 - 2.2$ \cite{146, 148}. Note that in the free-swimming case, $St$ is strictly an output parameter that is calculated after the swimmer has reached a steady-swimming speed.

It was shown in Chapter 4 of this thesis that a specific wavelength value of around $SW = 10$ maximizes swimming speed (for free-swimming simulations) and axial force (for static simulations). In the following sections, we use the constraint immersed boundary method \cite{13} to carry out fully-resolved 3D simulations to investigate some of the other dimensionless quantities shown here in order to determine whether or not an optimality principle can be established.
6.1.3. Parametric study

In this section, we systematically study the effect of a handful of the fundamental input parameters on the propulsive output parameters shown in the nondimensionalization Eqs. (6.4), (6.5), (6.8), and (6.9). In some cases, we reuse simulation data from Chapter 4 and connect those results to the dimensionless quantities described here.

6.1.3.1. Optimality in swimming speed and axial static force. Consider a fin held stationary ($U_s = 0$) but undulating according to its prescribed swimming kinematics given by Eq. (6.1). This fin imparts momentum into the fluid and would propel forward had its translational degrees of freedom not been locked. Instead, the fin generates a static thrust $F_{st}$, which is the axial force required to hold the fin stationary. Therefore, this static thrust is not a quantity measured using a drag–thrust decomposition.

Now suppose the same fin is allowed to swim freely: the swimming speed is an output parameter of the experiment or simulation. It is reasonable to expect that the kinematics that would produce faster swimming velocities in this case would produce greater static thrusts in the stationary case. This correlation has been shown to be true when SW is varied for robotic knifeﬁsh and numerical simulations of ribbon fins and constant amplitude undulating sheets [151].

Both of these output parameters appear on the left-hand side of our dimensionless equations (6.4) and (6.8) via the dimensionless static force $\hat{F}_{st}$ and the inverse of the Strouhal number $1/St$. Note that in our framework, either the Strouhal number does not exist (static; Eqs. (6.4) and (6.5)) or it is strictly an output parameter and not prescribed a priori (free-swimming; Eqs. (6.8) and (6.9)). Therefore, we argue that St itself does not
induce an optimality principle for a self-propelled swimmer. Rather, $St^{-1}$ is maximized by SW.

Fig. (6.1) shows data from computational parametric studies of BCF shown in Chapter 4 and MPF swimmers [151], as well as experimental studies on BCF [156] and MPF robots [151]. In all of these experiments, SW was varied and $U_s$ was measured and in all cases, $St^{-1}$ is maximized at some optimal specific wavelength (OSW). The St values in these cases are between 0.33 and 1.03, which is consistent with swimmers observed in nature. Hence, we forego the common belief that the Strouhal number can be used to prescribe the kinematics of an efficient swimming machine—it is the swimming speed equivalent to a drag or power coefficient since its definition mixes input and output parameters. Hereafter, we analyze only the free-swimming case.

6.1.3.2. Can hydrodynamic power spent be minimized? Having established that the axial output parameters $U_s$ and $F_{st}$ are correlated and maximized by SW, we now investigate whether the hydrodynamic power spent in the free-swimming ($P_{U_s}$) is minimized.
by the any of the fundamental parameters. As present, we only consider the relationship between power and the lateral Reynolds number, the scaled span, and the aspect ratio; correlating power $P_{U_s}$ to the other input parameters is ongoing work.

Fig. 6.2 show the power spend and swimming speed attained as a function of lateral Reynolds number for the undulating eel simulations presented in Chapter 4. It is seen that these two output parameters both increase as a function of $Re_{lat}$. This is expected because the frequency of undulation is the dimensional quantity that is varied in order to increase the lateral Reynolds number, while all other input parameters are held fix. An increase in $f$ will cause the swimmer to push fluid backwards at higher speeds (as quantified by the velocity mechanism described earlier) and hence increases the body’s swimming speed. What is interesting to note is that while the swimming speed increases linearly with $Re_{lat}$, the power spent increases nonlinearly at a faster rate. We are currently unsure of the exact physical mechanisms that induce this scaling. Next, we consider the relationship between power and the scaled span $\hat{h}$ and the aspect ratio $AR$. Fig. 6.3 shows simulation data from free-swimming, rectangular sheets undulating with carangiform kinematics for three different spans $h$ and varying tail amplitudes $a$. Unsurprisingly, as the tail amplitude decreases ($\hat{h}$ increases), the power decreases. In these cases, $a$ and $\lambda$ vary together to maintain a constant $SW = 6.67$. Moreover, it is seen that the power spent by these same carangiform sheets increases with aspect ratio.

Based on these results, it is evident that increasing the lateral area swept by the traveling wave increases the swimming power. Additionally, increasing the speed of the traveling wave $V_f = \lambda f$ via frequency ostensibly increases $P_{U_s}$. 
Figure 6.2. The hydrodynamic power $P_U$ spent and axial swimming speed vs. lateral Reynolds number $Re_{lat}$ for the simulated eel data presented in Chapter 4.

Figure 6.3. The hydrodynamic power $P_U$ spent and axial swimming speed vs. scaled height at three different aspect ratios for carangiform sheets.
6.1.3.3. A comment on swimming efficiency. In the previous section, we investigated how the power spent by an undulatory swimmer varies as a function of some of the fundamental input parameters. However, hydrodynamic power is a dimensional quantities and we would prefer to instead work with quantities that encapsulate both power and swimming speed in a dimensionless way. Hence, we will briefly discuss efficiency for the aquatic locomotion problem.

Determining the efficiency of a self–propelled swimmer is a fundamental issues in hydrodynamics. In biology, a commonly used measure is the mechanical cost of transport \( COT = P_{Us}/Us \), which quantifies the hydrodynamic energy spent per unit distance traveled. However, COT is a dimensional quantity, whereas a dimensionless efficiency metric is preferred to enable comparison across disparate speed and length scales. Moreover, there is no standardized way to nondimensionalize cost of transport by input parameters; a brief review of COT-based efficiency measures is presented in [229].

In our framework, a natural scaled COT can be obtained by dividing Eq. (6.9) by Eq. (6.8). After scaling by a factor of 1/2, we obtain an energy–consumption coefficient:

\[
C_{E,fa} = \frac{COT}{\frac{1}{2} \rho (fa)^2 Lh}
\]

(6.10)

This is akin to the energy-consumption coefficient presented in [228]. Note that \( 1/C_{E,fa} \) is admittedly a measure of efficiency that uses another “non-standard” nondimensionalization.

Nondimensionalizing cost of transport will necessarily require scaling COT by some combination of input parameters of the swimmer. Other dimensionless measures of efficiency, such as the Froude efficiency \( \eta_F \) [230, 231], have been used to describe the
kinematics of an optimal swimmer, but rely on normalization by the drag and thrust forces felt on the body. The separated drag and thrust values, which have not been uniquely calculated for a free-swimming animal [232, 233], are not readily available from an experiment and therefore these measures of efficiency have limited viability.

Another metric called the quasi-propulsive efficiency $\eta_{QP} = (F_{U_\infty} + R)U_\infty / P_{U_\infty}$ was proposed in [229]; in this new measure, $R$ is the magnitude of the drag on a dead fish without undulations, towed straight at a velocity $U_\infty$. Physically, $\eta_{QP}$ describes how well drag on a body is reduced by introducing undulatory mechanisms onto the body. For example, a negative $\eta_{QP}$ indicates that traveling undulations actually increase the drag on the swimmer ($F_{U_\infty} + R < 0$). Evaluating the quasi-propulsive efficiency requires two sets of simulations or experiments: one static simulation in which the body is not allowed to undulate, and one standard free-swimming case with the desired prescribed kinematics.

Determining the relationship between the various efficiency measures of aquatic locomotion remains challenging future work. Indeed is is unclear which efficiency measure is “correct” and it is likely that the “most suitable” measure of efficiency will depend strongly on the design process for one’s specific application. Henceforth, we will present results in terms of swimming speed, Strouhal number, and cost of transport.

6.1.3.4. A parametric landscape for optimal swimming. Next, we investigate a parametric landscape for optimal aquatic locomotion. For a swimmer of fixed body shape ($L$ and $h$) and morphology ($a$), the two remaining quantities that can be varied are the parameters controlling the kinematics of the undulatory motion: wavelength $\lambda$ and frequency $f$. These have already been nondimensionalized in Sec. 6.1.2 as the specific wavelength SW and the lateral Reynolds number $Re_{lat}$. We expect that these two parameters form a
phase space for undulatory swimming, which can be used to determine regions of optimal locomotion.

As a preliminary step, we carry out a parameter study of a swimming eel geometry with fixed length $L = 10$ cm and tail amplitude $a = 1$ cm. Results for the velocity landscape are shown in Fig. 6.4. It is seen that for a fixed frequency, the specific wavelength maximizes swimming speed, which is in agreement with the results presented in Chapter 4. For a fixed SW, the eel’s swimming speed increases as a function of $Re_{lat}$. Moreover, it is seen that Strouhal numbers between 0.2 and 0.5 are achieved when its swimming velocity is maximized at the optimal specific wavelength. Finally, it is seen that observations of undulatory swimmers from the meta-analysis of [148] cluster around the range of optimal specific wavelength values between 4 and 16, but take on a large range of $Re_{lat}$ values.

Fig. 6.5 shows the cost of transport expended by these same eel simulations, as a function of varying SW and $Re_{lat}$. It is seen that the highest COT values correspond to the locations of maximal velocities, at higher lateral Reynolds numbers. This is in agreement with the result shown in Fig. 6.2 which shows that, as a function of frequency, the power spent increases at a faster rate than swimming speed. We are currently investigating patterns in the efficiency landscape for the measures described in Sec. 6.1.3.3.

6.1.3.5. **Empirical scaling laws.** The discussion on optimality principles in the previous sections of this chapter is based on simulation data from parametric studies of simplified swimmers. In nature, the morphology of a fish can be significantly more complicated. The span of a natural swimmer often varies along its body length, and its body contains other appendages such as pectoral, dorsal, anal, and caudal fins, which
Figure 6.4. The (a) axial swimming speed, and (b) Strouhal number generated from simulations of an undulating eel body with fixed length $L = 10$ cm and tail amplitude $a = 1$ cm; (●, black) the Re$_{lat}$ and SW values used for each simulation in the parametric study; (■, white) the Re$_{lat}$ and SW values for observed BCF swimmers [148].

Figure 6.5. The cost of transport expended by an simulated, undulating eel body with fixed length $L = 10$ cm and tail amplitude $a = 1$ cm; (●, black) the Re$_{lat}$ and SW values used for each simulation in the parametric study; (■, white) the Re$_{lat}$ and SW values for observed BCF swimmers [148].
can affect the animal’s swimming efficiency; these additional complexities must be investigated in future simulation studies. However, establishing optimality principles using free-swimming organisms can be difficult because conducting parametric studies on a living swimmer is infeasible or impossible. Nonetheless, it is reasonable to assume that these natural swimmers are fine-tuned to be efficient through generations of natural selection \[146, 228, 159\].

Therefore, we hypothesize that empirical scaling relations, such as those shown in Fig. 4.14, can be interpreted as edges of optimal performance by an undulatory locomotor. Parametric studies done using numerical simulations or actual robotic prototypes can be used to interrogate the parameter space away from these edges. In Fig. 6.6, we show two such scaling laws between the Strouhal number and the wave-speed Reynolds number \( \text{Re}_{\text{wave}} = \rho(\lambda f)L/\mu \), and between the Reynolds number and the lateral Reynolds number, computed from the meta-analysis data from \[148\]. It is seen that the optimal values from simulation (from the present work) and experimental (from \[155, 156\]) parametric study data coincide with the scaling laws, which is evidence to support this hypothesis. Relating the various scaling laws and optimality principles remains challenging future work.

### 6.1.4. Future directions

In this chapter, we presented preliminary and ongoing investigations into the dimensionless aquatic locomotion parameter space. In the future, we plan to conduct further parametric studies to investigate the effect of the remaining fundamental parameters on both propulsive performance and power expenditure. Additionally, a correlation between the various scaling relationships for actual fish data will be interrogated. More simulation
Figure 6.6. (a) (♦, red) Empirical scaling between the Strouhal number and wave-speed Reynolds number, and (b) (∙, green) between Reynolds number and lateral Reynolds number for meta-analysis data of BCF swimmers [148], along with simulation data of undulatory (●) eel, (■) sheet, and (♦) mackerel bodies; (▲) robotic parametric study data of robotic, undulatory sheets [155, 156]. Parametric study data are colored from blue to orange to represent the variation from lower to higher swimming speeds within each set.

and observational data are required to fully quantify these relationships. We envision a phase space for which the hydrodynamic efficiency of an underwater vehicle can be directly evaluated, which will lead to better engineering design processes. Finally, we will develop a more concrete understanding of the relationship between the parameters described in this thesis and elastic swimming parameters for flexible swimmers.
6.2. Simulation of industrial fluid-structure interaction problems

6.2.1. Introduction

In Chapter 3, we presented a numerical method that enabled simulations of high density ratio, fluid-structure interaction. This method has been implemented within IBAMR to enable large-scale simulations of FSI problems in parallel. However, thus far, the techniques presented in this thesis have been restricted to relatively simple geometries and moderately inertial flows. Many engineering fluid flow problems involve the turbulent interaction between heavy, rigid objects one or more fluid phases. The goal of this chapter is to describe the ongoing and future work in applying our numerical methodology to industrially relevant problems, such as vehicle aerodynamics and renewable energy devices.

6.2.2. Turbulence modeling

The airflow past an object is characterized by the Reynolds number \( \text{Re} = UL/\nu \), where \( U \) and \( L \) are some characteristic velocity and length scale for the flow, and \( \nu = \mu/\rho \) is the kinematic viscosity of the fluid. As \( \text{Re} \) increases, the flow becomes less viscous and increasingly turbulent: large, energetic eddies “spin up” smaller vortices in a process called the energy cascade (see Fig. 6.7). The smallest scales contribute significantly to the overall dissipation of momentum and energy in the fluid \([234]\). Total resolution of these disparate length scales and energy cascade in computational fluid dynamics is vitally important to obtain accurate measurements of drag and to predict flow separation and reattachment points on a solid body. However, direct numerical simulation (DNS) of the incompressible Navier-Stokes momentum and continuity equations incurs an enormous computational
cost. Realistic flows around automobile bodies typically have Reynolds numbers above one million. A DNS of this type of flow requires an estimated $7 \times 10^{15}$ grid cells, a computational task realizable today by only the world’s largest supercomputers [235].

A numerical alternative to DNS is large eddy simulation (LES). In LES, the strategy is to resolve the most energetic eddies and model the effective dissipation due to the small-scale physics. Although LES is efficient in separated flow regimes, the computational cost scales strongly with the Reynolds number in the viscous sublayer near a solid body since the most energetic eddies have length scales on the order of the boundary layer thickness [234]. Whilst there is a moderate reduction in computational cost by using LES
over DNS, a full LES of flow past a ground vehicle would still require upwards of $10^{11}$ grid cells and $10^7$ time steps and is estimated to be a viable industrial approach by 2045 [236].

Another alternative is the Reynolds-averaged Navier-Stokes (RANS) approach, in which all of the turbulent fluctuations are modeled and the fluid flow equations are time- or ensemble-averaged. RANS is the current industry standard for engineering flows since it can accurately predict the growth and mean flow in attached boundary layers at a much lower computational cost than LES and DNS, although it performs poorly in massively separated and strongly unsteady regimes [234].

We note that many LES and RANS turbulence models make use of the eddy viscosity formulation, for which additional viscous stresses are applied to the filtered incompressible Navier-Stokes equations

$$\rho \left( \frac{\partial \tilde{u}}{\partial t} + \tilde{u} \cdot \nabla \tilde{u} \right) = -\nabla p + \nabla \cdot \left[ (\mu + \mu_T) \left( \nabla \tilde{u} + \nabla \tilde{u}^T \right) \right] + f,$$

(6.11)

$$\nabla \cdot \tilde{u} = 0,$$

(6.12)

in which $\tilde{u}(x, t)$ is the filtered velocity and $\mu_T(x, t)$ is a spatially and temporally varying eddy viscosity field. Note that the solution methodology described in Chapter 2 is used to implicitly treat this added viscosity term. In practice, these equations are solved with conventional CFD approaches, with the interpretation that the outputted numerical solution for velocity is already filtered. For LES, $\tilde{u}(x, t)$ represents the velocity of all the most energetic eddies up to a certain cutoff in the turbulent energy cascade (Fig. 6.7).

In contrast for RANS, $\tilde{u}(x, t)$ represents the ensemble averaged velocity field, effectively
smoothing out any turbulent fluctuations. In the next section, we describe a hybrid RANS-LES approach called detached eddy simulation (DES), which has been wildly successful in the automotive and aerospace industries.

### 6.2.3. Detached eddy simulation

As a first step towards an industrially viable simulation technique for accurate turbulent flow predictions past moving and self-propelled bodies, we combine the DLM immersed boundary solver from Chapter 3 with detached eddy simulation. DES was proposed by Spalart et al. in 1997 \(^{237}\) and a more robust version called delayed detached eddy simulation (DDES) was also proposed by Spalart et al. in 2006 \(^{238}\); henceforth we simply refer to “DDES” as “DES” since it is a strict improvement over the original method. Detached eddy simulation is a hybrid turbulence model that acts like RANS in attached boundary layer regions (where a pure LES would be cost prohibitive), and acts like LES in separated flow regimes (where a pure RANS is inaccurate) \(^{237}\). DES is specifically designed with aerodynamics in mind and it is gaining popularity and viability for industrial applications \(^{236, 239}\).

Similar to other eddy viscosity based turbulence models, the filtered momentum Eqs. (6.11) and (6.12) are solved to obtain a numerical solution to the filtered velocity. To compute the eddy viscosity, the Spalart-Allmaras (S-A) advection-diffusion-reaction equation is solved for to obtain an auxiliary quantity \(\tilde{\nu}\). Its evolution equation in the
immersed boundary framework reads,

\[
\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \tilde{\nu} = C_{b1} \left( S \tilde{\nu} + \frac{\tilde{\nu}^2}{\kappa^2 d^2} f_{e2} \right) + \frac{1}{\sigma} \left\{ \nabla \cdot [(\nu + \tilde{\nu}) \nabla \tilde{\nu}] + C_{b2} |\nabla \tilde{\nu}|^2 \right\} - C_{w1} f_w \frac{\tilde{\nu}^2}{d^2} + g_c(\mathbf{x}, t),
\]

(6.13)

\[
\int_{\Omega} \tilde{\nu}(x, t) \delta(x - X(s, t)) \, dx = 0,
\]

(6.14)

in which \( C_{b1} = 0.1355, C_{b2} = 0.622, \kappa = 0.41, \sigma = 2/3, C_{v1} = 7.1, C_{v1} = C_{b1}/\kappa^2 + (1 + C_{b2})/\sigma, C_{w2} = 0.3, \) and \( C_{w3} = 2.0 \) are model constants, \( S = \sqrt{2R} : \mathbf{R} \) where \( R = \frac{1}{2} (\nabla \mathbf{u} - \nabla \mathbf{u}^T) \), \( g_c(\mathbf{x}, t) \) is a Lagrange multiplier enforcing the vanishing eddy viscosity boundary condition on the body, and \( \tilde{d} = \min(\psi, 0.65\Delta) \) where \( \psi \) is the signed distance to the surface of the immersed body (see Sec. 3.4.1) and \( \Delta = \max(\Delta x, \Delta y, \Delta z) \). Note that if \( \tilde{d} \) is simply taken to be the signed distance function \( \psi \), then we recover the original Spalart-Allmaras one-equation RANS model. This modified choice of \( \tilde{d} \) for DES is chosen to ensure that the model transitions from RANS in the boundary layer of the object, to LES downstream of flow separation.

Following the projection approach described in Chapter 3, we enforce the constraint Eq. (6.14) by solving,

\[
\frac{\tilde{\nu}^{n+1,k+1} - \tilde{\nu}^*}{\Delta t} = g_c(\mathbf{x}, t) = \frac{S_h [\mathbf{X}^{n+1,k+1}] (0 - \mathcal{J}_h [\mathbf{X}^{n+1,k+1}] \tilde{\nu}^*)}{\Delta t},
\]

(6.15)

in which \( \tilde{\nu}^* \) represents the uncorrected value obtained from integrating Eq. 6.13, which will not necessarily vanish within the structure region. The final eddy viscosity is obtained via \( \mu_T = \rho \frac{(\tilde{\nu}/\nu)^3}{(\tilde{\nu}/\nu)^3 + C_{S1}^3} \). In our present implementation, Eq. (6.13) is numerically solved for with a standard second-order Crank-Nicolson (for linear terms) and Adams-Bashforth
(for nonlinear terms) integrator, and the DLM immersed boundary method of Chapter 3 is used to produce numerical solutions to the fluid-structure interaction.

Next, we present a preliminary qualitative result to validate the implementation of DES within the IB framework. We consider flow past a NACA 0012 airfoil at two angles of attack $\alpha$. At $\alpha = 5^\circ$, for a pre-stall configuration, the boundary layer around the body remains largely attached, with RANS and DES giving similar results (see Fig. 6.8). However some smoothing is seen in the foil tip vortices downstream of the body when the RANS model is employed, while the DES model provides finer detail. In contrast for the post-stall angle of attack $\alpha = 18^\circ$, the RANS model significantly smoothes out the vorticity and incorrectly predicts significant attachment of the boundary layer. However, DES shows that the detached airflow is well-resolved as this turbulence model has transitioned to LES after detecting massively separated flow features. This gives us confidence that these models have been implemented correctly, but a full quantitative validation is ongoing and remains challenging future work.

6.2.4. Self-propelled vehicles

Computational fluid dynamics has been used in the automotive industry to better understand the aerodynamics of heavy vehicles [1]. Simulations of stationary vehicles is the current industry standard, and these tools often make use of body-fitted meshes [240, 241, 242, 243]. However, continual generation of unstructured, conforming grids can be prohibitively expensive and inefficient for moving automobile bodies with complicated geometries. In recent years, immersed boundary methods have been used to successfully simulate the flow past idealized road vehicles and a pickup truck [244, 245].
In realistic driving scenarios, the vehicle is moving and its motion can be influenced by the surrounding airflow. Simulating these situations is important because aerodynamic drag causes significant mileage loss during highway driving and fluctuating wind conditions (e.g. crosswind) can affect driving stability. However, as far as we know, there has been no work done on the challenging problem of simulation self-propulsion of an entire automobile with full resolution of the surrounding flow. We are continuing to investigate this because accurate simulation of these phenomena would increase efficiency in the automotive design process. Our formulation described in Chapter 3 combined with an appropriate turbulence model has enabled these types of simulations, in which the density in the solid region may be 3-4 orders of magnitude larger than that of the surrounding air phase. By prescribing a force and density (larger than that of air) within the body
region, we are able to simulate self-acceleration of vehicles and fully take into account the weight of the structure.

A common benchmark case in the automotive industry is flow past a simplified car model called the Ahmed body; it dimensions are described in [246]. However, much of the investigation into the wake structure and flow around this model are restricted to stationary simulations due to the high computational costs and software infrastructure required for moving mesh problems. Here, we show preliminary results for moving and self-propelled automobile simulations for the Ahmed model. In Fig. 6.9, the body is set to translate at a constant speed of 29 mph, however its degrees of freedom in the $z$-direction are unlocked. As expected, the airflow around the object causes the model to shake back and forth.

Fig. 6.10 shows self-propulsion of a model vehicle. In this case, a heavy Ahmed body weighing about 661 pounds and initially at rest is given an imposed “engine forcing” of 7507 N in the horizontal direction; this represents the force generated by the engine due to a driver stepping on the accelerator. This causes the body to accelerate from its initial position, eventually reaching a maximum cruising speed of about 38 mph. We want to emphasize that the speed of the automobile is an output of the simulation, i.e. it is not prescribed. Moreover, the pitch degrees of freedom are locked for this particular case to ensure that the model does not flip forward due to the uneven downforce exerted by the external airflow. Validating and simulating complex automobile self-propulsion is left as exciting future work.

Finally, we note that there are two other simplified vehicle models that are used for validation of industrial CFD codes: the SAE Notchback model [247] and the DrivAer
Figure 6.9. (a) Visualization of a translating Ahmed body with unlocked degrees of freedom in the $z$-direction. (b) Temporal evolution of the $z$-coordinate of the body’s center of mass.

Each model is progressively closer to a realistic car geometry and contain additional complexities that are missing from the Ahmed body. However since these validation cases are more geometrically complicated, using a finite element methodology is vital to accurately represent them in a numerical simulation tool. We are pursuing techniques to import publicly available CAD files within IBAMR and efficient distance computing algorithms for finite element geometries to enable self-propelled simulations.
Figure 6.10. (a)-(d) Visualization of a self-propelled Ahmed body accelerating from rest at four instances in time. (e) Temporal evolution of the body’s forward velocity. The model has an approximate weight of 661 pounds.
of these more sophisticated models. In Fig. 6.11, we show preliminary results for the signed function computed from finite element representations of the Ahmed body and the SAE Notchback model within IBAMR. It not only seen that the distance function is qualitatively correct, but also the zero contour of the surface level set function $\psi$ conforms nicely to the shape of the Lagrangian structure. This gives us confidence that our initial algorithm is produces accurate results and we plan on testing it on the DrivAer model in the near future.

6.2.5. Future directions

In this chapter, we presented preliminary and ongoing investigations into the challenging problem of turbulence modeling in the immersed boundary framework and simulation of moving automobiles. In the future, we plan to develop additional software infrastructure
and numerical tools to develop a scalable and versatile technique for simulation turbulent fluid-structure interaction of self-propelled bodies. A more rigorous, quantitative validation of the DES model described here and support for wall models is planned as future work.

Computational fluid dynamics is also being used in the renewable energy industry. For example, CFD has played an important role in improving the design of wave energy converters (WEC), which generate electrical energy from ocean waves [249]. Realistic CFD simulations are extremely expensive because of the unsteady nature of the flow field and motion of the WEC on an air-water interface. The solution methodology described in Chapters 2 and 3 can be directly applied to simulate this complex scenario, which will be investigated in the near future.

The methodology described in Chapter 3 enabled simulation of high density ratio fluid-structure interaction but only for relatively simple solid geometries. To simulate more sophisticated structures (such as automobiles and WECs), we need to develop a robust method for handling surface CAD geometries within our solution algorithm. This can be done using the immersed boundary finite element (IBFE) framework, which is available within the IBAMR software [196, 197, 198, 199].

Recall that the level set in the solid region was reinitialized by an analytical expression for the signed distance function. However for CAD geometries, this analytical expression is intractable. Hence, we need a strategy for reinitializing the body’s surface level set function as it undergoes rigid body motion. By leveraging built-in support for triangular elements in IBAMR, we will investigate efficient methods for computing and updating signed distance fields for sophisticated rigid objects.
References


URL http://linkinghub.elsevier.com/retrieve/pii/S0021999113003173


URL http://www.mcs.anl.gov/petsc


URL http://www.mcs.anl.gov/petsc


J. C. Martin, W. J. Moyce, Penney, Part iv. an experimental study of the collapse of liquid columns on a rigid horizontal plane, Phil. Trans. R. Soc. Lond. A 244 (882) (1952) 312–324.


[95] H. Liu, St. anthony falls laboratory, department of mechanical engineering, university of minnesota, personal communication.


URL http://doi.wiley.com/10.1111/faf.12022


URL http://journals.cambridge.org/abstract_S0022112097008392

URL http://dx.plos.org/10.1371/journal.pbio.1002123


URL http://jeb.biologists.org/content/162/1/157.short


URL http://jeb.biologists.org/content/127/1/27.short

URL http://icb.oxfordjournals.org/content/36/6/656.short

URL http://jeb.biologists.org/content/109/1/209.short

URL http://jeb.biologists.org/cgi/doi/10.1242/jeb.00496


