

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Multiphase Flow Simulation: Grid-Based Navier-Stokes-Cahn-Hilliard Coupling and Buoyancy-Driven Phase Separation in the Material Point Method

Permalink

<https://escholarship.org/uc/item/0z20s9jw>

ISBN

9798173719256

Author

Ayazi, Mehrnaz

Publication Date

2026-04-01

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-ShareAlike License, available at <https://creativecommons.org/licenses/by-sa/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Multiphase Flow Simulation: Grid-Based Navier-Stokes-Cahn-Hilliard Coupling and
Buoyancy-Driven Phase Separation in the Material Point Method

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Mehrnaz Ayazi

September 2026

Dissertation Committee:

Professor Tamar Shinar, Co-Chairperson
Professor Craig Schroeder, Co-Chairperson
Professor Ioannis Karamouzas
Professor Christian Shelton

The Dissertation of Mehrnaz Ayazi is approved:

Committee Co-Chairperson

Committee Co-Chairperson

University of California, Riverside

Acknowledgments

0.1 Professional Acknowledgements

I am very grateful to my advisors, Professors Tamar Shinar and Craig Schroeder, who have patiently helped and guided me throughout my time as a graduate student. I have learned invaluable lessons from both of you.

I would like to acknowledge and thank Jason Goulding, Tamar Shinar and Craig Schroeder for their help as co-authors with “Simplified conservative discretization of the Cahn-Hilliard-Navier-Stokes equations” (Chapter 2). I would like to acknowledge and thank Craig Schroeder and Tamar Shinar for their work as co-authors on “Buoyancy-Driven Phase Separation in the Material Point Method” (Chapter 3). I would also like to thank Chen Zhao for her help with rendering the simulations in “Buoyancy-Driven Phase Separation in the Material Point Method” (Chapter 3) and allowing me to use her code to acquire the level-sets for the 3D simulations. I would like to acknowledge and thank Mike “Fox” Liu, Tamar Shinar and Craig Schroeder for their work as co-authors on “Implicit Time Integration for Buoyancy-Driven Phase Separation in the Material Point Method” (Chapter 4).

I would like to thank all my labmates for many conversations and discussions about research, and for their support during my time in the lab.

I would also like to thank Professor Ioannis Karamouzas, and Professor Christian Shelton for serving on my Dissertation Committee and I would like to thank Professor William Grover for serving on my proposal committee.

I would like to thank Craig Schroeder for mentoring as an instructor on record and Grace Caslavka, and Dr. Christian Shelton for giving me the opportunity to teach as an instructor of record during the summer of 2024. The experience I gained teaching CS111 and CS009A has been invaluable and I am grateful for the opportunity to have taught these courses. I would also like to thank you both for your support during the hard times in the last few months of my PhD, and your help with finding a position for me to teach during the summer of 2025. Even though it was not possible for me to accept the position, your kindness is something I will always remember and be grateful for. I would like to thank all my students for being so patient with me as I was navigating the challenges of teaching and learning how to be a better instructor. I would also like to thank all the students who have taken my courses, and I hope that you have learned something valuable from my courses.

I would like to thank Tyler Cohen for connecting me to the department of anthropology at UC Riverside and for giving me the opportunity to become a TA for ANTH 001.

I would like to thank all my students during the 13 quarters I have been a TA and I hope I created a pleasant learning environment for all of you. I am so grateful for the opportunity to have been a TA for so many courses and to have taught so many students. My days in the classroom have been some of the most rewarding and fulfilling days of my time in this program.

I would like to thank Dr. Christian Shelton for his support while I was navigating hardships around funding my PhD program and giving advice on my career. Your will-

ingness to help and meet with me, with your busy schedule, has been invaluable and I am grateful for your support.

I would like to thank Dr. Tamar Shinar for her understanding, for her empathy and for her support especially during the last year of the programs, and both Dr. Craig Schroeder and Dr. Tamar Shinar for our many heart to heart conversations.

I would like to thank Abby Lopez, Rebecca Gaton, Ivana Wu, Vanda Yamaguchi, Grace Caslavka, and all the CSE department staff for all their help during my time at GSA and beyond. It was an honor working with all of you. Thank you for supporting our vision. I would like to thank Nafis Mustakin, Neeha Gajula and Aditya Mohan Gupta for all their help with the GSA events and for being great supporters of all the ideas I had for the GSA.

I would like to thank my team at DreamWorks Animation and intern cohorts during summer 2023 and 2024. I have learned so much from all of you, and I fell in love with my research even more after working with you. I also would like to thank every person who had a coffee chat with me during my time at DreamWorks.

I would like to thank my collaborators Sepide Ahmadian, Angel Pevada Yanez and Benjamin Van-Raalte.

Last but not least, I would like to thank Mike “Fox” Liu for his support during the toughest times of my PhD. For staying up all night to make the code faster, to fix a bug or to implement a new feature. Thank you for being so dedicated to our project, and for your kindness along your professionalism, for teaching me so much about computers and software engineering and always being open for discussion.

0.2 Personal Acknowledgements

A few months ago, when I was writing the first draft of my thesis proposal, I fell into a rabbit hole of reading people's dissertation acknowledgements. I cried for 12 hours straight that night. It might have been the anxiety of not having my report ready or me feeling so much love for so many people and not having a way to express it.

In the past 6 years, there were times when I was not sure if I had what it takes to get a PhD, and there were people along the way who reminded me of who I am. This world is full of wonders, and all I ever wanted to do was learn how to discover them. The past 6 years have taught me more than just computer graphics, physics, mechanics, and all the other things I needed to learn to earn my PhD. They have taught me that we are all in this together. That with patience, you might find those few people who will make traveling 7000 miles worth it.

I have been told through the years that sometimes people have a hard time knowing how I feel about them. I hope those I love do not agree.

I have spent the most difficult years of my life while doing a PhD, and I have countless souls to thank for helping me get through it. I witnessed war in my country, my beloved Tehran being bombed, while I was preparing course material for my students. I witnessed the days I always thought I would spend with my loved ones through a screen: the weddings of women I grew up with, my sisters' teenage years, the funerals of my beloved cousin, and I spent many, many sleepless nights thinking of what would become of our lives when the internet was shut off in my home country. I endured all that with the fear of deportation, fear of dropping out because of the funding cuts, fear of having to endure all

that again because of where I was born. It would have been impossible if I had not had friends who would hold salons during the time of war for me to talk about all this, have I had not had a bug house to go to when I had nowhere to turn, have I had not had a Sepideh to call when I could not get out of bed. But most importantly, if I had not had parents who supported me through this journey. So here it goes: Thank you, Maman and Baba, who have been the source of my strength since I can remember. I know I am strong because I am not afraid to fight for who I am. Thank you for always loving me even at hard times. Thank you for reminding me every day that you love me.

Thank you for believing in me on days no one else believed in me. Thank you for sitting and listening to me when I could do nothing but cry. You are the source of my strength and the kindest souls I have ever known. Baba, so many times I wanted not to choose kindness, and your warm voice in my head would remind me to always choose kindness even if it's the more difficult path. So many times I chose forgiveness because I saw how easily you forgive. Maman, I learned resilience from you. You are the strongest woman I know, and every day since I can remember, you have been the source of inspiration for me. Your intelligence is unbelievable; your kindness is otherworldly. Oh, how my heart shrinks thinking I have not hugged you for four years.

I would like to thank my little sister, Mehrazin. There were countless days when your smile was the only thing that kept me going. Thank you for existing in this world. Every day I try to be a better person and make this world a better place for you. I am so inspired by your intelligence, by your wisdom, by your taste, by your beauty inside and out, and by your kind heart. I love you with all my heart. Mehrnoosh, my oldest friend, my

closest ally, you are the reason I believe in love. You are the reason I do not give up. We grew up together, and I grew so much with you. Thank you for all the meals you made for me. Thank you for all the love you gave me when I felt unloved. Thank you for believing in me and never doubting me for one second. We had ups and downs, but I will always love you. I will always be there for you no matter what. I am so grateful to have you here with me. It's no secret that getting a PhD is not easy, and it's no secret that immigrating is not easy. I would like to thank those who became my second family. Sepide Ahamdian, without whom I could not have endured any of the hardships I went through in the past 10 years. You have been by my side from the dirty alleys of Tehran, Derakhshan, to this new continent. Thank you for being there for me always. I am so grateful for all you add to my life. Hana and Rojin, thank you for being there for me on my toughest days, for sitting with me, listening to me. Rojin, my dear, thank you for traveling to celebrate the end of this journey with me; thank you for our calls on weekends.

Benjamin Van Raalte, I hope you know what a special place you have in my heart. How dear our conversations are to me, how easy it is to spend time with you when I don't have the energy to do anything else. Thank you for opening the doors of Bug House to me, for keeping Salon going when I had little hope for it, for cooking so many amazing dishes for us, for always supporting ideas we have, for always being there for me.

Bethany Tavenner, you are a ray of sunshine in our lives. Thank you for making every hangout so much sweeter, so much more fun. Thank you for always being honest with me, for seeing what others did not see, for always hearing me out, for your wit, your

smile, for the phone calls, for the coworking sessions, for the random hangouts. Thank you for nourishing us with the amazing dishes, and for always having your heart open.

Angel Poveda Yáñez, thank you for your consistent love, your consistent presence, for always showing up in good times and in bad times, for your openness, your honesty, your chaos, and your courage. Thank you for always supporting what I am and who I am and for always seeing the good in me. Thank you for your beautiful looks and your beautiful soul. I see your love in so many ways, and I am so grateful for all you add to our lives.

Jessica Maccaro, what words can describe what I have experienced from the moment you were added to my life? Thank you for Salon, for AEOR, for bug house hangouts, for all the colors you add to our lives. Thank you for the phone calls, for all the tears we shared, for always being honest with me, for your strength, for your ambition, and for reminding me to be proud of how deeply I feel. Thank you for the sardines and vintage smokes. . .

Tobias Moyneur, thank you for all the adventures, for all the talks, for always making me laugh, for all the good conversations, for the game nights, and for your calm. Thank you for pushing me when climbing, to remind me how much I love climbing. Thank you for telling me I am a good climber. And thank you for always welcoming me into your house.

Sasha Karapetrova, thank you for being my first friend in grad school and for being there all the way to the very end. Thank you for taking me climbing and making me go to the gym at 6:30 am.

Evan Sneed, thank you for your love, for showing up every time, for our heart-to-heart conversations, for your directness, and for reminding me of how much I love learning.

Hasti Sharifi, thank you for all the phone calls, for always welcoming me into your life, for support when I was spending the last year of my PhD, and for your support after. Thank you for all the laughter, for reminding me of home.

Khalyd Clay, thank you for being such a good role model for being a good scientist, for being there for me so fiercely through the toughest days. For showing up every time I reached out and didn't, for all our phone calls, for all the miles you drove, for all the climbing we did together, for seeing me, and for your love and support, and for showing me the best food spots in San Diego.

Will Mol, thank you for answering my many questions, for our conversations, for all the stories you have told, for all the salons and your opinions, and for your openness.

And thank you to all who loved me and supported me during this long journey: Sasha Korotneva, Sunya, Capucine, Jae, Pam, my AEOR cohort, and my French classmates.

Last but not least, I would like to thank this land for giving me love, showing me its beauties, and embracing me. Who taught me to be kind even to those unkind to me. I will never forget the days I set foot in Yellowstone, saw the magnificence of the Grand Canyon, and the glory of Yosemite.

To Amirmohammad... I'm sorry I could not be there when you said goodbye to this
world. I will always love you and miss you.

To Maman and Baba, and to the day I be able to hug you again

ABSTRACT OF THE DISSERTATION

Multiphase Flow Simulation: Grid-Based Navier-Stokes-Cahn-Hilliard Coupling and
Buoyancy-Driven Phase Separation in the Material Point Method

by

Mehrnaz Ayazi

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, September 2026
Professor Tamar Shinar, Co-Chairperson
Professor Craig Schroeder, Co-Chairperson

Simulation of fluids can be performed using a background grid (an Eulerian method), particles (a Lagrangian method), or a combination of both (a hybrid method). Simulating multiphase flows requires different approaches depending on how the phases interact. This thesis presents two distinct approaches: a phase-field model in an Eulerian setting, obtained by coupling the Cahn-Hilliard and Navier-Stokes equations, and a buoyancy-driven phase separation scheme in a hybrid setting using the Material Point Method (MPM).

In the phase-field model, a variable describes the concentration of each phase, and its evolution is governed by the Cahn-Hilliard equation, a fourth-order partial differential equation. We present a novel simplified discretization of the Cahn-Hilliard equation coupled with the Navier-Stokes equations describing the flow of the mixture, that solves the fourth-order equation as two coupled second-order Helmholtz equations.

In the second part of the thesis, we introduce a buoyancy-driven phase separation scheme for MPM. MPM is a hybrid method that uses particles to represent the state of

the system and a background grid to solve the equations of motion, and is capable of simulating a wide range of materials. However, it struggles to simulate multiphase flows: a single shared background grid causes what we call the sticking issue, where particles of different phases positioned close together move with similar velocities. Our scheme resolves this by introducing phase-specific background grids, allowing particles of different phases to move independently and separate by density in the presence of gravity. Unlike the Cahn-Hilliard equation, where separation arises from minimizing a free-energy functional via the chemical potential, separation here is driven directly by buoyancy forces.

Finally, we present an implicit time integration scheme for our MPM phase separation method. Explicit schemes usually require smaller time step size for stability. By instead solving a nonlinear system for the state at each time step, the implicit scheme allows for larger time steps, reducing the number of steps needed and resulting in faster simulations overall.

Contents

0.1	Professional Acknowledgements	iv
0.2	Personal Acknowledgements	vii
List of Figures		xviii
List of Tables		xxv
1	Introduction	1
2	Simplified conservative discretization of the Cahn-Hilliard-Navier-Stokes equations	7
2.1	Introduction	8
2.2	Governing equations	12
2.2.1	Cahn-Hilliard Equation	12
2.2.2	Navier-Stokes equations	13
2.3	Discretizations	13
2.3.1	Cahn-Hilliard equation	13
2.3.2	Navier-Stokes equations	18
2.3.3	Spatial Discretization	27
2.3.4	Discretization summary	31
2.4	Analysis	35
2.4.1	Von Neumann stability analysis	35
2.4.2	Truncation error analysis	40
2.5	Numerical examples	42
2.5.1	Manufactured solution	42
2.5.2	Binary separation of phases	45
2.5.3	Discretization stability	45
2.5.4	Stationary circle	47
2.5.5	Horizontal shear	50
2.5.6	Translating circle	53
2.5.7	Rising air bubble	57
2.5.8	Comparisons of surface tension	60
2.5.9	Rayleigh-Taylor instability	65

2.5.10	Dam break	67
2.5.11	Equilibrium Drop	74
2.6	Conclusion	75
2.7	Acknowledgements	76
3	Buoyancy-driven Phase Separation in the Material Point Method	77
3.1	Introduction	78
3.2	Related Work	82
3.2.1	The material point method	82
3.2.2	Multiphase MPM formulations	84
3.2.3	Multi-fluid and phase separation models outside MPM	86
3.2.4	Multi-fluid level set reconstruction	87
3.3	Method	89
3.3.1	Particle to Grid Transfer	89
3.3.2	Grid Forces	89
3.3.3	Emergence of buoyancy	90
3.3.4	Grid to Particle Transfer	91
3.3.5	Phase separation force	92
3.3.6	Projection Method	97
3.4	Results	100
3.4.1	Stable test	101
3.4.2	Partially mixed test	101
3.4.3	Rayleigh-Taylor instability	104
3.4.4	Dam break test	105
3.4.5	Fully mixed test with phase separation force	108
3.4.6	Rayleigh-Taylor instability with phase separation force	110
3.4.7	Discussion	110
3.4.8	Generalization to $N > 2$ phases	111
3.5	Conclusion and Limitations	112
4	An implicit time integration scheme for the multi-phase Material Point Method	113
4.1	introduction	114
4.2	Related Work	116
4.2.1	Material Point Method	116
4.2.2	Multiphase MPM formulations	117
4.2.3	Multiphase flow simulation outside MPM	117
4.2.4	Implicit time integration	118
4.2.5	Particle spacing methods	119
4.3	Method	120
4.3.1	Particle-to-grid transfer	121
4.3.2	Grid-to-particle transfer	121
4.3.3	Implicit integration of grid forces	122
4.3.4	Particle distancing force	125
4.3.5	Volume loss	129

4.4	Results	132
4.4.1	Stable state test	134
4.4.2	Partially mixed test	136
4.4.3	Rayleigh-Taylor test	138
4.4.4	Dam break test	144
4.4.5	Drop test	148
4.5	Conclusion and Limitations	150
5	Conclusion	152

List of Figures

1.1	Graph of Free energy density ($F(\gamma)$) vs. the material concentration (γ) . . .	3
1.2	Rayleigh–Taylor instability using the standard material point method. Left to right: $t = 0, 1, 2, 8, 16, 32s$	6
2.1	$g(\gamma)$ vs γ for different values of a	23
2.2	Example path $\gamma_0 \rightarrow \gamma_N$. The triangles represent the pressure jumps along this path.	24
2.3	The MAC grid stores velocities on faces (represented by triangles), and γ and pressure on cell centers (circles).	27
2.4	The components of the momentum flux for updating the x component of momentum in 2D are shown in 2.4a. The momentum at the red triangle will be updated with the fluxes at the surrounding blue and orange dots. The u and $\mathbf{m}^x$ values at these dots are interpolated from neighboring values as illustrated in 2.4b and 2.4c, respectively.	28
2.5	Binary fluid separation over time	45
2.6	Numerically determined stable time step size compared with the analytically predicted for (left) linearized problem and (right) the fully nonlinear problem. The stable time step size does not depend significantly on resolution and is in close agreement with the analytical prediction.	46
2.7	L_2 and L_∞ errors of velocity in the stationary circle test.	49
2.8	L_2 and L_∞ errors of velocity in the stationary circle test with fixed interface width.	49

2.9	Horizontal Shear (Case 3).	51
2.10	Results of the horizontal shear test for Cases 1-4.	52
2.11	Artifacts in horizontal shear Case 1, and the effects of different parameters and methods on those artifacts.	54
2.12	Initial and final ($t = 1$) state of the drop with $\sigma = 10^{-12}$. The solid black line is the initial state and the dotted red line is the final state.	56
2.13	Initial and final ($t = 1$) state of the drop with $\sigma = 1$. The solid black line is the initial state and the dotted red line is the final state.	57
2.14	Rising bubble test results with $X_\eta = 0, X_M = 1$	60
2.15	Rising bubble test results of test described in table 2.8 with $N = 256$	61
2.16	Comparisons of the angle between surface tension force and surface normal (radians) for different surface tension formulations.	63
2.17	Comparisons of the magnitude of surface tension force for different surface tension formulations.	64
2.18	Change in pressure in the x axis at $y = 0.5$ of Section 2.5.8.	66
2.19	Rayleigh-Taylor test at times $t = 0, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5$ and density ratio of 3.	68
2.20	Rayleigh-Taylor test results with $\rho_0 = 3$, from $t = 2.25$ to $t = 8$ in 0.25 increments.	69
2.21	Rayleigh-Taylor simulation with density ratio of 30, 1000 and 3000 at times $t = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75$	70
2.22	Location of the maximum and minimum x of the interface in the Rayleigh-Taylor test with density ratio of 3.	71
2.23	Fluid height and front over time in the dam break test.	71
2.24	The evolution of the dam break test with water and air from $T = 0$ to $T = 5$ in 0.5 increments.	72
2.25	The evolution of the dam break test with water and air from $T = 5$ to $T = 9.5$ in 0.5 increments.	73

2.26	Demonstration of the effects of the contact angle θ on a drop at rest.	75
3.1	An overview of our method. (a) Masses and velocities are transferred from particles to <i>separate phase</i> grids. (b) Gravity is applied to the separate phase velocity fields. (c) Constitutive model forces are computed on a combined grid using all particles. (d) The constitutive forces are distributed to the separate phase velocity fields proportional to the phase volume. (e) Grid phase velocities are transferred back to particles. (f) Grid phase velocities are combined into a unified grid velocity. (g) The unified grid velocity is used to update deformation gradients for all particles. (h) Particles are moved and the next time step begins.	81
3.2	(a) Initial condition for a partially mixed domain with lighter fluid on top and denser fluid on the bottom with a mixed region in the middle. (b) Expected phase separation: lighter particles rise and denser fluids sink in the mixed regions, with little movement elsewhere. (c) Behavior observed by [77]: this method preserves the volume of each phase, causing all light particles to rise, which results in an overall volume increase.	85
3.3	The traditional MPM algorithm (a) transfers mass and velocity from particles to grid, (b) applies gravity, (c) computes constitutive forces, (d) and applies them to velocities. Finally (e) particles are updated and (f) moved.	88
3.4	An overview of the method of [77]. (a) Velocities are transferred from particles to <i>unified</i> grids, then (c) copied to per-phase grids. Masses are transferred to (a) unified and (b) per-phase grids. (d) Constitutive model forces are computed on a combined grid and (e) distributed to the separate phase velocity grids with gravity. (f) Grid phase velocities are transferred back to particles, deformation gradient is updated, and (g) particles are moved. . .	88
3.5	Rayleigh–Taylor instability. Rows, top to bottom: our method, our method with phase separation force, standard MPM, and Li and Liu [77]. Left to right: $t = 0, 1, 2, 8, 16, 32s$. Our method successfully unmixes the phases, while the others do not. Our phase separation force promotes small scale clumping while maintaining larger scale dynamics.	93
3.6	Rayleigh–Taylor instability. Initial state followed by our method (middle) and Li and Liu [77] (right) at $t = 0, 1, 2, 16s$. Our method sharply separates the two phases, alleviating the residual mixing evident with previous approach.	94
3.7	Fully mixed domain <i>without</i> gravity with density ratio 10, including our phase separation force in 2D at $t = 0, 2, 8, 10s$. row 1: $c = 50$. row 2 : $c = 100$ and row 3: $c = 1000$ The phases clump into small subregions but do not separate vertically. Larger values for c result in faster separation.	96

3.8	Level set projection. Per-phase level sets are shown with solid contours and the combined level set with a dashed contour. (Left) Initial overlapping state of the multiple level sets. Background colors represent cases in our level set projection method: Case 1 (blue), Case 2 (green), Case 3 (yellow), and Case 4 (red). Comparison of our level set projection method (middle) with a the method of [85] (right). Leaving the combined level set fixed during projection avoids artifacts near triple junctions.	97
3.9	Projection map used to enforce consistency between the ordered level sets (ϕ_k, ϕ_ℓ) with $\phi_k \leq \phi_\ell$ and $\phi < 0$. The red polyline indicates the feasible set of consistent outputs (ϕ'_k, ϕ'_ℓ) . Given an input point (ϕ_k, ϕ_ℓ) , the projection algorithm computes a nearby feasible point (ϕ'_k, ϕ'_ℓ) . Selecting the nearest feasible point would produce a discontinuity at the dashed line. Instead, input space is divided into cases (Cases 2–4, indicated by the colored regions; Case 1 corresponds to $\phi \geq 0$ and is not shown). Cases 2 and 3 map to the closest feasible point. Points in Case 4 are moved to the feasible line along the direction of the arrows.	98
3.10	Lighter fluid on top of a denser fluid using our method (top row), standard MPM (middle row), and Li and Liu [77]’s method (bottom row) at $t = 0, 5, 10s$ from left to right.	102
3.11	Lighter fluid on top of a denser fluid with a mixed region in the middle. Our method (top row), standard MPM (middle row), and Li and Liu [77]’s method (bottom row) at $t = 0, 2, 4, 16s$ from left to right. Our method successfully unmixes the phases, while the other two do not.	103
3.12	Histogram of bin height vs. phase particle count for 100 bins from $y = 0$ to $y = 3$ of the Rayleigh-Taylor instability test at time $t = 32s$. Our method (3.12a) exhibits phase separation, while mixing is apparent with standard MPM (3.12b) and Li and Liu [77]’s method (3.12c).	104
3.13	3D dam-break test with denser fluid initially above lighter fluid. Row 1: our method at $t = 0, 4, 8s$. Row 2: standard MPM at $t = 0, 4, 8s$. Row 3: Li and Liu [77] at $t = 0, 4, 8s$. Our method robustly separates the phases whereas the other two do not.	105
3.14	Dam break test with denser fluid on top and lighter fluid on the bottom using our method (top row), the standard MPM (middle row) and Li and Liu [77]’s method (bottom row) at $t = 0, 0.25, 1, 16s$ from left to right. Our method gives sharp separation of phases, whereas evident mixing persists using the other two approaches.	106

3.15	We reproduce the example from Li and Liu [77] with the phases initially offset, giving comparable results with their method. Our method (top row) shows good separation. Standard MPM (middle row) maintains significant mixing at the interface. Li and Liu [77]’s method (bottom row) exhibits some residual mixing with ongoing slow separation. Left to right: $t = 0, 0.5, 1, 16, 62.5s$	107
3.16	Fully mixed domain under gravity with density ratio 10 using our method at $t = 0, 2, 4, 8, 16, 32s$ left to right, top to bottom.	108
3.17	3D Fully mixed domain test under gravity with density ratio 10, using our method at $t = 0, 0.5, 1, 8s$. Our method fully unmixes the phases.	109
3.18	Fully mixed domain <i>without</i> gravity with density ratio 10, including our phase separation force, using our method at $t = 0, 4, 16s$ in 3D. The phases clump into small subregions but do not separate vertically.	109
4.1	A stable state test with lighter fluid on top of a heavier fluid at times $t = 0, 5, 10s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), and phase-separating MPM with implicit time integration and the particle distancing force (fourth row).	135
4.2	A partially mixed test with lighter fluid on top of a heavier fluid with a mixed region in the middle at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).	137
4.3	Rayleigh-Taylor test with denser fluid on top of a heavier fluid with a mixed region in the middle at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).	139
4.4	Rayleigh-Taylor test with implicit time integration scheme without the particle distancing force (first row) and with the particle distancing force (second row) at times $t = 0.8, 1.6, 2.4s$. Both form patterns of high particle density particle as the fluids start to mix creating non-physical behavior.	140

4.5	Rayleigh–Taylor instability test utilizing the implicit time integration scheme with a particle distancing coefficient of $c = 50$ at $t = 1, 5, 20s$. A distinct sand-like behavior is observed during the settling phase; the fluid motion becomes less dynamic as the particle distancing potential becomes the dominant contributor to the total potential of the system.	141
4.6	Rayleigh-Taylor test with explicit time integration scheme (first row) and with implicit time integration scheme with $\Delta t = 2.5 \times 10^{-5}$ (second row) at times $t = 0, 1, 2, 4, 8s$. $\Delta t = 2.5 \times 10^{-5}$ is the average time step size used in the explicit scheme for this specific test. Both schemes show similar development when they use similar time step sizes.	142
4.7	Rayleigh-Taylor test with implicit time integration scheme and Young’s modulus of $E = 10^5$ (first row) and Young’s modulus of $E = 10^7$ (second row) at times $t = 0, 2.5, 5, 10s$. The test with higher stiffness shows less volume loss.	143
4.8	Dam break test with lighter fluid on top of a heavier fluid at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).	145
4.9	Dam break test with lighter fluid on top of a heavier fluid with an offset to the right at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).	146
4.10	Dam break test with lighter fluid on top of a heavier fluid with an offset to the right at times $t = 0, 10, 20, 40, 60s$ from left to right using the baseline [6] method without the particles distancing force (top row), and with the particles distancing force (second row). The test case with the particle distancing force maintains its volume compared to the test case without the particle distancing force, which shows a loss of volume as the simulation progresses.	147
4.11	a drop of lighter fluid fall into a pool of heavier fluid. Times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).	149

4.12	Computational efficiency comparison demonstrating execution time per frame across scaling mesh grid resolutions (16^2 to 512^2) for implicit and explicit multi-phase formulations. All simulations were run using 8 threads and for 10 frames with maximum Δt of 2×10^{-3}	150
------	--	-----

List of Tables

2.1	Analysis of our Cahn-Hilliard solver at $t = 1$ demonstrates second order convergence.	44
2.2	Analysis of our Cahn-Hilliard-Navier-Stokes solver at $t = 1$ demonstrates second order convergence.	44
2.3	Material settings for different cases of the stationary circle test.	49
2.4	Material settings for different cases of the horizontal shear.	52
2.5	Material settings for different cases of the translating circle test (surface tension, density).	55
2.6	Error results for the advection test without surface tension.	55
2.7	Error results for the advection test with surface tension.	56
2.8	L_2 error and order compared to $N = 256$ on all the cases for rising air bubble test.	62
2.9	Material settings for different cases for the surface tension and pressure comparison test.	62
3.1	Timing comparison of methods in 2D and 3D. Entries marked “-” indicate that the method was not evaluated. Initial elasticity values in dambreak 1 differ from other tests to match the results reported by Li and Liu [77].	100

4.1	Timing comparison of methods in 2D and 3D. Entries marked “-” indicate that the method was not evaluated. Initial elasticity values in dambreak 1 differ from other tests to match the results reported by [6]. PD indicates that the particle distancing force was used in addition to the implicit time integration scheme.	132
-----	---	-----

Chapter 1

Introduction

Fluids are complex, difficult to comprehend, and challenging to simulate, much like human beings. Just as our human interactions are profoundly intricate, the trajectory of our lives (a Lagrangian perspective of existence) is shaped both by our intrinsic characteristics and by the environment around us. Sometimes we possess all the inherent qualities needed to reach our desired destination, much like buoyant oil particles in a simulation, yet external conditions restrict us. Even though these oil particles are lighter and should naturally rise, their rising is blocked. Not because they lack the physical capacity to rise, but because the structural constraints of the simulated world they inhabit (such as the shared background grid in the Material Point Method) will not allow it.

A diverse array of numerical methods exists for simulating different physical phenomena [4, 12]. Depending on the governing equations and the specific problem being solved, such as whether the medium is a solid or a fluid, the presence of viscosity, or the expectation of large deformations, different methods offer distinct advantages. Some meth-

ods excel in extreme conditions or large deformations, while others prioritize mathematical exactness, computational efficiency, or generalizability across different physical states. The same considerations apply when simulating multiphase mixtures.

In thermodynamics, a phase is defined as a region of space throughout which all physical properties of a material are essentially uniform [18]. This uniform state can refer to the material’s physical state of matter (e.g., solid, liquid, gas) or its specific chemical composition.

A multiphase mixture is a combination of two or more distinct phases. For example, a mixture of oil and water constitutes a multiphase mixture comprising an oil phase and a water phase [59]. In this thesis, our references to multiphase mixtures specifically focus on multi-fluid mixtures. In such immiscible mixtures, phase separation can occur due to a variety of physical driving forces.

In physical systems, fluids separate through several distinct mechanisms. Thermodynamic incompatibility often drives this process; for example, polar water molecules and non-polar oil molecules do not readily mix, leading to the formation of distinct macroscopic regions [60]. Phase separation can also be driven by buoyancy, where density differences in a gravitational field cause the less dense fluid (e.g., oil) to float above the denser one (e.g., water) [15]. Furthermore, surface tension plays a critical role, as molecules within the same phase energetically favor bonding with one another over molecules of a differing phase. Variations in temperature and pressure, or shifts in chemical potential resulting from the mixing of materials, can also induce phase separation.

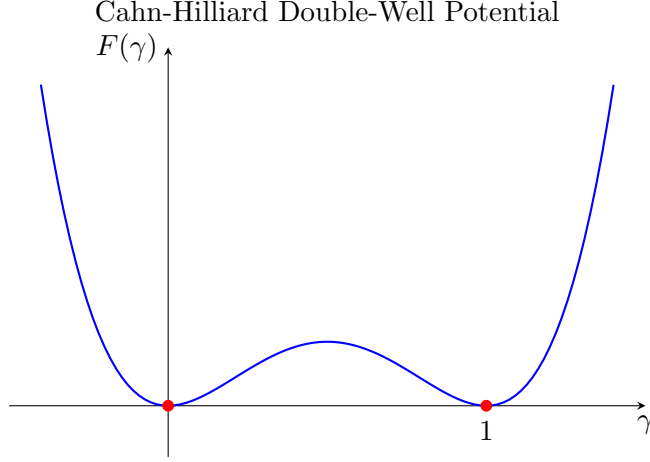


Figure 1.1: Graph of Free energy density ($F(\gamma)$) vs. the material concentration (γ)

Simulating multiphase mixtures remains a significant challenge in computational physics due to the complex interfacial interactions between the phases and the diverse physical processes governing their separation. In this thesis, we focus on two distinct types of multiphase separation. The first is driven by chemical potential and is described by the Cahn-Hilliard equation, often used to model phase separation in binary mixtures. The second type is buoyancy-driven separation, which is utilized to model the macroscopic separation of immiscible fluids like oil and water.

We first focus on simulating phase separation in binary mixtures driven by chemical potential. The Cahn-Hilliard equation is a fourth-order nonlinear partial differential equation derived from the free energy functional of the system. It tracks the time evolution of a scalar order parameter representing the concentration of one of the mixture's components, and it is widely used in material science and physics to describe the evolution of phase separation [17, 28]. This evolution is driven by the chemical potential, which dictates the system's tendency to minimize its free energy. The chemical potential is typically mod-

eled using a double-well potential function (Figure 1.1) with minima at or near the pure concentrations of each separate phase, causing the mixture to spontaneously separate into distinct regions.

To model the fluid flow of the mixture, the Cahn-Hilliard equation is often coupled with the Navier-Stokes equations, forming the Cahn-Hilliard-Navier-Stokes (CHNS) system [62]. In Chapter 2, we present a novel simplified, conservative discretization of the CHNS equations. Our model conserves both mass and momentum while maintaining consistency between the two. We demonstrate that our method achieves second-order accuracy while being significantly simpler to solve than traditional approaches.

The Material Point Method (MPM) [124] is a widely adopted framework for simulating a diverse range of materials, including solids, fluids, and granular media. However, it traditionally struggles with multiphase mixtures. In Chapter 3, we present a novel MPM scheme designed to simulate buoyancy-driven phase separation in immiscible fluids. We specifically address the “sticking” issue caused by the shared background grid in standard MPM. Because particles with different physical properties and phases map to the same grid nodes, they are subjected to unified velocity updates based purely on their spatial proximity to the background grid [111]. The single shared velocity field artificially prevents nearby particles of differing phases from moving in opposing directions, resulting in persistent mixing in scenarios such as the Rayleigh-Taylor instability (Figure 1.2). We introduce a new scheme that updates particle velocities based on their phase identities. We demonstrate that our method successfully simulates buoyancy-driven phase separation in immiscible fluids.

Finally in Chapter 4, we present a an implicit time integration scheme for the multi-phase Material Point Method presented in Chapter 3. Usually to solve equations of motions and in general differtial equations, you can either solve them explicittly or implicitly. An explicit scheme means you solve for your future state based on your current or past states. An example of this is the Forward Euler method where we solve a differential equation of the form $\dot{y} = f(t, y)$ by using the current state y_n to compute the next state y_{n+1} as $y_{n+1} = y_n + \Delta t f(t_n, y_n)$. In contrast, an implicit scheme involves solving for the future state based on both the current and future states. An example of this is the Backward Euler method where we solve for y_{n+1} using the equation $y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1})$. Implicit methods are generally more stable, especially for stiff equations, but they require solving a system of equations at each time step, which can be computationally intensive. In this chapter, we develop an implicit time integration scheme for MPM that allows for larger time steps. We show that the symmetry of the explicit multi-phase MPM scheme allows to solve the state of the system as a minimization problem in which we try to minimize the energy of the system.

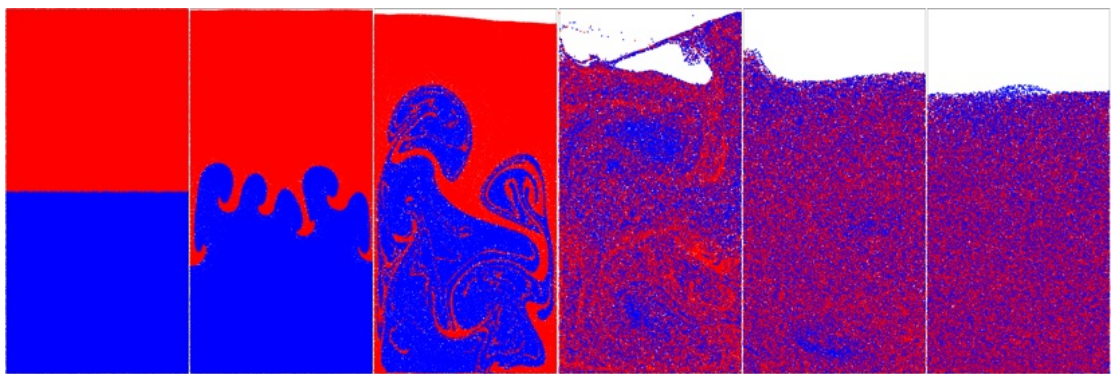


Figure 1.2: Rayleigh–Taylor instability using the standard material point method. Left to right: $t = 0, 1, 2, 8, 16, 32s$.

Chapter 2

Simplified conservative discretization of the Cahn-Hilliard-Navier-Stokes equations

In this chapter we construct a novel discretization of the Cahn-Hilliard equation coupled with the Navier-Stokes equations. The Cahn-Hilliard equation models the separation of a binary mixture. We construct a very simple time integration scheme for simulating the Cahn-Hilliard equation, which is based on splitting the fourth-order equation into two second-order Helmholtz equations. We combine the Cahn-Hilliard equation with the Navier-Stokes equations to simulate phase separation in a two-phase fluid flow in two dimensions. The scheme conserves mass and momentum and exhibits consistency between mass and

momentum, allowing it to be used with large density ratios. We introduce a novel discretization of the surface tension force from the phase-field variable that has finite support around the transition region. The model has a parameter that allows it to transition from a smoothed continuum surface force to a fully sharp interface formulation. We show that our method achieves second-order accuracy, and we compare our method to previous work in a variety of experiments.

2.1 Introduction

Phase-field models have been widely used to study many problems involving multiphase incompressible flows, including two-phase flows in a microtube [50], viscoelastic flows [146, 147], multiphase flows with soluble surfactants [83, 128, 152], phase separation and coarsening [153, 7], jet pinch-off [145, 71], multicomponent systems with three or more phases [72, 69, 139], tumor growth [136], microfluidics [108], and other physical phenomena. In these methods, the phase field represents the concentration of one of the components of the mixture, and fluid interfaces are represented by a smooth transition region between bulk regions of the different phases. One popular choice for the governing equation for the evolution of the phase field is the Cahn-Hilliard equation, which acts to minimize the free energy of the system, leading the mixture to segregate into separate regions [17, 28].

Many methods have been developed for the numerical simulation of the Cahn-Hilliard equation, including Fourier spectral methods [153, 51, 76], finite difference methods [35, 79, 46, 20, 57], and finite element methods [29, 13, 11, 32, 135, 49]. For example, Zhu et al. [153] examines properties of coarsening using a Fourier spectral solver, and He

et al. [51] developed a Fourier spectral solve which allows for large time steps. Schemes supporting adaptive time step sizes were presented in Li et al. [80] and the second-order accurate Guillén-González and Tierra [44]. Guillén-González and Tierra [43] presented two linear schemes derived through special treatment of the double-well potential function. Lee et al. [76] gives an overview of the mathematical derivation of the Cahn-Hilliard equation, discusses several methods for its numerical solution, and provides a Matlab implementation for a pseudospectral method.

A model of immiscible two-phase flows coupling the Cahn-Hilliard equation to the incompressible Navier-Stokes equations for the case of matching densities was described in Hohenberg and Halperin [52]; a derivation from first principles was given in Gurtin et al. [47]. Badalassi et al. [8] devised numerical simulations of this coupled system. For varying densities, some studies have used the Boussinesq approximation where the momentum equation is solved using a background density and the variable density appears only in additional buoyancy terms [61, 82, 7]. The more general case of the Cahn-Hilliard-Navier-Stokes (CHNS) equations with different densities and viscosities presents additional challenges, and Shen and Yang [113] devised a model which admits an energy law and developed an associated projection-type [21] numerical scheme. Shen and Yang [112] developed an unconditionally energy-stable numerical scheme which reduces the problem to the solution of a sequence of decoupled elliptic solves. Phase-field models of two-phase incompressible flow with large density ratios were simulated in Kim [68] and Ding et al. [24], which involved the solution of time-dependent variable coefficient linear systems. Dong and Shen [27] devised a scheme for large density and viscosity differences with improved efficiency through the use of constant

coefficient matrices. Yue et al. [146] coupled the Cahn-Hilliard equation to non-Newtonian flows. Guo et al. [46] devised a quasi-incompressible scheme with mass conservative and energy stability for the CHNS equations. Kim [71] provides a review of phase-field models and numerical methods for their approximation.

Consistency of mass and momentum transport was first addressed on the discrete level for two-phase interfacial flows in volume-tracking methods [16, 100] and the level set method [102]. For the coupled Cahn-Hilliard-Navier-Stokes equations, Huang et al. [57] formulated a set of consistency conditions in the continuous setting and furthermore devised a second-order accurate finite difference scheme with numerical consistency of mass and momentum flux. Huang et al. [56] similarly addressed consistency of mass and momentum transport in the Allen-Cahn equation and also devised a conservative redistribution scheme to maintain the physical bounds of the concentration.

In this work, we construct a novel conservative, second order accurate finite difference scheme for the simulation of the Cahn-Hilliard-Navier-Stokes equations for two phase flows with large density and viscosity differences. Following Huang et al. [57], we enforce discrete consistency of the mass and momentum fluxes. In contrast to Huang et al. [57], which used collocated grids for pressure, concentration, and velocity, our method uses a staggered MAC grid [48] to conveniently solve the pressure Poisson equation. We construct a very simple time integration scheme for simulating the Cahn-Hilliard equation, which is based on splitting the fourth-order equation into two second-order Helmholtz equations. Our splitting results in the solution of two Helmholtz equations with identical coefficient matrices. In our splitting, we use a stabilizing term introduced in Shen and Yang [112, 113].

The Cahn-Hilliard equation is coupled to the Navier-Stokes equations in part through the surface tension force, which depends on the gradient of the concentration phase field and the geometric curvature of the interface. Kim [68, 71] and Acar [2] discuss several models of surface tension for phase-field multiphase flows. The numerical simulation of surface tension can be challenging due to stability and accuracy issues including the formation of spurious currents about the interface. One common treatment of surface tension is the Continuum Surface Force (CSF) approach introduced in Brackbill et al. [14], where typically a discrete approximation to a smoothed delta function is used to spread the surface tension force over a small number of grid cells about the interface. Kim [70] used a continuum surface force combinations for more than two fluid phases. Another set of approaches treats the interface in a sharp manner, capturing the pressure jumps across the interface that balance the surface tension force [67]. Francois et al. [33] enforced exact force balance at the interface for both CSF and sharp interface schemes, eliminating one of the sources of spurious currents at the interface. In this work, we propose a novel discretization of surface tension that gives exact force balance at the interface and unifies the CSF and sharp interfaces schemes. A similar relationship between the CSF approach and the sharp interface approach was noted by Lalande et al. [75] for level set methods. Our method is based on taking the gradient of a smoothed Heaviside approximation, similar to CSF approaches. Notably, the degree of smoothing is controlled by a parameter and the sharp interface method is directly recovered as the smoothing is decreased.

2.2 Governing equations

2.2.1 Cahn-Hilliard Equation

The Cahn-Hilliard equation, initially proposed by Cahn and Hilliard [17] and widely utilized to describe the phase separation process in a mixture of two fluids, is expressed as follows:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{u}\gamma) = \nabla \cdot (M\nabla \xi) + S_\gamma \quad (2.1)$$

$$\xi = \lambda(F'(\gamma) - \nabla^2 \gamma) \quad (2.2)$$

$$F(\gamma) = \frac{1}{\eta^2} \gamma^2 (1 - \gamma)^2 \quad (2.3)$$

$$\lambda = \frac{3}{2\sqrt{2}} \sigma \eta. \quad (2.4)$$

The phase-field function $\gamma \in [0, 1]$ is the volume fraction of fluid phase 1 (with density ρ_1 and viscosity μ_1). $1 - \gamma$ is the volume fraction of fluid phase 0 (with density ρ_0 and viscosity μ_0). $\mathbf{u}$ is the velocity, M is the mobility, ξ is the chemical potential, S_γ is the source term for γ , λ is the mixing energy density, F is the double well potential function, F' is the derivative of F with respect to γ , η characterizes the thickness of the transition region between the two phases, and σ is the surface tension coefficient. The effective density and viscosity at a certain location is given by

$$\rho = \rho_0 + (\rho_1 - \rho_0) \gamma \quad (2.5)$$

$$\mu = \mu_0 + (\mu_1 - \mu_0) \gamma. \quad (2.6)$$

2.2.2 Navier-Stokes equations

The incompressible Navier-Stokes equations are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{m} = S_m \quad (2.7)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2.8)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\mathbf{m} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot (\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) + \rho \mathbf{g} + \sigma \kappa \nabla(g(\gamma)) + S_u. \quad (2.9)$$

σ , ρ , and $\mathbf{u}$ are as defined in Section 2.2.1, $\mathbf{m}$ is the mass flux, S_m represents an external source of mass, $\sigma \kappa \nabla g(\gamma)$ is the surface tension force, κ is the estimated interface curvature, $g(\gamma)$ is a regularized Heaviside function (see Section 2.3.2), p is the pressure, $\mathbf{g}$ is gravity, and S_u is the source term for momentum. Similar to [57], the mass flux $\mathbf{m}$ can be inferred from the γ evolution equation (2.1) and equations (2.7) and (2.5). Defining $S_m = (\rho_1 - \rho_0) S_\gamma$, it can be shown that a consistent definition of $\mathbf{m}$ is $\mathbf{m} = \rho \mathbf{u} - (\rho_1 - \rho_0) M \nabla \xi$. This is discussed in detail in Section 2.3.2.

2.3 Discretizations

In this section we describe the temporal and spatial discretizations of the Cahn-Hilliard and Navier-Stokes equations.

2.3.1 Cahn-Hilliard equation

Although the overall form for the splitting of the CH equations follows the same general strategy as prior discretizations, we achieve a discretization that is significantly sim-

pler than existing discretizations while not sacrificing desirable features of the discretization. Our discretization of the Navier-Stokes equations was designed to be as similar to a standard second-order MAC-grid discretization as possible while still achieving favorable numerical properties when coupling with the Cahn-Hilliard equations.

Cahn-Hilliard equation temporal discretization

We begin by combining (2.1) and (2.2) as the base Cahn-Hilliard equation to discretize.

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{u} \gamma) = M \lambda \nabla^2 (F'(\gamma) - \nabla^2 \gamma) + S_\gamma \quad (2.10)$$

We discretize the time derivative as $\frac{\alpha \hat{\gamma}^{n+1} - \alpha \gamma^{BD}}{\Delta t}$. To achieve second-order accuracy, we use a backwards-difference scheme with $\alpha = \frac{3}{2}$ and $\gamma^{BD} = \frac{4}{3} \gamma^n - \frac{1}{3} \gamma^{n-1}$. For the first time step we use a first-order single-step scheme with $\alpha = 1$ and $\gamma^{BD} = \gamma^n$. Explicit quantities for $\mathbf{u}$ and γ are discretized using an Adams-Bashforth scheme, where $\gamma^{AB} = \gamma^n$, $\mathbf{u}^{AB} = \mathbf{u}^n$ in the first order case and $\gamma^{AB} = 2\gamma^n - \gamma^{n-1}$, $\mathbf{u}^{AB} = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ in the second order case. Since in practice α and Δt always occur together, we let $\Delta t = \alpha \widehat{\Delta t}$. $\hat{\gamma}^{n+1}$ represents an estimate of the final γ obtained by solving this discretized equation; it will be later updated to the final γ^{n+1} bounded to the range $[0, 1]$ using the method described in Section 2.3.1.

Following Shen and Yang [113], we introduce a term $2\tau (\hat{\gamma}^{n+1} - \gamma^{AB})$ to help with splitting the equation. The choice of τ is discussed below. Since both $\hat{\gamma}^{n+1}$ and γ^{AB} are

estimates of γ^{n+1} , this term vanishes rapidly under refinement.

$$\frac{\hat{\gamma}^{n+1} - \gamma^{BD}}{\widehat{\Delta t}} + \nabla \cdot (\mathbf{u}^{AB} \gamma^{AB}) = M\lambda \nabla^2 (F'(\gamma^{AB}) - \nabla^2 \hat{\gamma}^{n+1} + 2\tau (\hat{\gamma}^{n+1} - \gamma^{AB})) + S_\gamma^{n+1} \quad (2.11)$$

Next we split out the advection term as an update from γ^{BD} to γ^* :

$$\frac{\gamma^* - \gamma^{BD}}{\widehat{\Delta t}} + \nabla \cdot (\mathbf{u}^{AB} \gamma^{AB}) = 0 \quad (2.12)$$

$$\frac{\hat{\gamma}^{n+1} - \gamma^*}{\widehat{\Delta t}} = M\lambda \nabla^2 (F'(\gamma^{AB}) - \nabla^2 \hat{\gamma}^{n+1} + 2\tau (\hat{\gamma}^{n+1} - \gamma^{AB})) + S_\gamma^{n+1}. \quad (2.13)$$

We group all explicit terms under the Laplacian into a new term A :

$$A = \frac{1}{\tau} F'(\gamma^{AB}) - 2\gamma^{AB} \quad (2.14)$$

$$\frac{\hat{\gamma}^{n+1} - \gamma^*}{\widehat{\Delta t}} = M\lambda \nabla^2 (\tau A - \nabla^2 \hat{\gamma}^{n+1} + 2\tau \hat{\gamma}^{n+1}) + S_\gamma^{n+1} \quad (2.15)$$

Grouping together $\hat{\gamma}^{n+1}$ with $\nabla^2 \hat{\gamma}^{n+1}$ on both sides of the equation:

$$\frac{(\hat{\gamma}^{n+1} - \tau M\lambda \widehat{\Delta t} \nabla^2 \hat{\gamma}^{n+1} + A) - \gamma^*}{\widehat{\Delta t}} = \tau M\lambda \nabla^2 \left(\hat{\gamma}^{n+1} - \frac{1}{\tau} \nabla^2 \hat{\gamma}^{n+1} + A \right) + \frac{A}{\widehat{\Delta t}} + S_\gamma^{n+1}. \quad (2.16)$$

For the two parenthesized expressions to match, we need $\tau^2 M\lambda \widehat{\Delta t} = 1$. We achieve this by letting

$$s = \sqrt{\frac{M\lambda}{\widehat{\Delta t}}} \quad \tau = \frac{1}{M\lambda} s \quad (2.17)$$

and define

$$\gamma' = \hat{\gamma}^{n+1} - s\widehat{\Delta t}\nabla^2\hat{\gamma}^{n+1} + A. \quad (2.18)$$

With these definitions, equation (2.16) becomes

$$\frac{\gamma' - \gamma^*}{\widehat{\Delta t}} = s\nabla^2\gamma' + \frac{1}{\widehat{\Delta t}}A + S_\gamma^{n+1}. \quad (2.19)$$

Observe that (2.18) and (2.19) are implicit Helmholtz equations for $\hat{\gamma}^{n+1}$ and γ' , respectively. Note that while differing from the stabilizing term introduced in Shen and Yang [113], our choice of τ results in a conditionally stable scheme, as shown in Section 2.4.1. With our choice of τ , the overall term is formally 1.5 order accurate (although within the resolutions studied below, our method exhibits second-order convergence). While the first-order stabilizing scheme in Shen and Yang [113] uses the difference $\gamma^{n+1} - \gamma^n$, we also note that [113] additionally introduced a second-order stabilizing term based on $\gamma^{n+1} - \gamma^{AB}$ as part of a second order scheme.

Cahn-Hilliard bounds-respecting redistribution scheme

When solving the Cahn-Hilliard equation, numerical errors or sufficiently large time steps can result in γ outside the range $[0, 1]$. Given a large density ratio between the fluids, this can result in negative densities from (2.5). In order to prevent this, we use a redistribution scheme for γ as detailed in Huang et al. [56] adapted for the range $[0, 1]$.

First $\hat{\gamma}^{n+1}$ is clamped to the feasible range at each index i, j :

$$\gamma_{i,j}^{b*} = \begin{cases} 1, & \text{if } \hat{\gamma}_{i,j}^{n+1} \geq 1 \\ 0, & \text{if } \hat{\gamma}_{i,j}^{n+1} \leq 0 \\ \hat{\gamma}_{i,j}^{n+1}, & \text{otherwise.} \end{cases} \quad (2.20)$$

The total difference between γ^{b*} and $\hat{\gamma}^{n+1}$ is then redistributed among all indices using a weight function:

$$W_b(\gamma) = 4\gamma - 4\gamma^2 \quad (2.21)$$

$$\gamma_{i,j}^{n+1} = \gamma_{i,j}^{b*} + \frac{W_b(\gamma_{i,j}^{b*})}{\sum_{r,s} W_b(\gamma_{r,s}^{b*})} \sum_{r,s} (\hat{\gamma}_{r,s}^{n+1} - \gamma_{r,s}^{b*}). \quad (2.22)$$

The indices r, s are over the entire computational domain, and a uniform grid is assumed. Since this update is globally conservative, we can express the change in the final quantity γ^{n+1} as the divergence of an unknown weighted gradient term $W_b(\gamma^{n+1}) \nabla Q$ with no-flux boundary conditions. This results in a Poisson equation for Q :

$$\frac{\gamma^{n+1} - \hat{\gamma}^{n+1}}{\widehat{\Delta t}} = \nabla \cdot (W_b(\gamma^{n+1}) \nabla Q), \quad (2.23)$$

$$\mathbf{n} \cdot \nabla Q = 0 \quad (2.24)$$

which we solve and use to compute the density flux $\mathbf{m}$ (See Section 2.3.2).

Cahn-Hilliard boundary conditions

In this paper, we consider two types of boundary conditions for γ : periodic and Neumann ($\mathbf{n} \cdot \nabla \hat{\gamma}^{n+1} = 0$). Since the Cahn-Hilliard equation is fourth order, we require another boundary condition; Noting (2.13)

$$\frac{\hat{\gamma}^{n+1} - \gamma^*}{\widehat{\Delta t}} = M\lambda \nabla^2 \left(F'(\gamma^{AB}) - \nabla^2 \hat{\gamma}^{n+1} + \frac{2s}{M\lambda} (\hat{\gamma}^{n+1} - \gamma^{AB}) \right) + S_\gamma^{n+1} \quad (2.25)$$

we define ξ^* to be a discrete version of the chemical potential from (2.2) as

$$\xi^* = \lambda \left(F'(\gamma^{AB}) - \nabla^2 \hat{\gamma}^{n+1} + \frac{2s}{M\lambda} (\hat{\gamma}^{n+1} - \gamma^{AB}) \right) \quad (2.26)$$

$$= \lambda (\tau A - \nabla^2 \hat{\gamma}^{n+1} + 2\tau \hat{\gamma}^{n+1}) \quad (2.27)$$

$$= \tau \lambda \left(\hat{\gamma}^{n+1} - \frac{1}{\tau} \nabla^2 \hat{\gamma}^{n+1} + A + \hat{\gamma}^{n+1} \right) \quad (2.28)$$

$$= \frac{s}{M} (\gamma' + \hat{\gamma}^{n+1}). \quad (2.29)$$

Following [57] we use the boundary condition $\mathbf{n} \cdot \nabla \xi^* = 0$, which leads to $\mathbf{n} \cdot \nabla \gamma' = 0$. This gives us the necessary boundary conditions for both of our Helmholtz systems.

2.3.2 Navier-Stokes equations

Navier-Stokes temporal discretization

As in Section 2.3.1, we discretize the time derivative as $\frac{\alpha \rho^{n+1} \mathbf{u}^{n+1} - \alpha (\rho \mathbf{u})^{BD}}{\Delta t}$. For the first time step, $\alpha = 1$ and $(\rho \mathbf{u})^{BD} = \rho^n \mathbf{u}^n$. To achieve second-order accuracy, we thereafter use a backwards-difference scheme with $\alpha = \frac{3}{2}$ and $(\rho \mathbf{u})^{BD} = \frac{4}{3} \rho^n \mathbf{u}^n - \frac{1}{3} \rho^{n-1} \mathbf{u}^{n-1}$. Explicit

quantities for $\mathbf{u}$ are discretized using Adams-Bashforth, where $\mathbf{u}^{AB} = \mathbf{u}^n$ in the first time step and $\mathbf{u}^{AB} = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ afterwards. As before, we use $\Delta t = \alpha \widehat{\Delta t}$ to simplify the equations.

In addition to these terms, $\mathbf{u}^v$ is an intermediate estimate of $\mathbf{u}^{n+1}$ using an approximation for pressure, which is computed in an intermediate implicit solve, and $\mathbf{m}$ is the total density flux detailed in Section 2.3.2. With these, (2.9) is discretized as

$$\begin{aligned} \frac{\rho^{n+1}\mathbf{u}^{n+1} - (\rho\mathbf{u})^{BD}}{\widehat{\Delta t}} + \nabla \cdot (\mathbf{m} \otimes \mathbf{u}^{AB}) &= -\nabla p^{n+1} + \rho^{n+1}g + \sigma\kappa\nabla(g(\gamma^{n+1})) \\ &+ \nabla \cdot \left(\mu^{n+1} \left(\nabla \mathbf{u}^v + (\nabla \mathbf{u}^{AB})^T \right) \right) + S_{\mathbf{u}}^{n+1}. \end{aligned} \quad (2.30)$$

The intermediate $\mathbf{u}^*$ is introduced to represent the result of the momentum advection and also the evolution of density through the Cahn-Hilliard equation. When the Cahn-Hilliard equation causes mass to move around, the momentum carried by that mass must move with it. Otherwise mass and momentum may become inconsistent. Following [57] this is accomplished through the mass flux term $\mathbf{m}$, which we derive in Section 2.3.2.

$$\frac{\rho^{n+1}\mathbf{u}^* - (\rho\mathbf{u})^{BD}}{\widehat{\Delta t}} + \nabla \cdot (\mathbf{m} \otimes \mathbf{u}^{AB}) = 0 \quad (2.31)$$

$$\begin{aligned} \frac{\rho^{n+1}\mathbf{u}^{n+1} - \rho^{n+1}\mathbf{u}^*}{\widehat{\Delta t}} &= -\nabla p^{n+1} + \rho^{n+1}g + \sigma\kappa\nabla(g(\gamma^{n+1})) \\ &+ \nabla \cdot \left(\mu^{n+1} \left(\nabla \mathbf{u}^v + (\nabla \mathbf{u}^{AB})^T \right) \right) + S_{\mathbf{u}}^{n+1}. \end{aligned} \quad (2.32)$$

After updating momentum according to equation (2.31), we divide the result through by ρ^{n+1} and formulate equation (2.32) as updates to velocity, beginning with explicit forces

and pressure. We split pressure here into predicted and correction parts as $p^{n+1} = p^n + p'$.

$$\begin{aligned} \frac{\mathbf{u}^{**} - \mathbf{u}^*}{\widehat{\Delta t}} &= -\frac{1}{\rho^{n+1}} \nabla p^n + g + \frac{1}{\rho^{n+1}} \sigma \kappa \nabla (g(\gamma^{n+1})) + \frac{1}{\rho^{n+1}} \nabla \cdot (\mu^{n+1} (\nabla \mathbf{u}^{AB})^T) \\ &\quad + \frac{1}{\rho^{n+1}} S_{\mathbf{u}}^{n+1} \end{aligned} \quad (2.33)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{**}}{\widehat{\Delta t}} = -\frac{1}{\rho^{n+1}} \nabla p' + \frac{1}{\rho^{n+1}} \nabla \cdot (\mu^{n+1} \nabla \mathbf{u}^v) \quad (2.34)$$

Finally, we compute the effects of viscosity implicitly and then project the velocity to be divergence free:

$$\frac{\mathbf{u}^v - \mathbf{u}^{**}}{\widehat{\Delta t}} = \frac{1}{\rho^{n+1}} \nabla \cdot (\mu^{n+1} \nabla \mathbf{u}^v) \quad (2.35)$$

$$\nabla \cdot \left(\frac{1}{\rho^{n+1}} \nabla p' \right) = \frac{\nabla \cdot \mathbf{u}^v}{\widehat{\Delta t}} \quad (2.36)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^v}{\widehat{\Delta t}} = -\frac{1}{\rho^{n+1}} \nabla p'. \quad (2.37)$$

Momentum flux

Since we are solving the Cahn-Hilliard equation, we do not directly evolve density. Instead, $\gamma^n \rightarrow \gamma^{n+1}$ is evolved according to the Cahn-Hilliard equation, and then (2.5) is used to compute ρ^{n+1} . The discrete γ evolution in Section 2.3.1 implies a density evolution of the form

$$\frac{\rho^{n+1} - \rho^{BD}}{\widehat{\Delta t}} + \nabla \cdot \mathbf{m} = S_m^{n+1}, \quad (2.38)$$

where $\mathbf{m}$ includes advection and Cahn-Hilliard separation, and S_m takes the effects of the forcing term S_γ . Using (2.5), we express $\mathbf{m}$ in terms of γ :

$$\nabla \cdot \mathbf{m} = -\frac{\rho^{n+1} - \rho^{BD}}{\widehat{\Delta t}} + S_m^{n+1} \quad (2.39)$$

$$= -(\rho_1 - \rho_0) \frac{\gamma^{n+1} - \gamma^{BD}}{\widehat{\Delta t}} + S_m^{n+1} \quad (2.40)$$

$$= -(\rho_1 - \rho_0) \left(\frac{\gamma^{n+1} - \hat{\gamma}^{n+1}}{\widehat{\Delta t}} + \frac{\hat{\gamma}^{n+1} - \gamma'}{\widehat{\Delta t}} + \frac{\gamma' - \gamma^*}{\widehat{\Delta t}} + \frac{\gamma^* - \gamma^{BD}}{\widehat{\Delta t}} \right) + S_m^{n+1}. \quad (2.41)$$

Using the update rules (2.12), (2.19), (2.18), and (2.23), defining $S_m^{n+1} = (\rho_1 - \rho_0) S_\gamma^{n+1}$, and canceling terms, this becomes

$$\nabla \cdot \mathbf{m} = (\rho_1 - \rho_0) \left(\nabla \cdot (\mathbf{u}^{AB} \gamma^{AB}) - \nabla \cdot (s \nabla \gamma' + s \nabla \hat{\gamma}^{n+1} + W_b (\gamma^{n+1}) \nabla Q) \right) \quad (2.42)$$

Note that this definition of $\mathbf{m}$ is only well-defined up to a divergence-free shift.

To determine what shift is necessary, we consider the case of $\rho_0 = \rho_1$, in which case Cahn-Hilliard has no affect on the density evolution and the mass flux is purely comprised of advection. This suggests the shift $\nabla \cdot (\rho_0 \mathbf{u}^{AB})$ so that

$$\nabla \cdot \mathbf{m} = \nabla \cdot (\rho_0 \mathbf{u}^{AB}) + (\rho_1 - \rho_0) \left(\nabla \cdot (\mathbf{u}^{AB} \gamma^{AB}) - \nabla \cdot (s \nabla \gamma' + s \nabla \hat{\gamma}^{n+1} + W_b (\gamma^{n+1}) \nabla Q) \right), \quad (2.43)$$

which is consistent with the shift in [57]. Observe that ρ_0 is a constant and $\mathbf{u}^{AB}$ is discretely divergence free, so adding this term does not affect the value of $\nabla \cdot \mathbf{m}$. This leads to a

definition of

$$\mathbf{m} = \rho_0 \mathbf{u}^{AB} + (\rho_1 - \rho_0) \left(\mathbf{u}^{AB} \gamma^{AB} - s \nabla \gamma' - s \nabla \hat{\gamma}^{n+1} - W_b(\gamma^{n+1}) \nabla Q \right). \quad (2.44)$$

To ensure consistency with the discretization of the Cahn-Hilliard equation, $\mathbf{u}^{AB} \gamma^{AB}$ is replaced by the flux computed by the WENO advection described in Section 2.3.3.

Surface tension discretization

For surface tension, we use a Continuum Surface Force [14, 33] formulation:

$$\sigma \kappa \nabla(g(\gamma)), \quad (2.45)$$

where κ is the estimated interface curvature:

$$\kappa = -\nabla \cdot \left(\frac{\nabla \gamma}{\|\nabla \gamma\|} \right). \quad (2.46)$$

We formulate $g(\gamma)$ to ensure that the surface tension force has finite support about the interface value $\gamma = 0.5$. In particular, for some value $a \in [0, 0.5]$, the surface tension force is exactly 0 for $\gamma \notin [a, 1 - a]$. We choose a piecewise cubic approximation of a Heaviside function for $g(\gamma)$ whose transition occurs between a and $1 - a$. Since $g(\gamma)$ approximates a Heaviside function, $\nabla g(\gamma)$ approximates a delta function times the normal. This is similar to the method used in [75], but it differs in that they use a level set function to track

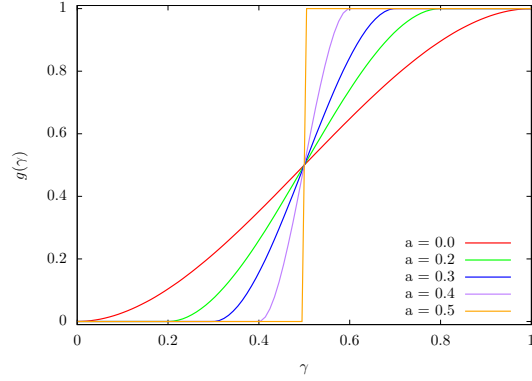


Figure 2.1: $g(\gamma)$ vs γ for different values of a

the interface rather than the phase-field approach used here and in the choice of Heaviside approximation.

As depicted in Figure 2.1, we define $g(\gamma)$ as

$$g(\gamma) = \begin{cases} 1, & \text{if } \gamma \geq 1 - a \\ 0, & \text{if } \gamma < a \\ \frac{(-\gamma+a)^2(2\gamma+4a-3)}{(2a-1)^3}, & \text{otherwise.} \end{cases} \quad (2.47)$$

In practice, we've found $a = 0.2$ works well, and at $a = 0.5$ we recover a sharp interface force, as used in [67].

We can see that this surface tension formulation is consistent with the standard one by considering a path that crosses the band near the interface where the force is nonzero. Given any path C (see Figure 2.2), with discretization $i = 0, \dots, N$, that crosses the fluid

interface, the total force per unit area is:

$$\int_C \sigma \kappa \nabla g(\gamma) \cdot dl \approx \sigma \left(\sum_{i=1}^N \kappa_i (g(\gamma_i) - g(\gamma_{i-1})) \right). \quad (2.48)$$

Assume that $g(\gamma_N) = 1$ and $g(\gamma_0) = 0$, so that the curve crosses completely from one phase to the other. Let κ_I denote the curvature at the interface. If we assume the path is localized and approximate the curvature as constant, so that $\kappa_i = \kappa_I$, (2.48) reduces to:

$$\sigma \left(\sum_{i=1}^N \kappa_i (g(\gamma_i) - g(\gamma_{i-1})) \right) = \sigma \kappa_I (g(\gamma_N) - g(\gamma_0)) = \sigma \kappa_I, \quad (2.49)$$

which is the ideal pressure jump caused by surface tension. In reality, the curvature will not be constant, but when the interface width is small and the curvature is sufficiently smooth, we can write $\kappa_i = \kappa_I + O(\Delta x)$, so that the approximate pressure jump is

$$\sigma \left(\sum_{i=1}^N \kappa_i (g(\gamma_i) - g(\gamma_{i-1})) \right) = \sigma \kappa_I + O(\Delta x). \quad (2.50)$$

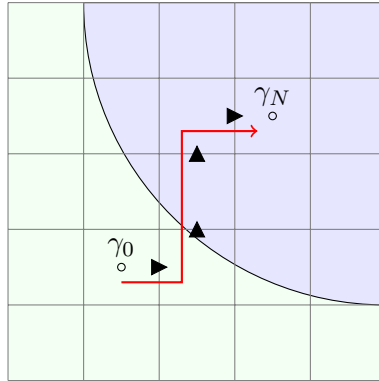


Figure 2.2: Example path $\gamma_0 \rightarrow \gamma_N$. The triangles represent the pressure jumps along this path.

This suggests that smearing the pressure jump over a transition region using $g(\gamma)$ can give a consistent discretization of the surface tension force. We next show this explicitly for our discretization of the pressure Poisson equation (2.36).

For the pressure projection equation, we solve $\nabla \cdot (\beta \nabla p') = \nabla \cdot \mathbf{u}^v$ with $\beta = \frac{\widehat{\Delta t}}{\rho^{n+1}}$.

The standard five-point discretization of this equation in two dimensions is given by

$$\begin{aligned} & \frac{\beta_{i+\frac{1}{2},j}(p'_{i+1,j} - p'_{i,j}) - \beta_{i-\frac{1}{2},j}(p'_{i,j} - p'_{i-1,j})}{\Delta x^2} \\ & + \frac{\beta_{i,j+\frac{1}{2}}(p'_{i,j+1} - p'_{i,j}) - \beta_{i,j-\frac{1}{2}}(p'_{i,j} - p'_{i,j-1})}{\Delta y^2} = (\nabla \cdot \mathbf{u}^v)_{i,j} \end{aligned} \quad (2.51)$$

Following [67], an interface can be treated in a sharp manner by incorporating the pressure jump J across the interface into the equation. Incorporating a jump term (which may be 0) across every difference in the stencil, the discretization becomes

$$\begin{aligned} & \frac{\beta_{i+\frac{1}{2},j}(p'_{i+1,j} - (p'_{i,j} + J_{i+\frac{1}{2},j})) - \beta_{i-\frac{1}{2},j}(p'_{i,j} - (p'_{i-1,j} + J_{i-\frac{1}{2},j}))}{\Delta x^2} \\ & + \frac{\beta_{i,j+\frac{1}{2}}(p'_{i,j+1} - (p'_{i,j} + J_{i,j+\frac{1}{2}})) - \beta_{i,j-\frac{1}{2}}(p'_{i,j} - (p'_{i,j-1} + J_{i,j-\frac{1}{2}}))}{\Delta y^2} = (\nabla \cdot \mathbf{u}^v)_{i,j}. \end{aligned} \quad (2.52)$$

Moving all terms with jumps to the right hand side, we get

$$\begin{aligned} & \frac{\beta_{i+\frac{1}{2},j}(p'_{i+1,j} - p'_{i,j}) - \beta_{i-\frac{1}{2},j}(p'_{i,j} - p'_{i-1,j})}{\Delta x^2} + \frac{\beta_{i,j+\frac{1}{2}}(p'_{i,j+1} - p'_{i,j}) - \beta_{i,j-\frac{1}{2}}(p'_{i,j} - p'_{i,j-1})}{\Delta y^2} \\ & = (\nabla \cdot \mathbf{u}^v)_{i,j} + \frac{\beta_{i+\frac{1}{2},j}J_{i+\frac{1}{2},j} - \beta_{i-\frac{1}{2},j}J_{i-\frac{1}{2},j}}{\Delta x^2} + \frac{\beta_{i,j+\frac{1}{2}}J_{i,j+\frac{1}{2}} - \beta_{i,j-\frac{1}{2}}J_{i,j-\frac{1}{2}}}{\Delta y^2}. \end{aligned} \quad (2.53)$$

With the central difference discretization of divergence

$$(\nabla \cdot \mathbf{u}^v)_{i,j} = \frac{u_{i+\frac{1}{2},j}^v - u_{i-\frac{1}{2},j}^v}{\Delta x} + \frac{v_{i,j+\frac{1}{2}}^v - v_{i,j-\frac{1}{2}}^v}{\Delta y}, \quad (2.54)$$

and with $\hat{\mathbf{J}} = \left(\frac{J_x}{\Delta x}, \frac{J_y}{\Delta y} \right)$, Equation (2.53) becomes

$$\begin{aligned} & \frac{\beta_{i+\frac{1}{2},j} \left(p'_{i+1,j} - p'_{i,j} \right) - \beta_{i-\frac{1}{2},j} \left(p'_{i,j} - p'_{i-1,j} \right)}{\Delta x^2} \\ & + \frac{\beta_{i,j+\frac{1}{2}} \left(p'_{i,j+1} - p'_{i,j} \right) - \beta_{i,j-\frac{1}{2}} \left(p'_{i,j} - p'_{i,j-1} \right)}{\Delta y^2} = \left(\nabla \cdot \left(\mathbf{u}^v + \beta \hat{\mathbf{J}} \right) \right)_{i,j}. \end{aligned} \quad (2.55)$$

We define the pressure jump at $(i + \frac{1}{2}, j)$ to be

$$J_{i+\frac{1}{2},j} = \sigma \kappa_{i+\frac{1}{2},j} (g(\gamma_{i+1,j}) - g(\gamma_{i,j})). \quad (2.56)$$

The discretization of the x component of the surface tension force (2.45) is then

$$\sigma \kappa_{i+\frac{1}{2},j} \left(\frac{\partial g}{\partial x}(\gamma) \right)_{i+\frac{1}{2},j} \approx \sigma \kappa_{i+\frac{1}{2},j} \frac{g(\gamma_{i+1,j}) - g(\gamma_{i,j})}{\Delta x} = \frac{1}{\Delta x} J_{i+\frac{1}{2},j}, \quad (2.57)$$

which is similar to [33]. Therefore

$$\beta \hat{\mathbf{J}} = \frac{\widehat{\Delta t}}{\rho^{n+1}} \sigma \kappa \nabla g(\gamma), \quad (2.58)$$

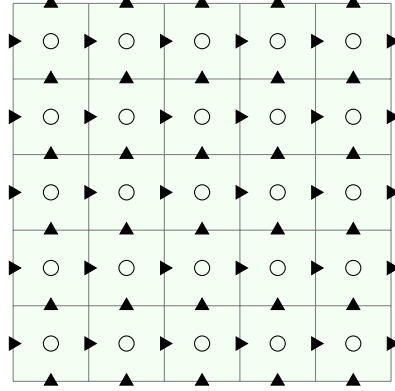


Figure 2.3: The MAC grid stores velocities on faces (represented by triangles), and γ and pressure on cell centers (circles).

which shows that solving the Poisson equation with the above defined pressure jumps is equivalent to applying surface tension with other explicit forces:

$$\nabla \cdot \left(\frac{\widehat{\Delta t}}{\rho^{n+1}} \nabla p' \right) = \nabla \cdot \left(\mathbf{u}^v + \frac{\widehat{\Delta t}}{\rho^{n+1}} \sigma \kappa \nabla g(\gamma) \right). \quad (2.59)$$

Therefore we apply surface tension as in (2.59), and we can recover the sharp formulation of [67] by choosing $a = 0.5$ in the definition of $g(\gamma)$.

2.3.3 Spatial Discretization

We use a uniform MAC grid (Figure 2.3) with γ and pressure stored at cell centers. Velocities are split by components and stored on cell faces. Unless otherwise specified, all divergences are central differences from faces to cells and all gradients are central differences from cells to faces. All Poisson equations are discretized using the standard five-point central difference stencil, with standard treatment of axis-aligned boundary conditions.

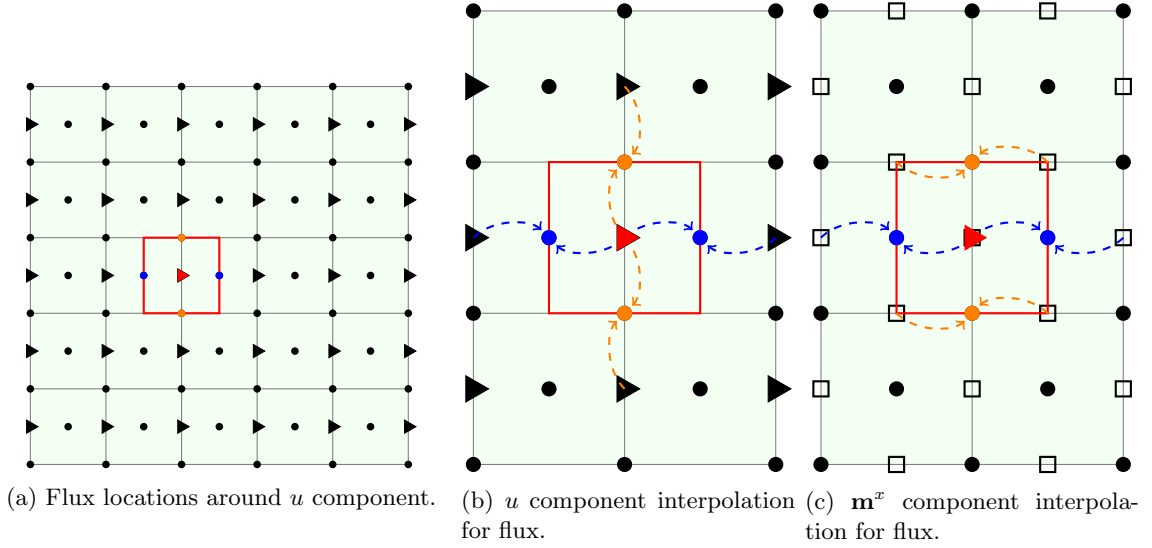


Figure 2.4: The components of the momentum flux for updating the x component of momentum in 2D are shown in 2.4a. The momentum at the red triangle will be updated with the fluxes at the surrounding blue and orange dots. The u and $\mathbf{m}^x$ values at these dots are interpolated from neighboring values as illustrated in 2.4b and 2.4c, respectively.

Advective flux for γ

When computing the divergence operator for γ advection in (2.12), we use a flux-based 5th order WENO advection scheme [66] with local Lax-Friedrichs [132]. The WENO flux $\mathcal{F}^W$ is stored for later use in the momentum flux $\mathbf{m}$.

Divergence operator (momentum advection)

The divergence operator for momentum advection is discretized as a central difference from dual-cell faces to MAC grid faces, as shown in Figure 2.4. The components of the divergence $\nabla \cdot (\mathbf{m} \otimes \mathbf{u})$ are needed at MAC faces to update the components of our velocities. Correspondingly we discretize $\nabla \cdot (\mathbf{m}^x u)$ and $\nabla \cdot (\mathbf{m}^y v)$ at MAC x -faces and y -faces respectively. For example, $\mathbf{m}^x$ is interpolated to faces of a uniform dual grid centered

on the x-faces:

$$\mathbf{m}_{i,j}^x = \frac{\mathbf{m}_{i+1/2,j} + \mathbf{m}_{i-1/2,j}}{2} \quad (2.60)$$

$$\mathbf{m}_{i+1/2,j+1/2}^x = \frac{\mathbf{m}_{i+1,j+1/2} + \mathbf{m}_{i,j+1/2}}{2}. \quad (2.61)$$

We also interpolate u to the same dual grid faces

$$u_{i,j} = \frac{u_{i+1/2,j} + u_{i-1/2,j}}{2} \quad (2.62)$$

$$u_{i+1/2,j+1/2} = \frac{u_{i+1/2,j+1} + u_{i+1/2,j}}{2}. \quad (2.63)$$

The final component divergence is discretized as

$$\begin{aligned} (\nabla \cdot (\mathbf{m}^x u))_{i+1/2,j} &= \frac{\mathbf{m}_{i+1,j}^x u_{i+1,j} - \mathbf{m}_{i,j}^x u_{i,j}}{\Delta x} \\ &+ \frac{\mathbf{m}_{i+1/2,j+1/2}^x u_{i+1/2,j+1/2} - \mathbf{m}_{i+1/2,j-1/2}^x u_{i+1/2,j-1/2}}{\Delta y}. \end{aligned} \quad (2.64)$$

The $\mathbf{m}^y$ term is discretized analogously.

Divergence gradient transpose

In (2.33), we discretize the $\nabla \cdot (\mu (\nabla \mathbf{u})^T)$ term following [57], which we show here:

$$\begin{aligned}
\left(\nabla \cdot (\mu (\nabla \mathbf{u})^T) \right)_{i,j} &= \frac{1}{\Delta x} \left(\mu_{i+1/2,j} \frac{u_{i+1,j} - u_{i,j}}{\Delta x} - \mu_{i-1/2,j} \frac{u_{i,j} - u_{i-1,j}}{\Delta x} \right) \\
&+ \frac{1}{\Delta y} \left(\mu_{i,j+1/2} \frac{v_{i+1/2,j+1/2} - v_{i-1/2,j+1/2}}{\Delta x} - \mu_{i,j-1/2} \frac{v_{i+1/2,j-1/2} - v_{i-1/2,j-1/2}}{\Delta x} \right) \\
&+ \frac{1}{\Delta y} \left(\mu_{i,j+1/2} \frac{v_{i,j+1} - v_{i,j}}{\Delta y} - \mu_{i,j-1/2} \frac{v_{i,j} - v_{i,j-1}}{\Delta y} \right) \\
&+ \frac{1}{\Delta x} \left(\mu_{i+1/2,j} \frac{u_{i+1/2,j+1/2} - u_{i+1/2,j-1/2}}{\Delta y} - \mu_{i-1/2,j} \frac{u_{i-1/2,j+1/2} - u_{i-1/2,j-1/2}}{\Delta y} \right).
\end{aligned} \tag{2.65}$$

This discretization is used to ensure discrete consistency when μ is constant, such that

$$\nabla \cdot (\mu (\nabla \mathbf{u})^T) = \mu \nabla (\nabla \cdot \mathbf{u}) = 0 \tag{2.66}$$

is satisfied and momentum is conserved [57].

Interface curvature

The Balanced-Force algorithm [33] requires that the surface tension force is discretized at the same spatial location as the pressure gradient. When calculating the estimated interface curvature κ ,

$$\kappa = -\nabla \cdot \left(\frac{\nabla \gamma}{\|\nabla \gamma\|} \right), \tag{2.67}$$

we smooth the curvature estimate by computing the divergence on cells and interpolating to faces. We calculate $\nabla\gamma$ as individual components $\nabla\gamma^x, \nabla\gamma^y$ on faces,

$$\nabla\gamma_{i+1/2,j}^x = \frac{\gamma_{i+1,j} - \gamma_{i,j}}{\Delta x}, \quad (2.68)$$

which we interpolate to cell centers as vectors in order to normalize:

$$\widetilde{\nabla}\gamma_{i,j} = \left(\frac{\nabla\gamma_{i+1/2,j}^x + \nabla\gamma_{i-1/2,j}^x}{2}, \frac{\nabla\gamma_{i,j+1/2}^y + \nabla\gamma_{i,j-1/2}^y}{2} \right). \quad (2.69)$$

The normalized gradients are interpolated back to faces as full vectors:

$$\widehat{\nabla}\gamma_{i+1/2,j} = \frac{1}{2} \left(\frac{\widetilde{\nabla}\gamma_{i+1,j}}{\|\widetilde{\nabla}\gamma_{i+1,j}\|} + \frac{\widetilde{\nabla}\gamma_{i,j}}{\|\widetilde{\nabla}\gamma_{i,j}\|} \right). \quad (2.70)$$

From these gradient vectors on faces, we calculate the divergence at cell centers

$$\nabla \cdot (\widehat{\nabla}\gamma)_{i,j} = \frac{1}{\Delta x} (\widehat{\nabla}\gamma_{i+1/2,j}^x - \widehat{\nabla}\gamma_{i-1/2,j}^x) + \frac{1}{\Delta y} (\widehat{\nabla}\gamma_{i,j+1/2}^y - \widehat{\nabla}\gamma_{i,j-1/2}^y), \quad (2.71)$$

and interpolate to faces:

$$\kappa_{i+1/2,j} = -\frac{\nabla \cdot (\widehat{\nabla}\gamma)_{i+1,j} + \nabla \cdot (\widehat{\nabla}\gamma)_{i,j}}{2}. \quad (2.72)$$

2.3.4 Discretization summary

We finish our treatment of discretization with a summary of the full algorithm.

First, the Adams-Bashforth and backwards-difference quantities γ^{BD} and γ^{AB} on cell cen-

ters and $(\rho \mathbf{u})^{BD}$ and $\mathbf{u}^{AB}$ on cell faces are computed from the known initial quantities and previous time step quantities. The initial time step is computed with first order time discretizations:

$$\gamma^{BD} = \gamma^n \quad \gamma^{AB} = \gamma^n \quad (\rho \mathbf{u})^{BD} = \rho^n \mathbf{u}^n \quad \mathbf{u}^{AB} = \mathbf{u}^n \quad \widehat{\Delta t} = \Delta t, \quad (2.73)$$

and successive time steps with second order:

$$\begin{aligned} \gamma^{BD} &= \frac{4}{3}\gamma^n - \frac{1}{3}\gamma^{n-1} & \gamma^{AB} &= 2\gamma^n - \gamma^{n-1}; \\ (\rho \mathbf{u})^{BD} &= \frac{4}{3}\rho^n \mathbf{u}^n - \frac{1}{3}\rho^{n-1} \mathbf{u}^{n-1} & \mathbf{u}^{AB} &= 2\mathbf{u}^n - \mathbf{u}^{n-1} & \widehat{\Delta t} &= \frac{2}{3}\Delta t. \end{aligned} \quad (2.74)$$

Cahn-Hilliard discretization

Unlike [57], we observed better results using a WENO advection scheme (Section 2.3.3) to compute γ^* from γ^{BD} . The flux $\mathcal{F}^W$ from this operation is stored for later use in $\mathbf{m}$. These terms are related by:

$$\gamma_{i,j}^* = \gamma_{i,j}^{BD} - \widehat{\Delta t} \left(\frac{\mathcal{F}_{i+1/2,j}^W - \mathcal{F}_{i-1/2,j}^W}{\Delta x} + \frac{\mathcal{F}_{i,j+1/2}^W - \mathcal{F}_{i,j-1/2}^W}{\Delta y} \right). \quad (2.75)$$

Next we compute $\hat{\gamma}^{n+1}$ from γ^* by solving two Helmholtz equations:

$$\lambda = \frac{3}{2\sqrt{2}}\sigma\eta \quad s = \sqrt{\frac{M\lambda}{\widehat{\Delta t}}} \quad A = \frac{M\lambda}{s}F'(\gamma^{AB}) - 2\gamma^{AB} \quad (2.76)$$

$$\gamma' - s\widehat{\Delta t}\nabla^2\gamma' = \gamma^* + A + \widehat{\Delta t}S_\gamma^{n+1} \quad (2.77)$$

$$\hat{\gamma}^{n+1} - s\widehat{\Delta t}\nabla^2\hat{\gamma}^{n+1} = \gamma' - A. \quad (2.78)$$

We obtain the final γ^{n+1} through the redistribution scheme described in Section 2.3.1. The flux corresponding to the redistribution is found with a Poisson solve:

$$\gamma_{i,j}^{b*} = \begin{cases} 1, & \text{if } \hat{\gamma}_{i,j}^{n+1} \geq 1 \\ 0, & \text{if } \hat{\gamma}_{i,j}^{n+1} \leq 0 \\ \hat{\gamma}_{i,j}^{n+1}, & \text{otherwise.} \end{cases} \quad (2.79)$$

$$W_b(\gamma) = 4\gamma - 4\gamma^2 \quad (2.80)$$

$$\gamma_{i,j}^{n+1} = \gamma_{i,j}^{b*} + \frac{W_b(\gamma_{i,j}^{b*})}{\sum_{r,s} W_b(\gamma_{r,s}^{b*})} \sum_{r,s} (\hat{\gamma}_{r,s}^{n+1} - \gamma_{r,s}^{b*}) \quad (2.81)$$

$$\widehat{\Delta t} \nabla \cdot (W_b(\gamma^{n+1}) \nabla Q) = \gamma^{n+1} - \hat{\gamma}^{n+1}. \quad (2.82)$$

To avoid rows of all zeros in our Poisson matrix, we follow [56] in replacing $W_b(\gamma^{n+1})$ with $\max(W_b(\gamma^{n+1}), 10^{-6})$ in (2.82). Using the above, we can compute the discrete mass flux, which should satisfy (2.38) to round-off error:

$$\mathbf{m} = \rho_0 \mathbf{u}^{AB} + (\rho_1 - \rho_0) (\mathcal{F}^W - s \nabla \gamma' - s \nabla \hat{\gamma}^{n+1} - W_b(\gamma^{n+1}) \nabla Q). \quad (2.83)$$

The density and viscosity are computed on faces using γ^{n+1} that has been linearly interpolated to faces:

$$\rho^{n+1} = \rho_0 + (\rho_1 - \rho_0) \gamma^{n+1} \quad (2.84)$$

$$\mu^{n+1} = \mu_0 + (\mu_1 - \mu_0) \gamma^{n+1}. \quad (2.85)$$

Navier-Stokes discretization

We first update the momentum to be consistent with the movement of mass that occurred due to the Cahn-Hilliard equation by:

$$\rho^{n+1} \mathbf{u}^* = (\rho \mathbf{u})^{BD} - \widehat{\Delta t} \nabla \cdot (\mathbf{m} \otimes \mathbf{u}^{AB}), \quad (2.86)$$

after which ρ^{n+1} is divided off to obtain $\mathbf{u}^*$. Next we apply explicit forces:

$$\begin{aligned} \mathbf{u}^{**} = \mathbf{u}^* + \widehat{\Delta t} \mathbf{g} + \frac{\widehat{\Delta t}}{\rho^{n+1}} (\sigma \kappa \nabla (g(\gamma^{n+1}))) + \frac{\widehat{\Delta t}}{\rho^{n+1}} \nabla \cdot (\mu^{n+1} (\nabla \mathbf{u}^{AB})^T) \\ - \frac{\widehat{\Delta t}}{\rho^{n+1}} \nabla p^n + \widehat{\Delta t} S_{\mathbf{u}}^{n+1} \end{aligned} \quad (2.87)$$

where

$$\kappa = -\nabla \cdot \left(\frac{\nabla \gamma}{\|\nabla \gamma\|} \right) \quad (2.88)$$

$$g(\gamma) = \begin{cases} 1, & \text{if } \gamma \geq 1 - a \\ 0, & \text{if } \gamma < a \\ \frac{(-\gamma+a)^2(2\gamma+4a-3)}{(2a-1)^3}, & \text{otherwise} \end{cases} \quad (2.89)$$

We use $a = 0.2$ unless otherwise stated. We solve the viscosity equation for each component of $\mathbf{u}^v$ by solving a Helmholtz equation:

$$\mathbf{u}^v - \frac{\widehat{\Delta t}}{\rho^{n+1}} \nabla \cdot (\mu^{n+1} \nabla \mathbf{u}^v) = \mathbf{u}^{**}. \quad (2.90)$$

The pressure correction is obtained by solving the Poisson equation:

$$\nabla \cdot \left(\frac{\widehat{\Delta t}}{\rho^{n+1}} \nabla p' \right) = \nabla \cdot \mathbf{u}^v \quad (2.91)$$

$$p^{n+1} = p^n + p', \quad (2.92)$$

and the pressure correction is applied to obtain the final velocity:

$$\mathbf{u}^{n+1} = \mathbf{u}^v - \frac{\widehat{\Delta t}}{\rho^{n+1}} \nabla p'. \quad (2.93)$$

2.4 Analysis

2.4.1 Von Neumann stability analysis

We can get a sense of the stability of the Cahn-Hilliard discretization by performing von Neumann analysis. To do this, we must first identify nonlinearities. The first is advection, which uses conservative WENO and has a CFL restriction; we must ensure $|u|\Delta t \leq \Delta x$, which we always obey. In the analysis that follows, we ignore advection by assuming zero velocity and also ignore redistribution. The other nonlinearity we must address is the double-well potential, which we replace by a single-well potential, so that $F'(\gamma) = k\gamma$, for $k > 0$.

With the nonlinearities eliminated, we may assume periodic boundary conditions and perform von Neumann analysis. To do this, we assume a solution of the form $\gamma_{r,s}^n = g^n e^{ia\Delta x r + ib\Delta x s}$, where a and b are the wave numbers of the solution and g is the (potentially complex-valued) growth factor. Note that g^n is g to the n -th power. The scheme is stable

if $|g| < 1$ and unstable if $|g| > 1$; the case $|g| = 1$ is more complex but not of importance here.

$$\gamma_{r,s}^n = g^n e^{ia\Delta x r + ib\Delta x s} \quad (2.94)$$

$$\gamma_{r,s}^{n-1} = g^{n-1} e^{ia\Delta x r + ib\Delta x s} \quad (2.95)$$

$$\gamma_{r,s}^{BD} = \frac{4}{3}\gamma_{r,s}^n - \frac{1}{3}\gamma_{r,s}^{n-1} = c_0 g^n e^{ia\Delta x r + ib\Delta x s} \quad c_0 = \frac{4}{3} - \frac{1}{3g} \quad (2.96)$$

$$\gamma_{r,s}^{AB} = 2\gamma_{r,s}^n - \gamma_{r,s}^{n-1} = c_1 g^n e^{ia\Delta x r + ib\Delta x s} \quad c_1 = 2 - \frac{1}{g} \quad (2.97)$$

$$\gamma_{r,s}^* = \gamma_{r,s}^{BD} = c_0 g^n e^{ia\Delta x r + ib\Delta x s} \quad (2.98)$$

$$A_{r,s} = \frac{M\lambda}{s} k \gamma_{r,s}^{AB} - 2\gamma_{r,s}^{AB} = c_2 g^n e^{ia\Delta x r + ib\Delta x s} \quad c_2 = \left(\frac{M\lambda}{s} k - 2 \right) c_1 \quad (2.99)$$

$$B_{r,s} = \gamma_{r,s}^* + A_{r,s} = c_3 g^n e^{ia\Delta x r + ib\Delta x s} \quad c_3 = c_0 + c_2 \quad (2.100)$$

$$\gamma'_{r,s} = c_4 g^n e^{ia\Delta x r + ib\Delta x s} \quad (2.101)$$

$$B_{r,s} = \gamma'_{r,s} - s \widehat{\Delta t} \nabla^2 \gamma'_{r,s} \quad (2.102)$$

$$B_{r,s} = \gamma'_{r,s} - s \widehat{\Delta t} \frac{\gamma'_{r+1,s} + \gamma'_{r-1,s} + \gamma'_{r,s+1} + \gamma'_{r,s-1} - 4\gamma'_{r,s}}{\Delta x^2} \quad (2.103)$$

$$c_3 = c_4 \left(1 - s \frac{\widehat{\Delta t}}{\Delta x^2} \left(e^{ia\Delta x} + e^{-ia\Delta x} + e^{ib\Delta x} + e^{-ib\Delta x} - 4 \right) \right)$$

$$C_{r,s} = \gamma'_{r,s} - A_{r,s} = c_5 g^n e^{ia\Delta x r + ib\Delta x s} \quad c_5 = c_4 - c_2 \quad (2.104)$$

$$\gamma_{r,s}^{n+1} = g^{n+1} e^{ia\Delta x r + ib\Delta x s} \quad (2.105)$$

$$C_{r,s} = \hat{\gamma}_{r,s}^{n+1} - s\widehat{\Delta t} \nabla^2 \hat{\gamma}_{r,s}^{n+1} \quad (2.106)$$

$$C_{r,s} = \gamma_{r,s}^{n+1} - s\widehat{\Delta t} \frac{\gamma_{r+1,s}^{n+1} + \gamma_{r-1,s}^{n+1} + \gamma_{r,s+1}^{n+1} + \gamma_{r,s-1}^{n+1} - 4\gamma_{r,s}^{n+1}}{\Delta x^2} \quad (2.107)$$

$$c_5 = g \left(1 - s \frac{\widehat{\Delta t}}{\Delta x^2} \left(e^{ia\Delta x} + e^{-ia\Delta x} + e^{ib\Delta x} + e^{-ib\Delta x} - 4 \right) \right)$$

Combining the equations relating the constants $c_0, \dots, c_5$, we obtain an equation for g . To simplify the algebra, we let $q = 4 - e^{ia\Delta x} - e^{-ia\Delta x} - e^{ib\Delta x} - e^{-ib\Delta x}$, so that $0 \leq q \leq 8$. We eliminate M and λ using $M\lambda = s^2 \widehat{\Delta t}$. Our equations then become

$$c_0 = \frac{4}{3} - \frac{1}{3g} \quad c_1 = 2 - \frac{1}{g} \quad c_2 = (s\widehat{\Delta t}k - 2)c_1 \quad c_3 = c_0 + c_2 \quad (2.108)$$

$$c_3 = c_4 \left(1 + s \frac{\widehat{\Delta t}}{\Delta x^2} q \right) \quad c_5 = c_4 - c_2 \quad c_5 = g \left(1 + s \frac{\widehat{\Delta t}}{\Delta x^2} q \right) \quad (2.109)$$

We can further simplify these with $m = s \frac{\widehat{\Delta t}}{\Delta x^2} q$ and $p = s\widehat{\Delta t}k$, where $m \geq 0$ and $p > 0$.

Combining these leads to the quadratic equation

$$3g^2(1+m)^2 + (6mp - 12m - 4)g - 3mp + 6m + 1 = 0. \quad (2.110)$$

We must show that this equation never has a root with $|g| > 1$. At $m = 0$, we have $3g^2 - 4g + 1 = 0$, which has solutions $g = 1, \frac{1}{3}$. Solving (2.110) using the quadratic formula and taking a Taylor series in m we see that $g = 1 - \frac{3}{2}mp + O(m^2)$, so that if m is positive but sufficiently small, we have real g with $0 < g < 1$. Next, we must show that if m continues

to increase, we continue to have $|g| < 1$. If we were to violate this as we increase m , then for some m we must have (a) $g = 1$, (b) $g = -1$, or (c) g is a complex conjugate pair with $|g| = 1$. We consider each possibility in turn. For case (a), we let $g = 1$ in (2.110), which yields $3m(m + p) = 0$, which is impossible for $m > 0$ and $p > 0$.

For case (c), note that if $g^2 + rg + s = 0$ has complex conjugate roots, then $|g|^2 = g\bar{g} = s$, so that

$$|g|^2 = \frac{-3mp + 6m + 1}{3(1 + m)^2} \leq \frac{6m + 1}{3(1 + m)^2} = \frac{6m + 1}{(6m + 1) + (2 + 3m^2)} < 1. \quad (2.111)$$

Finally, we conclude that any instability (and thus time step restriction) must occur as a result of case (b). For case (b), we plug $g = -1$ into (2.110), which yields

$$3m^2 - 9mp + 24m + 8 = 0. \quad (2.112)$$

Plugging in m and p , eliminating s (which depends on $\widehat{\Delta t}$), and simplifying yields

$$3q(q - 3k\Delta x^2) \left(\frac{\sqrt{\widehat{\Delta t} M \lambda}}{\Delta x^2} \right)^2 + 24q \left(\frac{\sqrt{\widehat{\Delta t} M \lambda}}{\Delta x^2} \right) + 8 = 0, \quad (2.113)$$

which is a quadratic polynomial in $\sqrt{\widehat{\Delta t}}$. If $q - 3k\Delta x^2 \geq 0$, all of the coefficients of the polynomial will be positive, so that no positive roots can exist. Thus, we must have $q - 3k\Delta x^2 < 0$. Let $q = 3fk\Delta x^2$, where $0 < f < 1$. (Note that the entire range of f might not be feasible, since it may violate $q \leq 8$, but for Δx sufficiently small this will not be the

case.) This leads to

$$27f(f-1) \left(k\sqrt{\widehat{\Delta t} M \lambda}\right)^2 + 72f \left(k\sqrt{\widehat{\Delta t} M \lambda}\right) + 8 = 0 \quad (2.114)$$

If $0 < f < 1$, this has one positive solution and one negative solution. The positive solution is

$$k\sqrt{\widehat{\Delta t} M \lambda} = \frac{2(6f + \sqrt{30f^2 + 6f})}{9f(1-f)} \quad (2.115)$$

Since we want the wavenumber leading to the tightest time step restriction, we choose f to minimize the right, which occurs at

$$f = \frac{\sqrt{6} - 1}{10} \approx 0.1449 \quad (2.116)$$

$$k\sqrt{\widehat{\Delta t} M \lambda} = \frac{24 + 4\sqrt{6}}{9} \approx 3.755 \quad (2.117)$$

This finally leads to the time step restriction

$$\Delta t \leq \frac{16(7 + 2\sqrt{6})}{9k^2 M \lambda} \approx \frac{21.15}{k^2 M \lambda}, \quad (2.118)$$

where we have used $\Delta t = \frac{3}{2}\hat{\Delta t}$. We note that the time step restriction on Δt is independent of Δx . We numerically evaluate the stability of the linearized stability analysis in Section 2.5.3.

2.4.2 Truncation error analysis

In this section, we compute the truncation error for the Cahn-Hilliard discretization. We omit advection from the analysis, since WENO advection is difficult to analyze and is more accurate than the rest of the scheme. We also omit spatial accuracy from the analysis, since all of the spatial stencils are second order accurate. Instead, we focus on the temporal accuracy. We compute the truncation error by plugging the analytical solution $\gamma(\mathbf{x}, t)$ into the discretization. Because of the Helmholtz solves, it is easiest to compute intermediates from the beginning and the end, measuring the error where they meet. We perform our Taylor expansion about the time $t = n + 1$ and in 1D for simplicity.

$$\gamma^{n+1} = \gamma \tag{2.119}$$

$$\gamma^n = \gamma - \Delta t \gamma_t + \frac{1}{2} \Delta t^2 \gamma_{tt} + O(\Delta t^3) \tag{2.120}$$

$$\gamma^{n-1} = \gamma - 2\Delta t \gamma_t + 2\Delta t^2 \gamma_{tt} + O(\Delta t^3) \tag{2.121}$$

$$\gamma^* = \gamma^{BD} = \frac{4}{3} \gamma^n - \frac{1}{3} \gamma^{n-1} = \gamma - \frac{2}{3} \Delta t \gamma_t + O(\Delta t^3) \tag{2.122}$$

$$\gamma^{AB} = 2\gamma^n - \gamma^{n-1} = \gamma - \Delta t^2 \gamma_{tt} + O(\Delta t^3) \tag{2.123}$$

Next, we note that $\widehat{\Delta t} = \frac{2}{3}\Delta t$ and $s = \sqrt{\frac{M\lambda}{\Delta t}} = \sqrt{\frac{3M\lambda}{2}}\Delta t^{-1/2} = \hat{s}\Delta t^{-1/2}$, where we have introduced $\hat{s}$ to simplify the algebra.

$$A = \frac{M\lambda}{s}F'(\gamma^{AB}) - 2\gamma^{AB} \quad (2.124)$$

$$= \Delta t^{1/2}\frac{M\lambda}{\hat{s}}F'(\gamma) - \Delta t^{5/2}\frac{M\lambda}{\hat{s}}\gamma_{tt}F''(\gamma) - 2\gamma + 2\Delta t^2\gamma_{tt} + O(\Delta t^3) \quad (2.125)$$

$$= -2\gamma + \Delta t^{1/2}\frac{M\lambda}{\hat{s}}F'(\gamma) + 2\Delta t^2\gamma_{tt} - \Delta t^{5/2}\frac{M\lambda}{\hat{s}}\gamma_{tt}F''(\gamma) + O(\Delta t^3) \quad (2.126)$$

$$\gamma' = \gamma^{n+1} - s\widehat{\Delta t}\nabla^2\gamma^{n+1} + A \quad (2.127)$$

$$= \gamma - \frac{2}{3}\hat{s}\Delta t^{1/2}\gamma_{xx} + \Delta t^{1/2}\frac{M\lambda}{\hat{s}}F'(\gamma) - \Delta t^{5/2}\frac{M\lambda}{\hat{s}}\gamma_{tt}F''(\gamma) - 2\gamma + 2\Delta t^2\gamma_{tt} + O(\Delta t^3) \quad (2.128)$$

$$= -\gamma - \frac{2}{3}\hat{s}\Delta t^{1/2}\gamma_{xx} + \Delta t^{1/2}\frac{M\lambda}{\hat{s}}F'(\gamma) + 2\Delta t^2\gamma_{tt} - \Delta t^{5/2}\frac{M\lambda}{\hat{s}}\gamma_{tt}F''(\gamma) + O(\Delta t^3) \quad (2.129)$$

$$\gamma'_x = -\gamma_x - \frac{2}{3}\hat{s}\Delta t^{1/2}\gamma_{xxx} + \Delta t^{1/2}\frac{M\lambda}{\hat{s}}\gamma_xF''(\gamma) + 2\Delta t^2\gamma_{ttx} + O(\Delta t^{5/2}) \quad (2.130)$$

$$\begin{aligned} \gamma'_{xx} = & -\gamma_{xx} - \frac{2}{3}\hat{s}\Delta t^{1/2}\gamma_{xxxx} + \Delta t^{1/2}\frac{M\lambda}{\hat{s}}\gamma_{xx}F''(\gamma) + \Delta t^{1/2}\frac{M\lambda}{\hat{s}}\gamma_x\gamma_xF'''(\gamma) \\ & + 2\Delta t^2\gamma_{ttxx} + O(\Delta t^{5/2}) \end{aligned} \quad (2.131)$$

Finally, we compute the truncation error as

$$\delta = \gamma' - s\widehat{\Delta t}\nabla^2\gamma' - \gamma^* - A \quad (2.132)$$

$$= \gamma - \frac{2}{3}\hat{s}\Delta t^{1/2}\gamma_{xx} - s\widehat{\Delta t}\gamma'_{xx} - \gamma^* \quad (2.133)$$

$$= -\frac{2}{3}\hat{s}\Delta t^{1/2}\gamma_{xx} - \frac{2}{3}\hat{s}\Delta t^{1/2}\gamma'_{xx} + \frac{2}{3}\Delta t\gamma_t + O(\Delta t^3) \quad (2.134)$$

$$= \frac{4}{9}\hat{s}^2\Delta t\gamma_{xxxx} - \frac{2}{3}\Delta tM\lambda\gamma_{xx}F'''(\gamma) - \frac{2}{3}\Delta tM\lambda\gamma_x\gamma_xF'''(\gamma) + \frac{2}{3}\Delta t\gamma_t \quad (2.135)$$

$$- \frac{4}{3}\hat{s}\Delta t^{5/2}\gamma_{ttxx} + O(\Delta t^3)$$

$$= \frac{2}{3}\Delta t(\gamma_t - M\lambda(F'(\gamma) - \gamma_{xx})_{xx}) - \frac{4}{3}\hat{s}\Delta t^{5/2}\gamma_{ttxx} + O(\Delta t^3) \quad (2.136)$$

$$= -\frac{4}{3}\hat{s}\Delta t^{5/2}\gamma_{ttxx} + O(\Delta t^3) \quad (2.137)$$

Thus we see an $O(\Delta t^{5/2})$ local truncation error, which corresponds to a global truncation error of $O(\Delta t^{3/2})$, so that the discretization is formally order 1.5 accurate.

2.5 Numerical examples

All the units in the following tests are SI units.

2.5.1 Manufactured solution

In this test, we evaluate the accuracy of our discrete solvers using a manufactured solution for γ and $\mathbf{u}$. We choose the source terms to enforce the manufactured solution as

$$S_\gamma = \frac{\partial\gamma}{\partial t} + \nabla \cdot (\mathbf{u}\gamma) - \nabla \cdot (M\nabla\xi) \quad (2.138)$$

$$S_{\mathbf{u}} = \frac{\partial\rho\mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{m} \otimes \mathbf{u}) + \nabla p - \rho g - \sigma\kappa\nabla(g(\gamma)) - \nabla \cdot \left(\mu \left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T \right) \right). \quad (2.139)$$

We compute the error and its order at the end of the simulation to measure the accuracy of our method. We first assess the accuracy of the Cahn-Hilliard solver alone. We then incorporate the Navier-Stokes solve and evaluate the accuracy of the full method. Our parameters are chosen following [57]. In this section, we use a square domain $[-\pi, \pi]^2$. We set the density of the fluid phases to be $\rho_0 = 1$ and $\rho_1 = 3$, with viscosities $\mu_0 = 0.01$ and $\mu_1 = 0.02$. We use a time step of $\Delta t = 0.08/N$ and cell size $\Delta x = 2\pi/N$. We select the Cahn-Hilliard parameters M , σ , and η such that $M = 0.001$, $\lambda = 0.001$, and $\eta = 0.1$.

Cahn-Hilliard solver

We assess the convergence of our Cahn-Hilliard discretization using the manufactured solution

$$\gamma(x, y, t) = \frac{1}{2} (\cos(x) \cos(y) + 1) \sin(t). \quad (2.140)$$

To focus solely on the accuracy of the Cahn-Hilliard discretization, we exclude the Navier-Stokes update and set $u = v = 0$. We set the initial and boundary conditions using the manufactured solution. We measure the error in our simulation at $t = 1$. We determine the L_2 and L_∞ convergence by refining N from 8 to 256 with a constant $\frac{\Delta t}{\Delta x}$. The results are listed in Table 2.1 and show that our Cahn-Hilliard discretization is second order accurate.

Cahn-Hilliard-Navier-Stokes solver

We evaluate the convergence of our full method employing both the Cahn-Hilliard and Navier-Stokes solves. For this purpose, we utilize the following manufactured solutions

Table 2.1: Analysis of our Cahn-Hilliard solver at $t = 1$ demonstrates second order convergence.

	γ			
N	L_2		L_∞	
8	1.79e-05		2.96e-05	
16	4.40e-06	2.03	7.73e-06	1.93
32	1.23e-06	1.84	2.04e-06	1.92
64	3.03e-07	2.02	5.32e-07	1.94
128	7.65e-08	1.99	1.39e-07	1.93
256	1.98e-08	1.95	3.71e-08	1.91

Table 2.2: Analysis of our Cahn-Hilliard-Navier-Stokes solver at $t = 1$ demonstrates second order convergence.

	γ				u				v				p			
N	L_2		L_∞		L_2		L_∞		L_2		L_∞		L_2		L_∞	
8	1.86e-02		3.24e-02		6.88e-02		1.69e-01		1.40e-02		3.60e-02		1.60e-02		4.38e-02	
16	3.33e-03	2.48	1.20e-02	1.44	1.42e-02	2.28	4.12e-02	2.04	3.17e-03	2.14	9.93e-03	1.86	3.45e-03	2.21	1.31e-02	1.74
32	5.74e-04	2.54	2.43e-03	2.30	3.40e-03	2.06	1.01e-02	2.04	7.43e-04	2.09	3.05e-03	1.70	7.92e-04	2.12	3.64e-03	1.85
64	1.10e-04	2.38	2.80e-04	3.12	8.53e-04	1.99	2.54e-03	1.98	1.83e-04	2.02	7.22e-04	2.08	1.89e-04	2.07	6.32e-04	2.52
128	2.75e-05	2.00	7.30e-05	1.94	2.16e-04	1.98	6.14e-04	2.05	4.60e-05	1.99	1.83e-04	1.98	4.85e-05	1.96	1.39e-04	2.19
256	6.85e-06	2.01	1.90e-05	1.94	5.62e-05	1.94	1.40e-04	2.13	1.25e-05	1.88	4.80e-05	1.93	1.35e-05	1.84	3.76e-05	1.89

for γ , $\mathbf{u}$, and p , which we have adapted from the corresponding solution in [57]:

$$\gamma(x, y, t) = \frac{1}{2} (\cos(x) \cos(y) + 1) \sin(t) \quad (2.141)$$

$$u(x, y, t) = \sin(x) \cos(y) \cos(t) \quad (2.142)$$

$$v(x, y, t) = -\cos(x) \sin(y) \cos(t) \quad (2.143)$$

$$p(x, y, t) = \cos(x) \cos(y) \sin(t) \quad (2.144)$$

Note that these continuous solutions maintain a divergence-free velocity field, $\nabla \cdot \mathbf{u} = 0$.

Additionally, gravity is $g = (1, -2)$. We set the initial and boundary conditions using the manufactured solution. We measure the error for γ , $\mathbf{u}$, and p at $t = 1$. We determine the L_2 and L_∞ convergence by refining N from 8 to 256 with a constant $\frac{\Delta t}{\Delta x}$. As shown in

Table 2.2, our method exhibits second-order accuracy in γ , $\mathbf{u}$, and p .

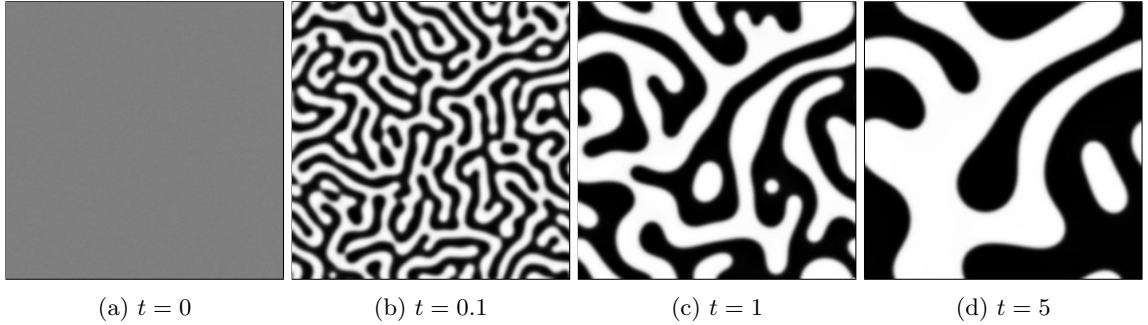


Figure 2.5: Binary fluid separation over time

2.5.2 Binary separation of phases

In this test, we demonstrate the evolution of the fluid phase separation over time with $\mathbf{u} = 0$ and $S_\gamma = 0$ using a random initial distribution for γ . We use the same setup as in [51]. In the following tests, we use a square domain $[0, 2\pi]^2$ that is periodic in all directions. γ is randomly distributed uniformly in the range $[0.475, 0.575]$. We use a time step of $\Delta t = 0.0001$ and cell size $\Delta x = 1/512$, and we run the test to $t = 5$. We select the Cahn-Hilliard parameters M , σ , and η such that $ML = \eta^2$, $\sigma = 1$, and $\eta^2 = 0.001$.

Figure 2.5 shows the increasing phase separation from $t = 0$ to $t = 5$. Our results are qualitatively similar to the results depicted in [51].

2.5.3 Discretization stability

In this test, we numerically evaluate the stability of our Cahn-Hilliard discretization and verify the analytical stability analysis we performed in Section 2.4.1. The first test we perform is to mimic the stability analysis setup, where we run Cahn-Hilliard without advection ($\mathbf{u} = 0$) using a linearized chemical potential derivative $F'(\gamma) = \frac{\gamma}{\eta^2}$, which mimics the scaling on the nonlinear potential. We use $[0, 1]^2$ with periodic boundary conditions as

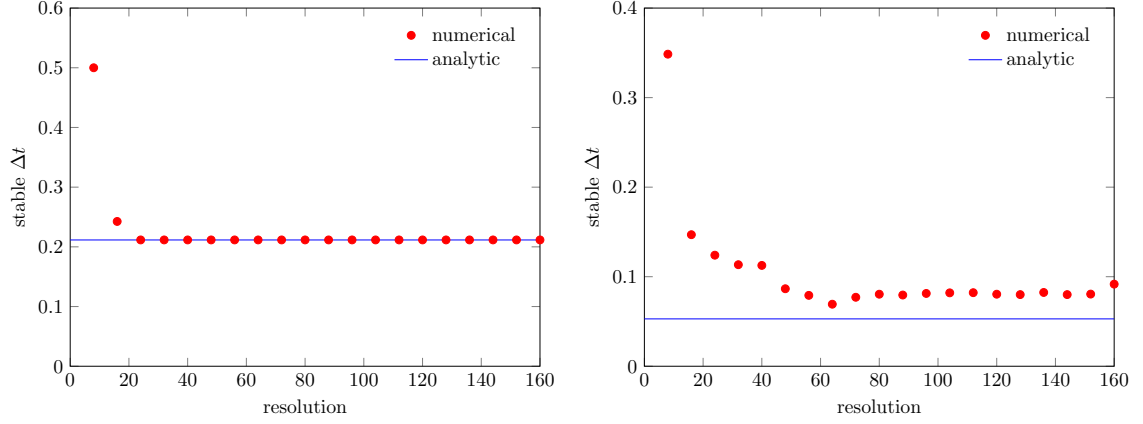


Figure 2.6: Numerically determined stable time step size compared with the analytically predicted for (left) linearized problem and (right) the fully nonlinear problem. The stable time step size does not depend significantly on resolution and is in close agreement with the analytical prediction.

our domain, and we fix $\eta = 0.01$, $M = 0.001$, and $\lambda = 0.001$. We do not perform the postprocessing step to keep γ in bounds. γ is initialized randomly between 0 and 1. If stable, this simulation will converge to a constant γ , whose value is equal to the average of the original values, which will be close to 0.5. If unstable, the simulation diverges. For the purposes of this test, we define a simulation to be unstable if during the first 10^4 time steps $\gamma \geq 2$ is observed. For each $\Delta x = \frac{1}{N}$ with $N = 8, 16, 24, \dots, 160$ we numerically search for the maximum stable time step size Δt_s and minimum unstable time step size Δt_u using binary search, stopping when $\Delta t_u - \Delta t_s < 10^{-4}$. Noting that $k = \eta^{-2} = 10^4$, the predicted stable time step size from (2.118) is approximately 0.2115 with no dependence on Δx . Numerically we computed $\Delta t_s \approx 0.2116$ and $\Delta t_u \approx 0.2117$ at all resolutions $N \geq 24$ and higher values for $N = 8, 16$. The results are shown in Figure 2.6. The numerically evaluated stable time step size is expected to be slightly larger than the analytical one, since there will be a range of time step sizes that are unstable but which take more than 10^4 time steps to diverge far enough to reach $\gamma = 2$.

We repeat the test with the same parameters and setup but using the original nonlinear chemical potential so that the system undergoes phase separation. We compute Δt_s and Δt_u as before. The results are shown in Figure 2.6. The stable time step stabilizes at $\Delta t_s \approx 0.07 - 0.08$ across resolutions, except for $N = 8 - 40$, where significantly larger time step sizes are stable. For the stable equilibria at $\gamma = 0$ and $\gamma = 1$ we have $k = F''(\gamma) = 2\eta^{-2}$, which corresponds to a stable time step size of 0.05288. We conclude from this that the linearized analysis provides a reasonable estimate of the stability of the nonlinear Cahn-Hilliard discretization in the absence of advection. This is perhaps not surprising, since $\gamma \approx 0$ or $\gamma \approx 1$ most of the time. In particular, the insensitivity of Δt_s to Δx is also observed in the nonlinear case.

2.5.4 Stationary circle

We simulate a stationary drop with zero initial velocity and pressure using the same parameters as [57]. In an ideal system, the pressure and surface tension forces balance, and therefore the drop should maintain its shape and remain stationary. Discretely solving the system introduces spurious currents, leading to deformation. This test quantifies these currents and observes the impact of altering the resolution, density ratio, viscosity, and surface tension. The test is carried out in a $[0, 1]^2$ domain and time step of $\Delta t = 0.064/N$ and cell size $\Delta x = 1/N$. All walls are modeled with free-slip boundary conditions. The simulation runs until $t = 10$.

The initial configuration features a drop with initial radius of $r = 0.2$, centered at $(0.5, 0.5)$. The drop, designated as fluid 1, has a density of $\rho_1 = 1000$ and a viscosity that varies according to test case. The density and viscosity of the fluid surrounding the

drop vary in each case. The specific values are detailed in Table 2.3 for density, viscosity and surface tension. Cases 2 and 4 investigate the effect of viscosity, while case 3 examines the impact of viscosity with a larger density ratio. Lastly, case 5 investigates the effects of surface tension.

We set $\eta = \eta_0(\Delta x/h_0)^{X^\eta} = \eta_0(Nh_0)^{-X^\eta}$ and $M = M_0(\eta/\eta_0)^{X^M} = \frac{M_0}{Nh_0}$ where $\eta_0 = h_0 = 1/32$, $M_0 = 10^{-5}$, $X^\eta = 2/3$ and $X^M = 3/2$. Note that we choose a larger value for M than what was used in [57] since we observed their M to be insufficient to maintain a steady interface using our method. We refine N from 16 to 256 and compute the L_2 and L_∞ errors by comparing the magnitude of velocity $||\mathbf{u}||_2$ averaged to each cell center with zero. The effect of refinement in each case is illustrated in Figure 2.7. We observe that refinement shows a decrease in the L_2 and L_∞ errors for most cases of slightly better than first order. However, in the case without viscosity, we note that the L_∞ error does not show much decrease with refinement.

The above test couples the parameters η and M to the refinement parameter N to test convergence to the sharp interface solution. This means that the coefficients of the PDE are changing under refinement, and therefore the test is not a proper convergence test. As further examination of the convergence properties of the proposed method on this test, we repeat the test with the PDE fixed for a proper refinement study. The results with fixed parameters $\eta = 0.02$ and $M = 10^{-6}$ are shown in Figure 2.8. We observe convergence closer to second order in both L_2 and L_∞ in this revised test and can conclude that while

Table 2.3: Material settings for different cases of the stationary circle test.

Case ID	ρ_0	ρ_1	μ_0	μ_1	σ
Case 1	1000	1000	0	0	1
Case 2	1000	1000	0.1	0.1	1
Case 3	1	1000	0.1	0.1	1
Case 4	1000	1000	0.0001	0.1	1
Case 5	1000	1000	0.1	0.1	10

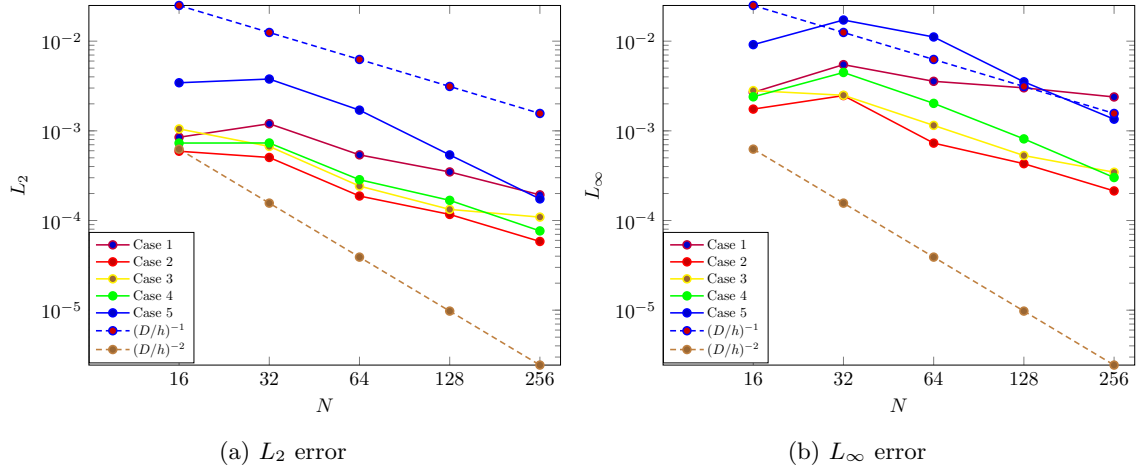


Figure 2.7: L_2 and L_∞ errors of velocity in the stationary circle test.

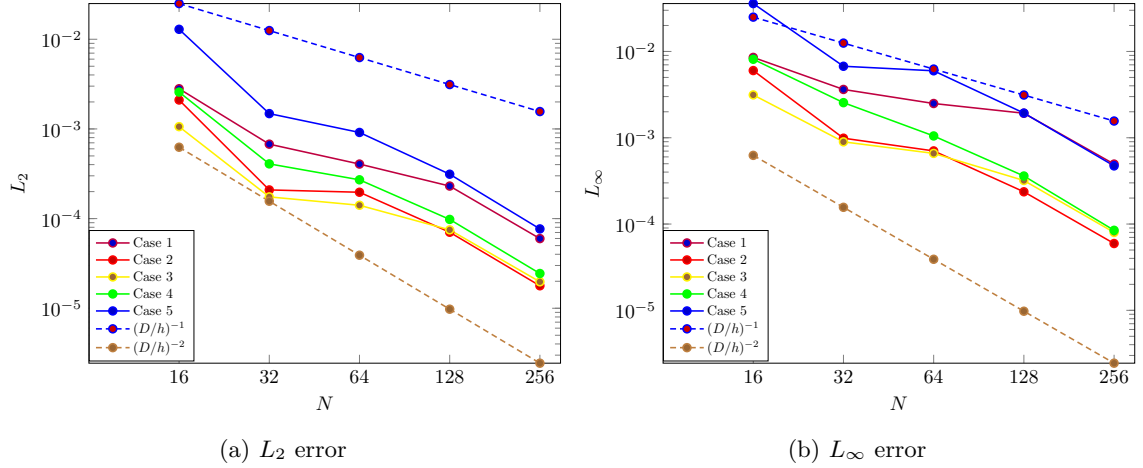


Figure 2.8: L_2 and L_∞ errors of velocity in the stationary circle test with fixed interface width.

the method is somewhat sensitive to the sharpness of the interface, it does converge under refinement.

2.5.5 Horizontal shear

The purpose of this test is to assess the conservation of mass, momentum, and kinetic energy within our method. We evaluate several scenarios with different densities and viscosities in the absence of external forces. We replicate the test outlined in [57]. The domain is $[0, 1]^2$, with periodic boundary conditions on all sides. The cell size is $\Delta x = 1/128$. The initial velocity and γ are given by

$$u(x, y, 0) = \begin{cases} \tanh\left(\frac{y-y_1}{\delta_1}\right), & \text{if } y \leq y_0 \\ \tanh\left(\frac{y_2-y}{\delta_1}\right), & \text{if } y > y_0 \end{cases} \quad (2.145)$$

$$v(x, y, 0) = \delta_2 \sin(kx) \quad (2.146)$$

$$\gamma(x, y, 0) = \begin{cases} \frac{1}{2} \left(1 + \tanh\left(\frac{y-y_1}{\delta_1}\right)\right), & \text{if } y \leq y_0 \\ \frac{1}{2} \left(1 + \tanh\left(\frac{y_2-y}{\delta_1}\right)\right), & \text{if } y > y_0 \end{cases} \quad (2.147)$$

with $y_0 = 0.5, y_1 = 0.25, y_2 = 0.75, \delta_1 = 1/30, \delta_2 = 0.5$ and $k = 2\pi$. As shown in Figure 2.9, fluid 1 is initialized in a band at the center of the domain with a velocity with positive x component, and fluid 0 is initialized elsewhere with a velocity with negative x component, leading to a strong velocity gradient at the interface. We test four different cases, as described in Table 2.4. In all cases, the surface tension is set to $\sigma = 10^{-12}$. We vary density and viscosity across the four cases. The time step is $\Delta t = 0.0005$ and the simulation ends at $t = 1$. We set $\eta = \delta_1/\sqrt{2}$ and $M = 10^{-7}$.

Mass and momentum should be conserved in all four cases investigated. Since surface tension is negligible, kinetic energy should be conserved in inviscid cases and gradually

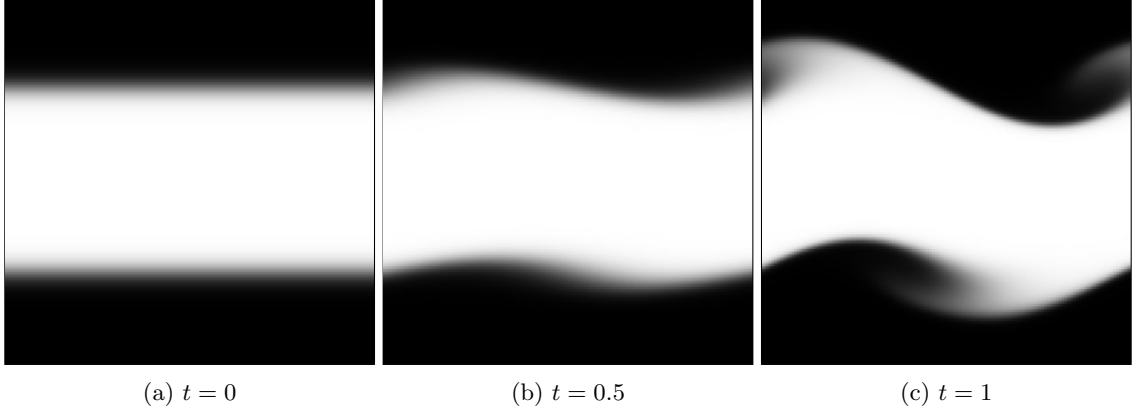


Figure 2.9: Horizontal Shear (Case 3).

decline in viscous cases. We calculate the total mass, momentum and kinetic energy as

$$m_{\text{total}} = \sum_{i,j} (\rho_0 + (\rho_1 - \rho_0) \gamma_{i,j}) \Delta\Omega \quad (2.148)$$

$$(m\mathbf{u})_{\text{total}}^x = \sum_{i,j} (\rho_{i-1/2,j}) (u_{i-1/2,j}) \Delta\Omega \quad (2.149)$$

$$E_k = \frac{1}{2} \sum_{i,j} \left((\rho_{i-1/2,j}) (u_{i-1/2,j})^2 + (\rho_{i,j-1/2}) (v_{i,j-1/2})^2 \right) \Delta\Omega, \quad (2.150)$$

where $\Delta\Omega = \Delta x \Delta y$ is the cell volume, and $(m\mathbf{u})_{\text{total}}^y$ is computed analogously.

The results of the tests are presented in Figure 2.10. Cases 1 and 3 exhibits zero mass change, while cases 2 and 4 have small decline with a magnitude of 10^{-12} . As expected, the inviscid cases maintain constant kinetic energy, while cases 3 and 4, with viscosity, show a gradual decline. Because our velocities are stored on faces instead of cell centers, we compute our kinetic energy differently from [57], but we observe similar results. We observe almost no change in momentum for cases without viscosity, and we see a minor change in viscous cases.

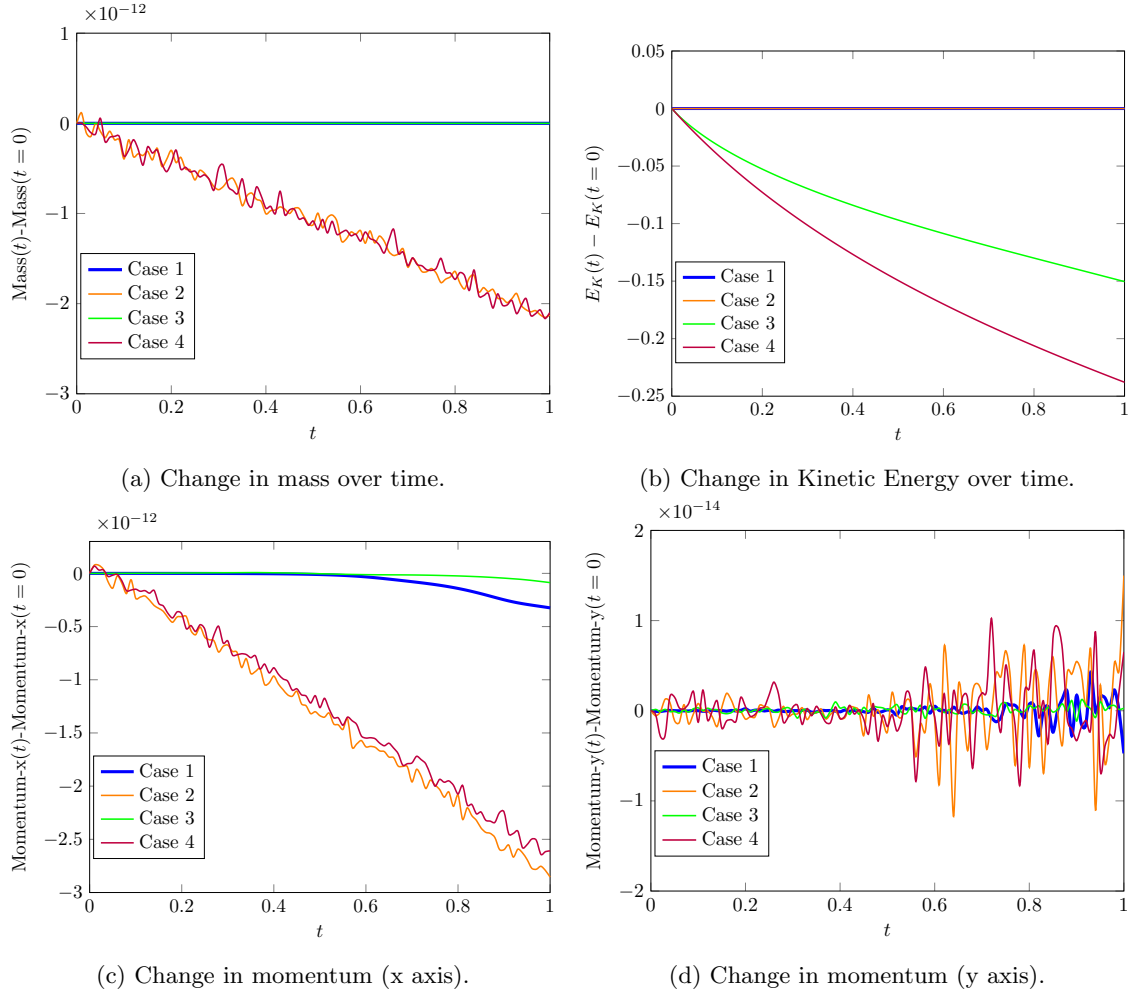


Figure 2.10: Results of the horizontal shear test for Cases 1-4.

Shearing artifacts with large gradients

We observe some artifacts at the interface at the tested resolutions for inviscid cases 1 and 2, as illustrated in Figure 2.11. Increasing the resolution from 128 to 512 demonstrates

Table 2.4: Material settings for different cases of the horizontal shear.

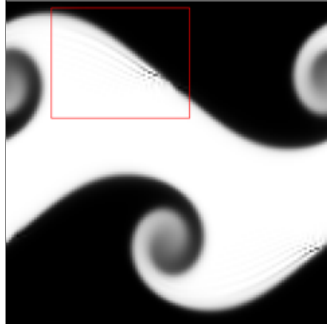
Case ID	ρ_0	ρ_1	μ_0	μ_1
Case 1	1	1	0	0
Case 2	1	10	0	0
Case 3	1	1	0.001	0.01
Case 4	1	10	0.001	0.01

that the artifacts go away under refinement. The artifact is caused by consistent advection in the presence of sharp γ gradients and shearing flow. The artifact is not present if WENO advection is used for momentum advection, however that scheme fails to preserve numerical consistency of mass and momentum. Strengthening the effects of Cahn-Hilliard also eliminates the artifact. It is unknown if these artifacts were also present in [57], as they only give bulk averaged measurements (which are consistent with ours), which do not reveal the artifacts we observed.

2.5.6 Translating circle

This test examines the accuracy of advection in our method at different density ratios, based on the method used in [57]. We use a $[0, 1]^2$ domain, with a cell size of $\Delta x = 1/128$ and the time step of $\Delta t = \Delta x/10$, with periodic boundary conditions in all axes. The initial configuration of the test comprises a circular drop of fluid 1 with a radius of $r = 0.1$ centered at $(0.5, 0.5)$. The initial velocity is $\langle 1, 1 \rangle$ everywhere. The simulation ends at $t = 1$, when the drop is expected to return to the initial position. Fluid 1 is tested with varying densities of $\rho_1 = 1, 10^3, 10^6, 10^9$. The surrounding fluid 0 has density $\rho_0 = 1$. The viscosity of both fluids is zero. All density ratios are tested with surface tension values set to $\sigma = 1$ and $\sigma = 10^{-12}$. The test is carried out in the absence of external forces. We set $\eta = 3\Delta x$ and $M = 10^{-7}$. The material settings are summarized in Table 2.5.

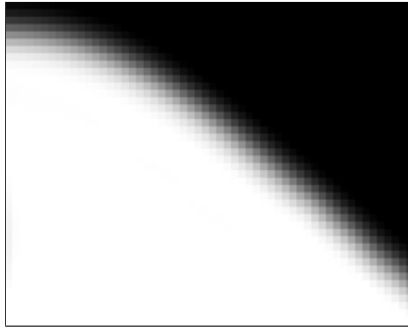
In the ideal solution of the test, the drop returns to its initial location, preserving its original circular shape. The velocity remains constant throughout the simulation for all cases. Pressure maintains its initial value of zero in the absence of surface tension. However,



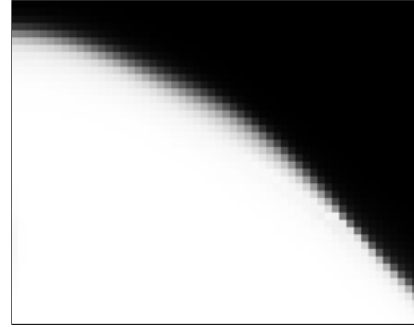
(a) One of the regions where artifacts can be seen in Case 1.



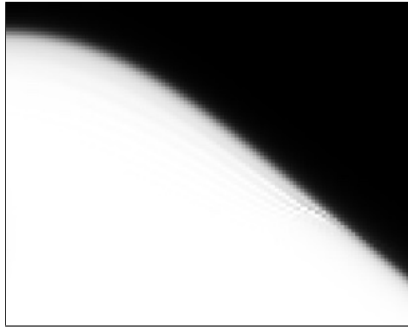
(b) A closer look at that region (a).



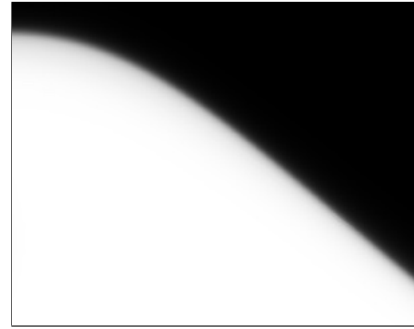
(c) The same region (a) if Case 1 is repeated with Cahn-Hilliard parameters $M = 10^{-5}$ and $\sigma = 1$ (no surface tension forces are applied in this example).



(d) The same region (a) if Case 1 is repeated using WENO advection for momentum advection.



(e) The same region (a) if Case 1 is repeated with $N = 256$.



(f) The same region (a) if Case 1 is repeated with $N = 512$.

Figure 2.11: Artifacts in horizontal shear Case 1, and the effects of different parameters and methods on those artifacts.

in the presence of surface tension, a pressure jump should be observed around the surface of the bubble. A snapshot of velocity streamlines at the conclusion of the simulation is depicted for cases without surface tension in Figure 2.12, and with surface tension in Figure 2.13.

Table 2.5: Material settings for different cases of the translating circle test (surface tension, density).

Case ID	ρ_0	ρ_1	σ
Case 1	1	1	10^{-12}
Case 2	1	10^3	10^{-12}
Case 3	1	10^6	10^{-12}
Case 4	1	10^9	10^{-12}
Case 5	1	1	1
Case 6	1	10^3	1
Case 7	1	10^6	1
Case 8	1	10^9	1

Table 2.6: Error results for the advection test without surface tension.

Variable	Norm	$\rho_1/\rho_0 = 1$	$\rho_1/\rho_0 = 10^3$	$\rho_1/\rho_0 = 10^6$	$\rho_1/\rho_0 = 10^9$
u	L_2	8.81e-13	2.60e-13	3.49e-11	6.07e-08
	L_∞	7.36e-12	1.87e-12	2.55e-10	1.69e-06
v	L_2	8.81e-13	2.60e-13	3.49e-11	6.06e-08
	L_∞	7.36e-12	1.88e-12	2.55e-10	1.57e-06

The drop consistently preserves its shape until the end of the simulation, and velocities remain almost constant, aligning with their initial streamlines in both cases. As a result, the streamlines in Figures 2.12 and 2.13 are nearly indistinguishable.

The L_2 and L_∞ errors for velocities are detailed in Tables 2.6 and 2.7 for cases without and with surface tension, respectively. The error is computed by comparing the final velocities with the initial value. We observe that our error is inversely correlated with the density ratio in cases without surface tension and they are consistent with the findings reported by Huang et al. [57], and the errors in the cases with surface tension are somewhat better.

Table 2.7: Error results for the advection test with surface tension.

Variable	Norm	$\rho_1/\rho_0 = 1$	$\rho_1/\rho_0 = 10^3$	$\rho_1/\rho_0 = 10^6$	$\rho_1/\rho_0 = 10^9$
u	L_2	7.93e-04	2.25e-05	7.44e-07	8.47e-08
	L_∞	6.42e-03	1.50e-04	5.40e-06	1.16e-06
v	L_2	7.93e-04	2.25e-05	7.44e-07	8.47e-08
	L_∞	6.42e-03	1.50e-04	5.40e-06	1.27e-06

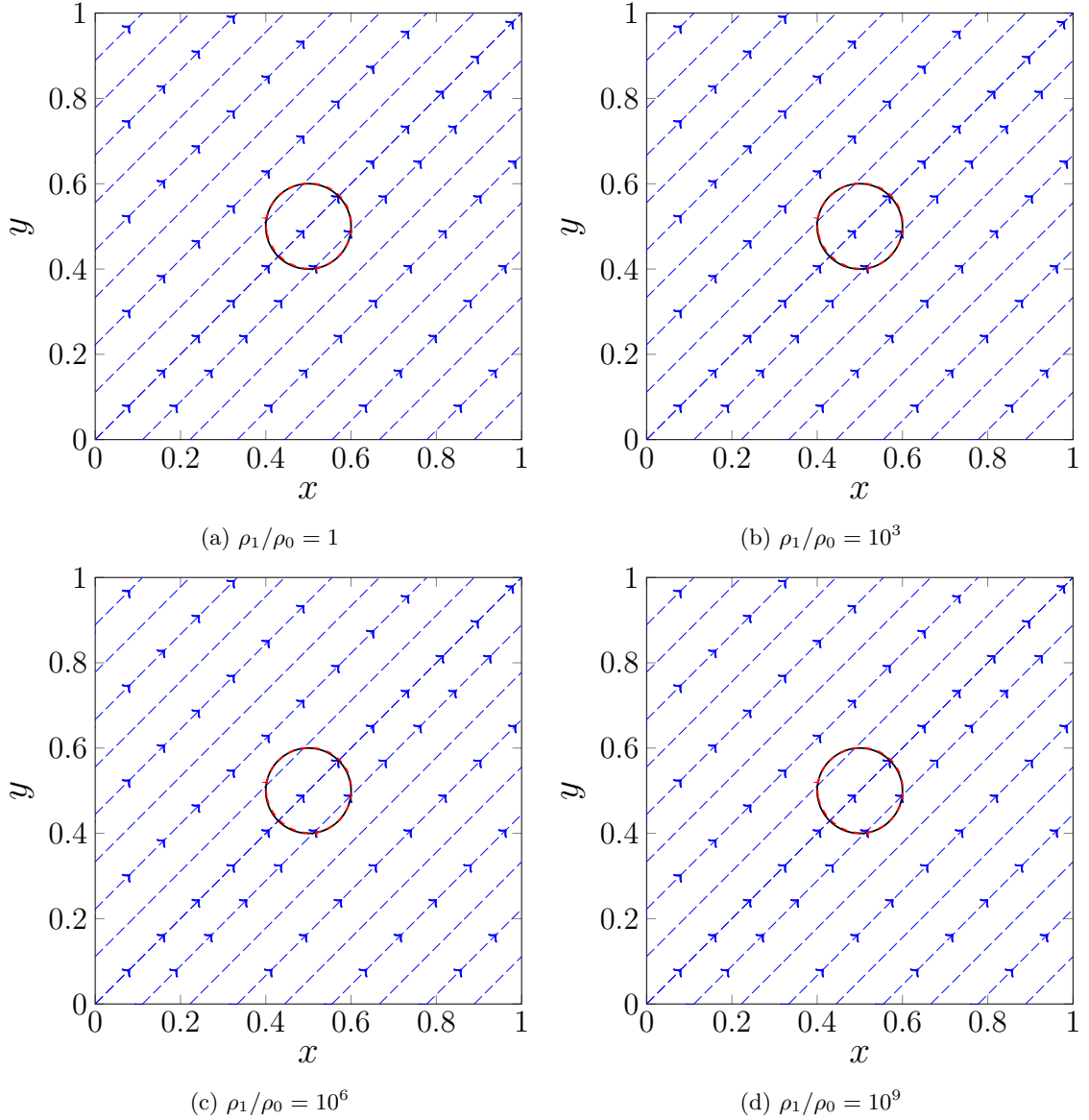


Figure 2.12: Initial and final ($t = 1$) state of the drop with $\sigma = 10^{-12}$. The solid black line is the initial state and the dotted red line is the final state.

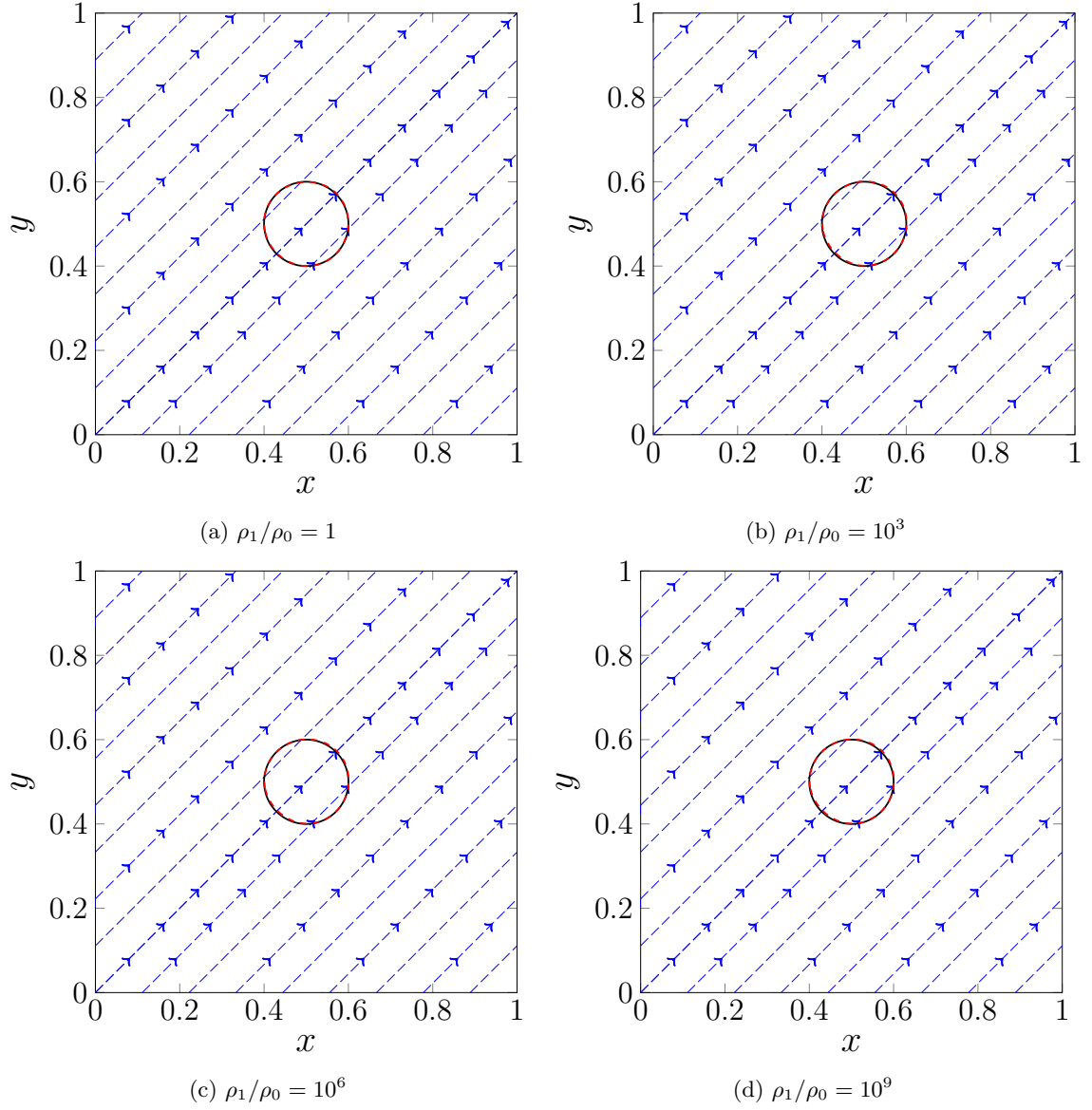


Figure 2.13: Initial and final ($t = 1$) state of the drop with $\sigma = 1$. The solid black line is the initial state and the dotted red line is the final state.

2.5.7 Rising air bubble

The primary objective of this experiment is to assess the convergence of our method toward the sharp interface solution within a real-world scenario characterized by a substantial difference in density and viscosity. This evaluation holds particular significance because

it examines the convergence toward the sharp interface solution by adjusting the parameter η while simultaneously modifying the resolution N . As suggested by the findings of Huang et al. [57], increasing the resolution while keeping η constant leads to the experiment's results converging toward a solution with a transitional region of length $O(\eta)$. However, by decreasing η while simultaneously augmenting the resolution, we ensure the convergence toward the desired sharp interface solution.

We examine a $[0, 1] \times [0, 2]$ domain with cell size $\Delta x = 1/N$. The top and bottom walls exhibit a no-slip boundary condition, while the left and right walls follow a slip boundary condition. The initial setup involves a bubble with an initial diameter of 0.5 positioned at the center (0.5, 0.5) and assigned a zero initial velocity. Fluid 0 within the bubble has properties $\rho_0 = 1$ and $\mu_0 = 0.1$, whereas the surrounding fluid 1 has $\rho_1 = 1000$ and $\mu_1 = 10$. This results in a density ratio of 1000 and a viscosity ratio of 100. Surface tension σ is 1.96, and gravity is $(0, -0.98)$. The time step size is $\Delta t = 0.128/N$ and the simulation stops at $t = 1$. We reduce the length of the transition region by setting $\eta = \eta_0(Nh_0)^{-X_\eta}$ where $\eta_0 = h_0 = 1/32$. $X_\eta \leq 1$ so that we have the same number of cells in the transition region (or more) as we refine [57]. We set the mobility variable $M = M_0(\eta/\eta_0)^{X_M}$ where $M_0 = 10^{-7}$. Based on [61], X_M must be $1 \leq X_M < 2$.

We conduct six sets of tests and run each set with $N = 16, 32, 64, 128, 256$. In the first 5 sets of tests, we set $X_M = 1$ and $X_\eta = 0, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, 1$ to test the effects of X_η . In the last test, we set $X_M = \frac{3}{2}, X_\eta = \frac{2}{3}$ which makes $M \sim \Delta x$. To test the convergence of all test

cases we compute the following three parameters:

$$\psi_c = \frac{P_a}{P_b} = \frac{2\sqrt{\int_{\gamma < \frac{1}{2}} \pi d\Omega}}{P_b} \quad (2.151)$$

$$y_c = \frac{\int_{\Omega} y(1 - \gamma) d\Omega}{\int_{\Omega} (1 - \gamma) d\Omega} \quad (2.152)$$

$$v_c = \frac{\int_{\Omega} v(1 - \gamma) d\Omega}{\int_{\Omega} (1 - \gamma) d\Omega}. \quad (2.153)$$

In (2.151), P_a is the perimeter of the circle with the same area as the bubble, P_b is the perimeter of the bubble and ψ_c is the circularity of the bubble. P_b is computed by discretizing the interface surface using a marching squares algorithm and computing the total length between successive nodes. $\psi_c = 1$ when the shape of the bubble is a circle and $\psi_c < 1$ when the bubble deforms [57]. y_c is center of mass and v_c is the velocity of center mass in the y direction (rising velocity).

In the case where $X_M = 1, X_\eta = 0$, we expect the simulation to converge to the exact solution of the phase-field model where $\eta = \eta_0$ since setting $X_\eta = 0$ results in using the same value for η in all of the simulations [57]. We show our results for this particular case in Figure 2.14. We observe that our results match the results shown in [57]. In the case where $X_\eta \neq 0$ we expect the simulation to converge to a sharp interface solution. We compare our results for each case where $N = 256$ in Figure 2.15. The error and convergence order of each case can be found in Table 2.8 where we compare all cases with $N = 16$ to $N = 128$ to the results from the simulation where $N = 256$. We observe second order convergence where $X_\eta = 0$ and maintain an order of about 1.5-2.0 on all other cases.

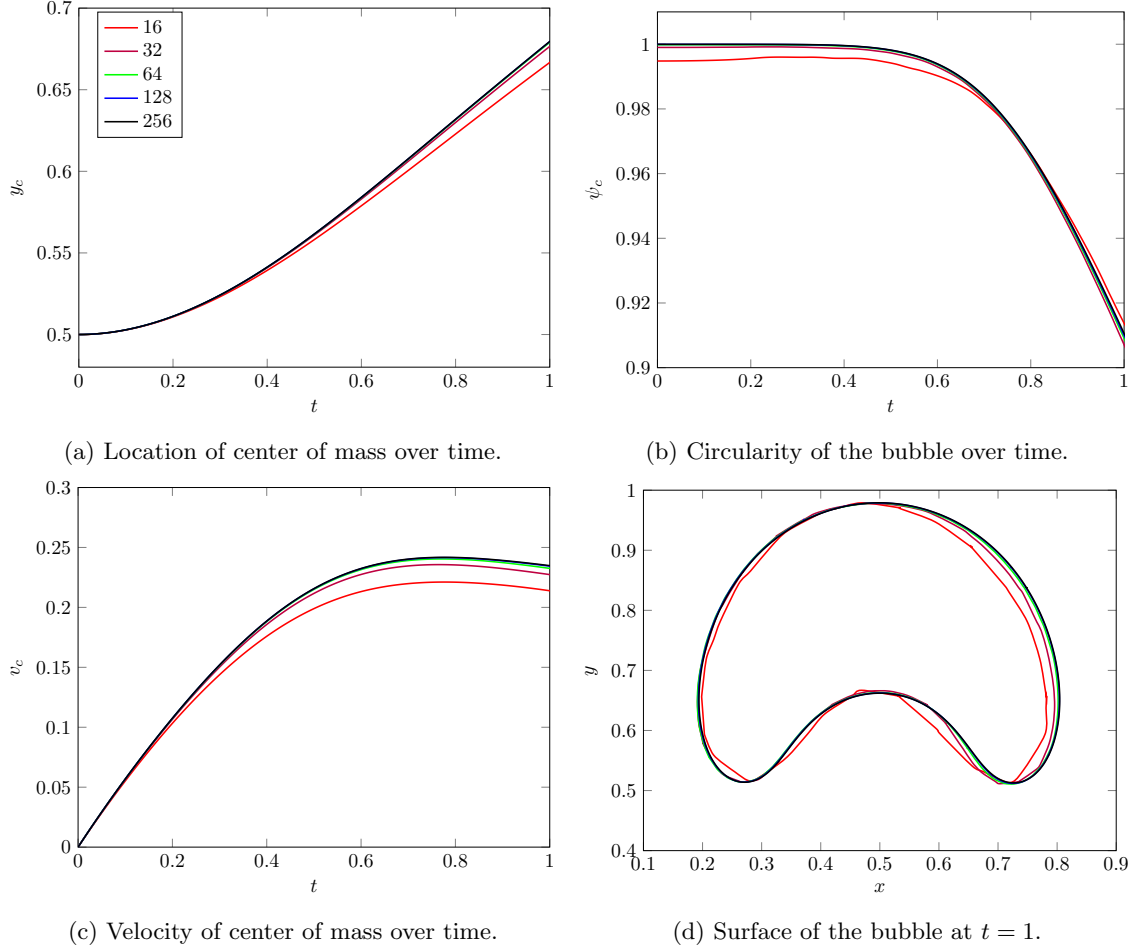


Figure 2.14: Rising bubble test results with $X_\eta = 0, X_M = 1$.

2.5.8 Comparisons of surface tension

The aim of this test is to compare different surface tension methods to determine how well they show expected properties. We measure the direction and location of surface tension in Section 2.5.8 and its effects on pressure in Section 2.5.8. We explore the impact of introducing $g(\gamma)$ as detailed in Section 2.3.2 and varying the parameter a in (2.47). The specific functions utilized in our comparative tests are outlined in Table 2.9. In cases 1 and 2 we use the surface tension used by Shen and Yang [112] and Huang et al. [57]

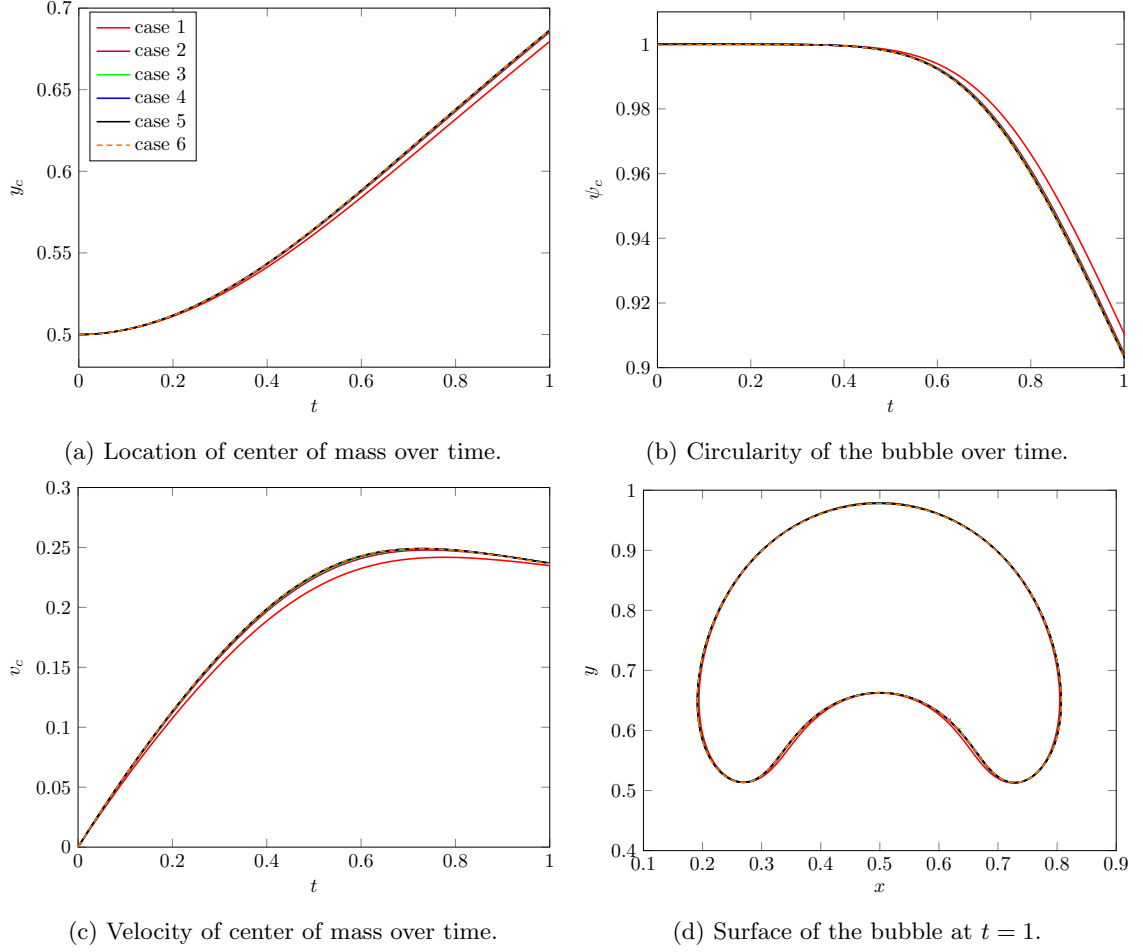


Figure 2.15: Rising bubble test results of test described in table 2.8 with $N = 256$.

respectively. Cases 3 to 7 we apply the function introduced in (2.47), varying the values of a as $a = 0, 0.1, 0.2, 0.3, 0.5$. In cases 8 and 9 we use the same scheme as cases 3 to 7 using a different functions as $g(\gamma)$. In all the tests described we set $\sigma = 1$.

We conduct the test in a $[0, 1]^2$ domain with a cell size of $\Delta x = 1/256$, where all the walls feature slip boundary conditions. We use $\Delta t = 0.00025$ and run the simulation until $t = 10$. The initial configuration features a bubble with an initial radius of $r = 0.2$, centered at $(0.5, 0.5)$, with zero initial velocity. The fluid inside the bubble and the surrounding fluid both have a density of $\rho_0 = \rho_1 = 1000$ and viscosity of $\mu_0 = \mu_1 = 0$.

	Grid	$X_\eta = 0, X_M = 1$		$X_\eta = \frac{1}{3}, X_M = 1$		$X_\eta = \frac{1}{2}, X_M = 1$		$X_\eta = \frac{2}{3}, X_M = 1$		$X_\eta = \frac{2}{3}, X_M = \frac{3}{2}$		$X_\eta = 1, X_M = 1$	
		L_2		L_2		L_2		L_2		L_2		L_2	
ψ_c	16	3.54e-03		7.60e-03		9.70e-03		1.22e-02		1.21e-02		1.88e-02	
	32	2.50e-03	0.51	1.85e-03	2.04	2.26e-03	2.10	2.46e-03	2.30	2.45e-03	2.31	2.61e-03	2.85
	64	1.06e-03	1.24	8.01e-04	1.21	8.01e-04	2.10	9.07e-04	1.44	9.36e-04	1.39	1.20e-03	1.12
	128	3.92e-04	1.44	3.00e-04	1.42	2.58e-04	1.63	4.47e-04	1.02	4.86e-04	0.95	6.94e-04	0.80
y_c		L_2		L_2		L_2		L_2		L_2		L_2	
	16	9.24e-03		1.50e-02		1.69e-02		1.88e-02		1.87e-02		2.36e-02	
	32	2.27e-03	2.03	6.23e-03	1.27	6.86e-03	1.30	7.12e-03	1.40	7.10e-03	1.40	7.17e-03	1.72
	64	4.61e-04	2.30	2.65e-03	1.23	2.68e-03	1.30	2.47e-03	1.53	2.45e-03	1.53	1.92e-03	1.90
v_c	128	6.49e-05	2.83	9.49e-04	1.48	8.64e-04	1.63	7.05e-04	1.81	6.96e-04	1.82	4.26e-04	2.17
		L_2		L_2		L_2		L_2		L_2		L_2	
	16	1.75e-02		2.42e-02		2.65e-02		2.89e-02		2.89e-02		3.53e-02	
	32	5.71e-03	1.62	1.00e-02	1.27	1.07e-02	1.31	1.10e-02	1.40	1.10e-02	1.40	1.10e-02	1.68
v_c	64	1.68e-03	1.76	4.06e-03	1.30	4.05e-03	1.31	3.75e-03	1.55	3.76e-03	1.55	2.99e-03	1.88
	128	3.00e-04	2.49	1.36e-03	1.58	1.25e-03	1.70	1.06e-03	1.82	1.07e-03	1.81	6.83e-04	2.13

Table 2.8: L_2 error and order compared to $N = 256$ on all the cases for rising air bubble test.

Table 2.9: Material settings for different cases for the surface tension and pressure comparison test.

Case ID	Surface Tension	a
Case 1	$\lambda \nabla \cdot (\nabla \gamma \otimes \nabla \gamma)$	0
Case 2	$4\xi \nabla \gamma$	
Case 3	$\sigma \kappa \nabla(g(\gamma))$	
Case 4	$\sigma \kappa \nabla(g(\gamma))$	
Case 5	$\sigma \kappa \nabla(g(\gamma))$	0.1
Case 6	$\sigma \kappa \nabla(g(\gamma))$	0.2
Case 7	$\sigma \kappa \nabla(g(\gamma))$	0.3
Case 8	$\sigma \kappa \nabla(\frac{1}{2}(1 + \tanh(4\gamma - 2)))$	0.5
Case 9	$\sigma \kappa \nabla \gamma$	

We define η and M similar to Section 2.5.7, with $M_0 = 10^{-5}$, $\eta_0 = h_0 = 1/8$, $X^\eta = 2/3$ and $X^M = 3/2$. This gives $\eta = 0.03125$ and $M = 1.25 \times 10^{-6}$, which we use for all cases.

Magnitude and angle

Visual representations of the test results are shown in Figures 2.16 and 2.17. The angle between the surface normal and the surface tension force is measured in radians. The contour $\gamma = 0.5$ representing the center of the interface is displayed as a dashed circle.

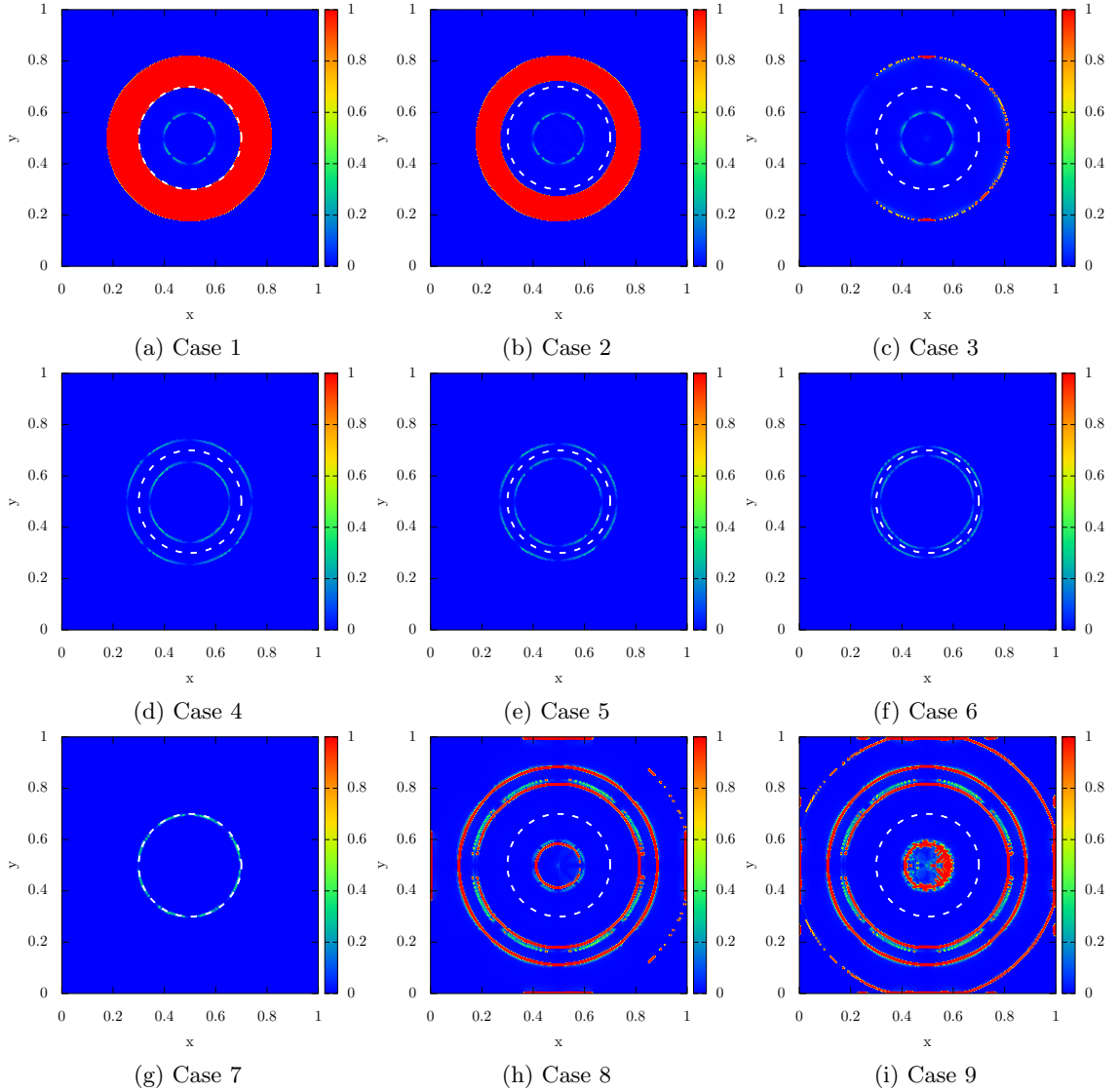


Figure 2.16: Comparisons of the angle between surface tension force and surface normal (radians) for different surface tension formulations.

Notably, cases 1 and 2 show strong disagreement between the surface tension direction and normal just beyond the interface center, and case 1 shows negligible magnitude at the interface center. Cases 2, 3, 8, and 9 all show strong disagreement in the angle far from the interface, but the magnitude of the surface tension in those areas is negligible as expected.

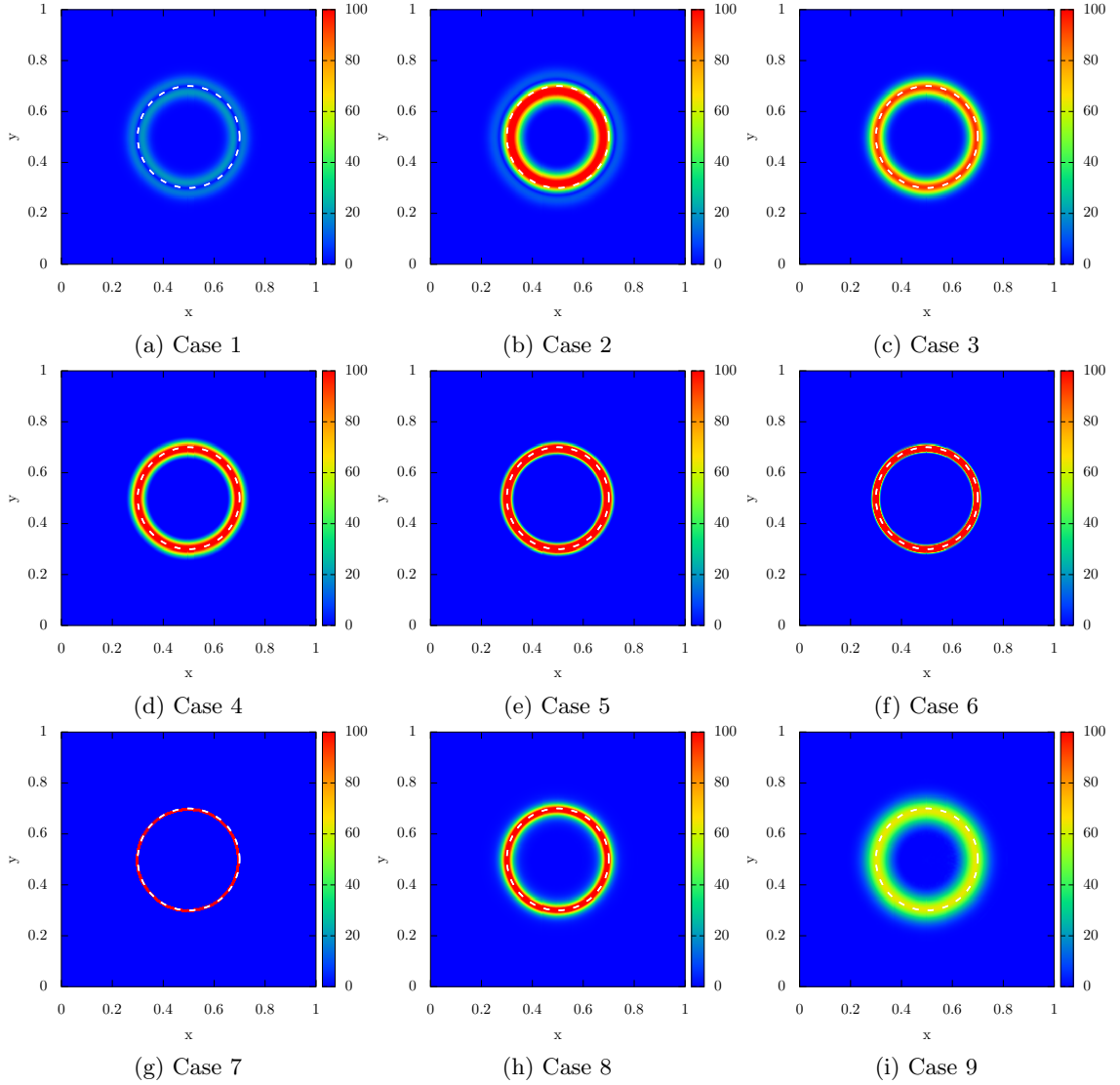


Figure 2.17: Comparisons of the magnitude of surface tension force for different surface tension formulations.

Cases 3 through 7 show the effects of the a parameter in our surface tension method. We observe that, as expected, the areas where the surface tension magnitude is nonzero begins to closely fit the interface center contour as a approaches 0.5. In the case where $a = 0.5$, the surface tension force closely matches the interface contour, as we would expect from a sharp interface solution. These cases also show the advantage of a finite

support around the interface (in cases 4-7), which gives a surface tension force vector $\mathbf{0}$ away from the interface. In cases 3 (where $a = 0$), 8, and 9, we see that the surface tension is still being calculated away from the interface. While the magnitudes are negligible in these areas, they are not 0.

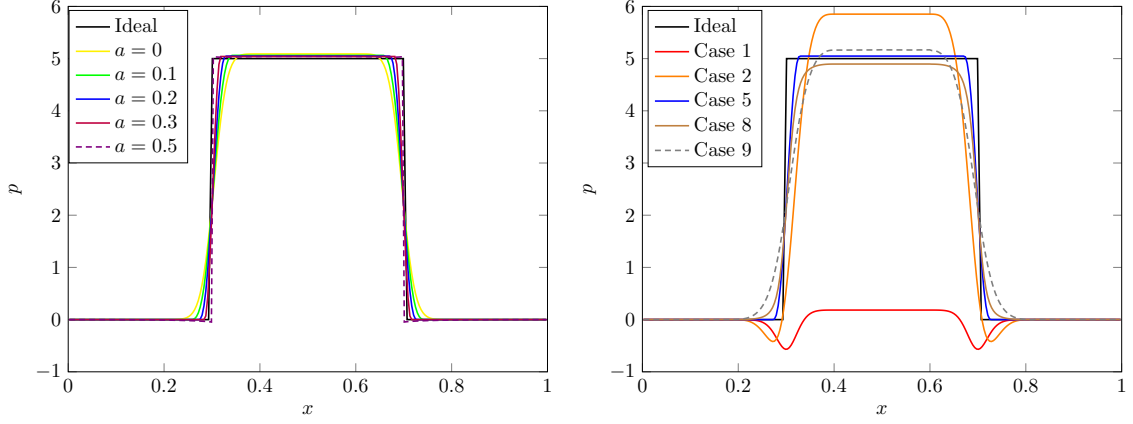
Effects on pressure

Ideally, we anticipate the pressure to be a constant value inside and outside of the bubble with a pressure jump on the surface. We compare our results to the ideal physical pressure along the x -axis at $y = 0.5$ in Figure 2.18.

We show in Figure 2.18a that as a goes from 0 to 0.5, the results move towards the ideal pressure jump caused by a sharp interface. All parameters of a produce acceptable results for pressure, which makes $\sigma\kappa\nabla g(\gamma)$ the best of the methods by this criteria. Figure 2.18b shows that cases 8 and 9, which are similar to our preferred method, are also reasonably close to the ideal pressure. Note that in Case 1, the formulation of the surface force introduces a gradient term which will be compensated for in the pressure correction; the plot for Case 1 shows the pseudo-pressure which includes the compensating term [114]. Case 2 deviates significantly (about 15%) from the ideal in our tests; this is consistent with Huang et al. [57], which reported errors of about 9% on a similar test at lower resolution.

2.5.9 Rayleigh-Taylor instability

We simulate the classic Rayleigh-Taylor instability, where a denser fluid is initially positioned above a lighter one in the presence of gravitational force. This instabil-



(a) Comparing pressure in cases using $\sigma\kappa\nabla(g(\gamma))$ for surface tension. (b) Comparing pressure in the remaining cases with case 5 ($a = 0.2$).

Figure 2.18: Change in pressure in the x axis at $y = 0.5$ of Section 2.5.8.

ity test serves as a benchmark for evaluating the behavior of two-phase flows. The tests was first investigated by Rayleigh [104] and is normally characterized by “Atwood ratio” $A_t = (\rho_1 - \rho_0)/(\rho_1 + \rho_0)$, a measure of the density difference between the two phases. Daly [22] conducted a numerical analysis of the test. Long-time instability analysis of the test for inviscid incompressible fluids without surface tension with $A_t = 0.5$ is done by Tryggvason [129]. We examine a $[0, 1] \times [0, 4]$ domain with cell sizes $\Delta x = 1/128$. Free-slip boundary conditions are enforced at the top and bottom walls, while the left and right walls are subject to periodic boundary conditions. The initial setup involves a heavier fluid with density $\rho_1 = 3, 30, 1000, 3000$ and viscosity $\mu_1 = 0.001$ positioned on top of a lighter fluid with density $\rho_0 = 1$ and viscosity $\mu_0 = 0.001$. The velocity is initialized to zero, and the initial interface between the two fluids is situated at $y = 2 + 0.1 \cos(2\pi x)$. The specified parameters for the simulation include a surface tension $\sigma = 10^{-12}$, gravitational force vector $\langle 0, -1 \rangle$, and a time step size of $\Delta t = 5 \times 10^{-4}/\sqrt{A_t}$. The simulation concludes at $t\sqrt{A_t} = 8$. We set M so that $\lambda M = 10^{-15}$ and $\eta = 0.01$.

Snapshots of the test with $\rho_1 = 3$ are presented in Figure 2.19, offering a comparative analysis with the outcomes from our implementation of Huang et al. [57]’s method. Notably, our simulation closely aligns with the results obtained by Huang et al. [57]. Figure 2.20 provides a visual depiction of the Rayleigh-Taylor test’s progression at a density ratio of 3, while Figure 2.21 showcases the evolution of simulations across various density ratios. In both Figure 2.20 and Figure 2.21, the simulations maintain symmetry throughout, even as the interface develops intricate shapes in later stages. As the density ratio increases, the simulation exhibits accelerated movement and a simpler evolution. Due to the minor impact of viscosity in these simulations, a stable state is not reached before the simulations conclude at $t\sqrt{A_t} = 8$.

In Figure 2.22 we plot a quantitative measurement of the test by measuring the highest and lowest point of the interface between the two fluids with density ratio of 3 from $t = 0$ to $t = 2.5$. Our results are comparable to the results reported by Huang et al. [57], [24] and Guermond and Quartapelle [42] which use similar values for density and viscosity.

2.5.10 Dam break

The dam break test is employed to assess the method’s performance in scenarios characterized by high density ratios. In this test, we simulate the classic test as in Martin and Moyce [90] where a specific volume of water is confined within a rectangular box of dimensions $[a \times a]$, enclosed by walls on both the left and right sides. In the initial frame, the right wall of the box containing the water, is removed, leading to the observation of

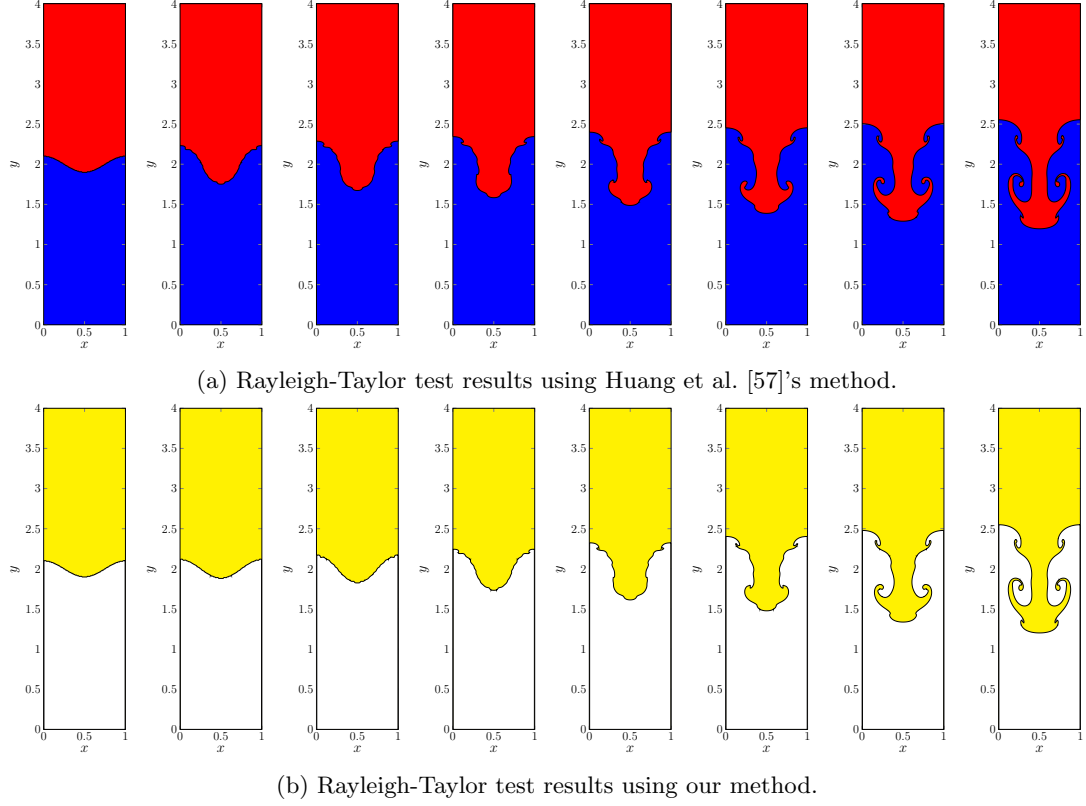


Figure 2.19: Rayleigh-Taylor test at times $t = 0, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5$ and density ratio of 3.

the gravitational evolution of water movement within a larger $[0, 4a] \times [0, 2a]$ domain. We examine this domain with cell sizes $\Delta x = a/64$ where $a = 0.05715$. The all the walls are subject to a no-slip boundary condition. The initial setup involves a heavier fluid (water) with density $\rho_0 = 998.207$ and viscosity $\mu_0 = 1.002 \times 10^{-3}$ confined in a $[a \times a]$ box surrounded by a lighter fluid (air) with density $\rho_1 = 1.204$ and viscosity $\mu_1 = 1.78 \times 10^{-5}$ with initial velocity of zero. The specified parameters for the simulation include a surface tension $\sigma = 7.28 \times 10^{-2}$, gravitational force vector $\langle 0, -9.8 \rangle$, and a time step size of $\Delta t = 7.1437 \times 10^{-5}$. The simulation concludes at $t = 10$. We set M so that $\lambda M = 10^{-7}$ and $\eta = 0.01a$.

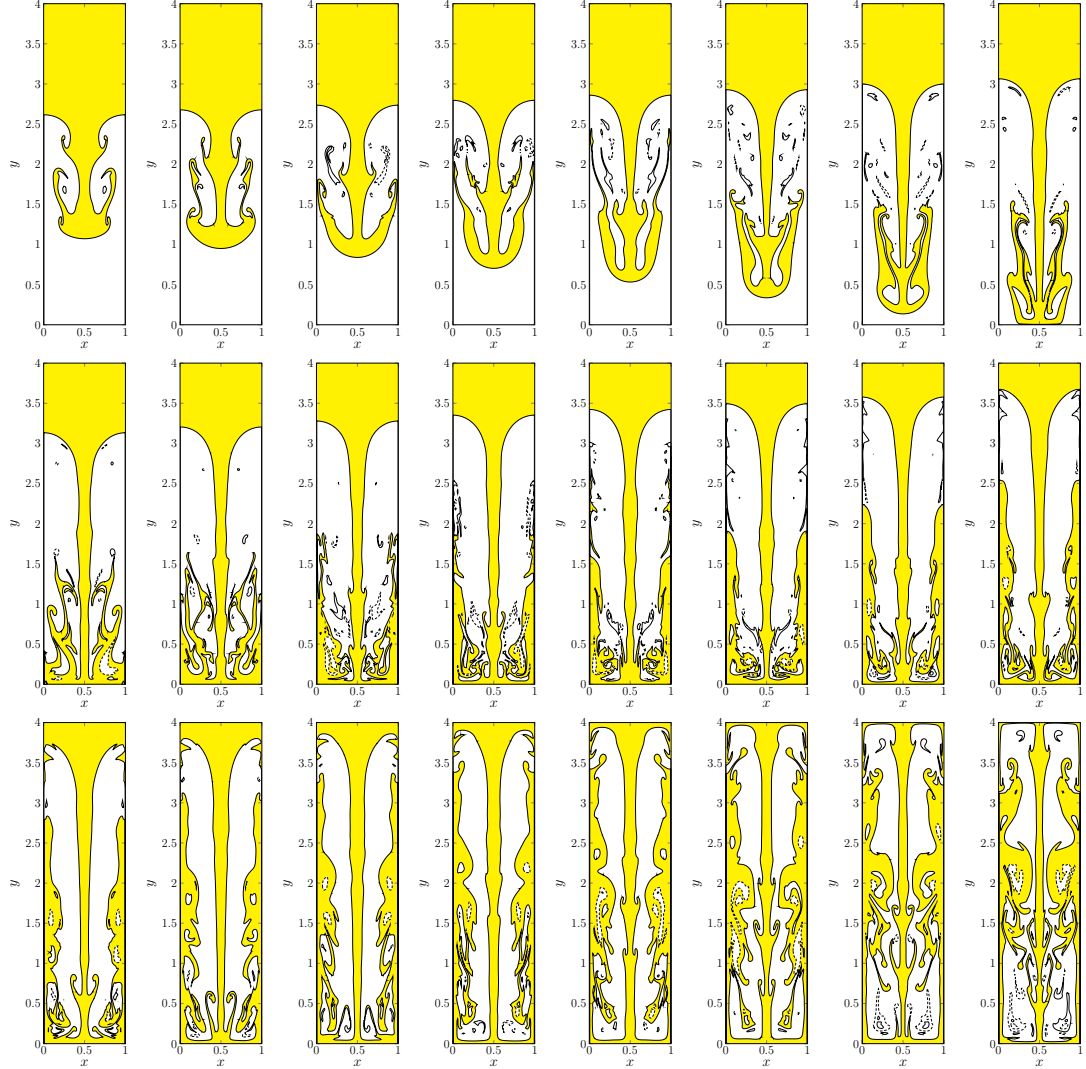


Figure 2.20: Rayleigh-Taylor test results with $\rho_0 = 3$, from $t = 2.25$ to $t = 8$ in 0.25 increments.

As previously noted, the simulation begins with zero velocity, and at time $t = 0$, removing the box containing the denser fluid instigates fluid motion due to gravity. As the simulation progresses, the denser fluid reaches the right wall of the domain, where it is pushed to rise along the wall before crashing back down. The simulation attains a stable condition by the conclusion of the simulation. Visual snapshots of distinct stages within the simulation are shown in Figures 2.24 and 2.25. We non-dimensionalize t by dividing it

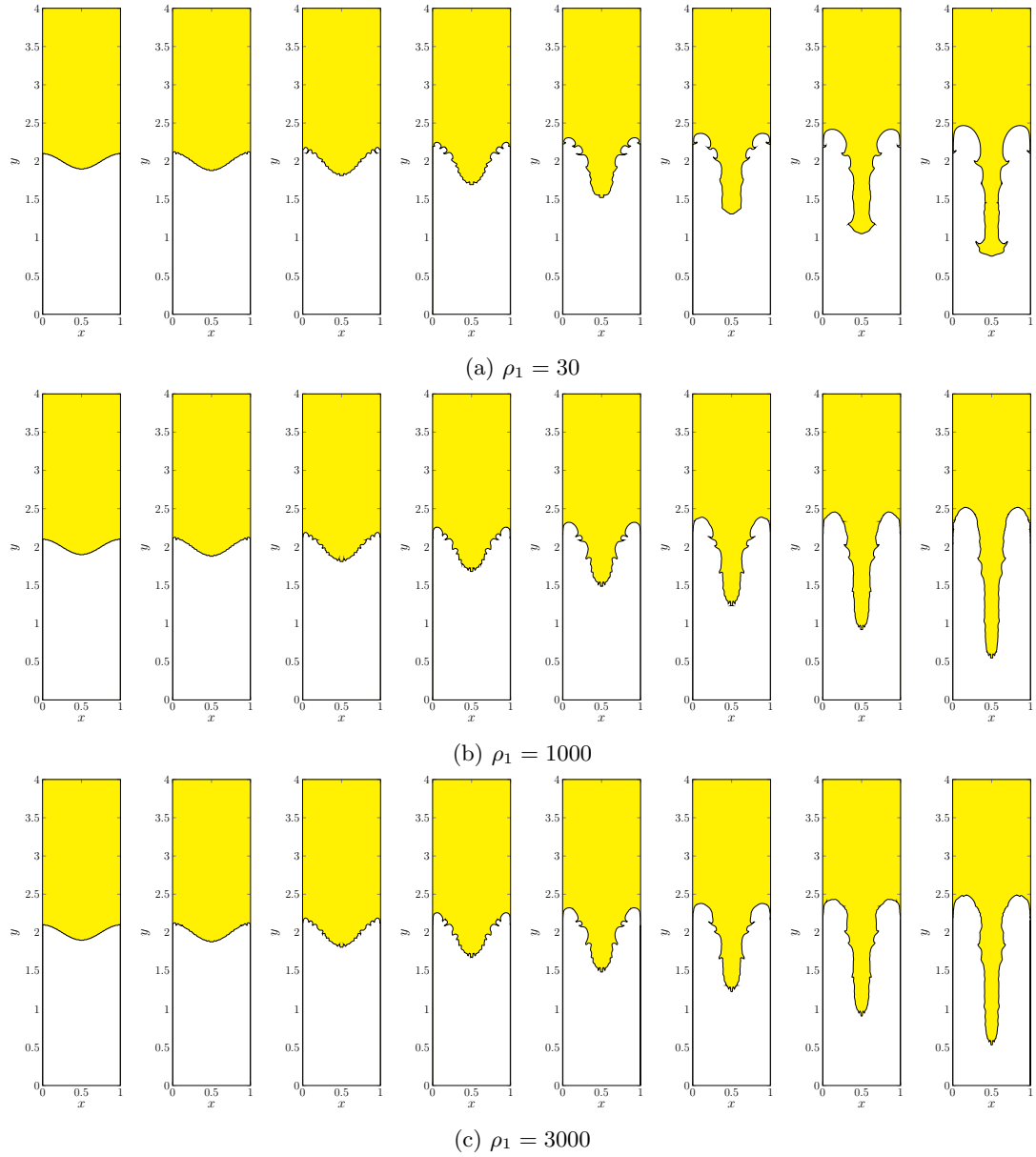


Figure 2.21: Rayleigh-Taylor simulation with density ratio of 30, 1000 and 3000 at times $t = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75$.

by $\sqrt{\frac{a}{|g_y|}}$ so that $T = t / \sqrt{\frac{a}{|g_y|}}$. We do the same for length as well by dividing it by a . We measure our results quantitatively by computing the front of the interface and the highest

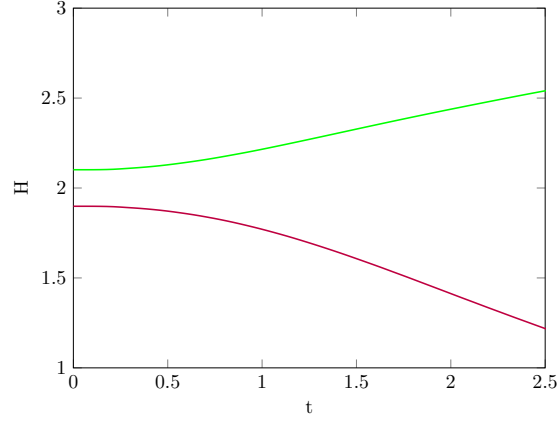


Figure 2.22: Location of the maximum and minimum x of the interface in the Rayleigh-Taylor test with density ratio of 3.

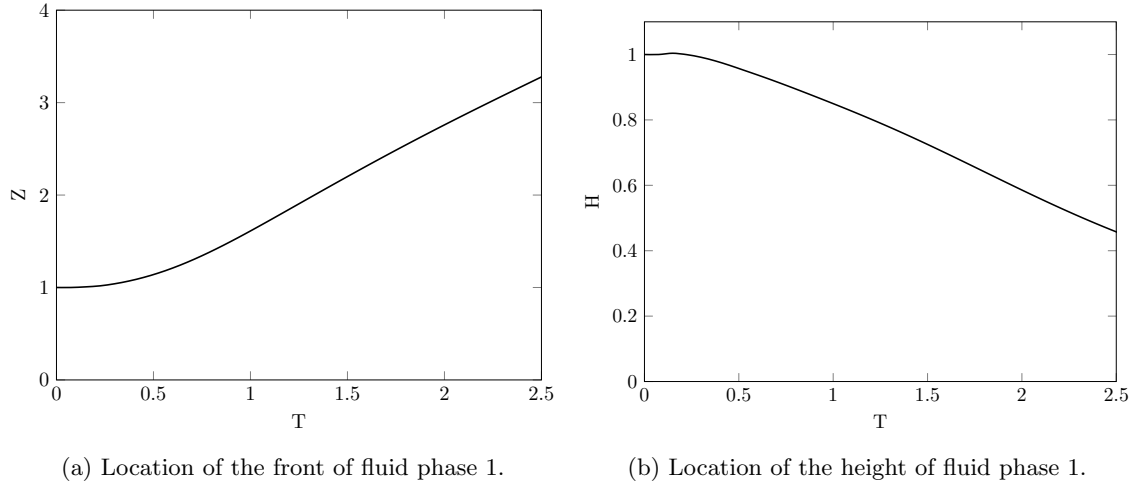


Figure 2.23: Fluid height and front over time in the dam break test.

point of the interface from $T = 0$ to $T = 2.5$ in Figure 2.23. Our results are comparable to those reported by Huang et al. [57] and Martin and Moyce [90].

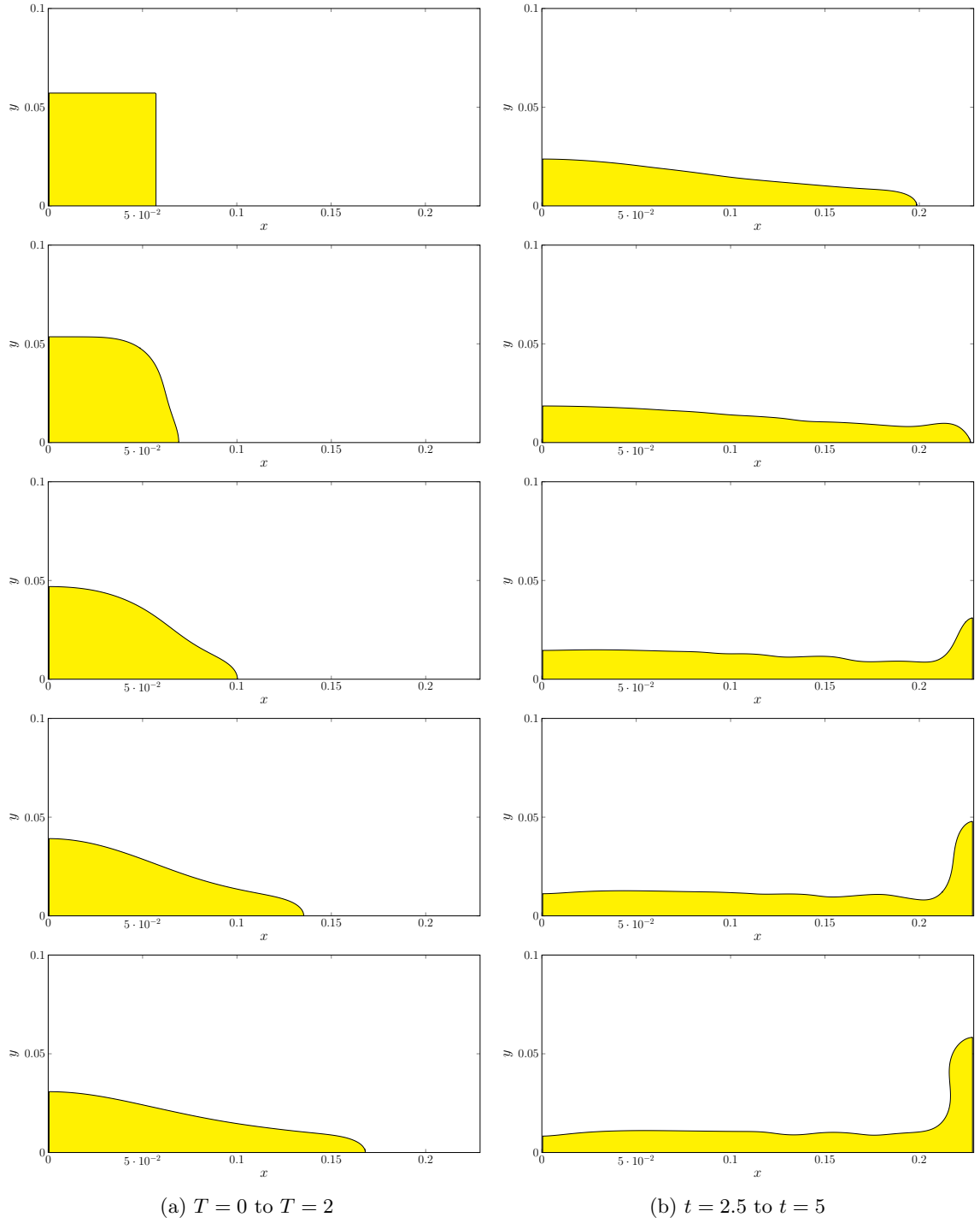


Figure 2.24: The evolution of the dam break test with water and air from $T = 0$ to $T = 5$ in 0.5 increments.

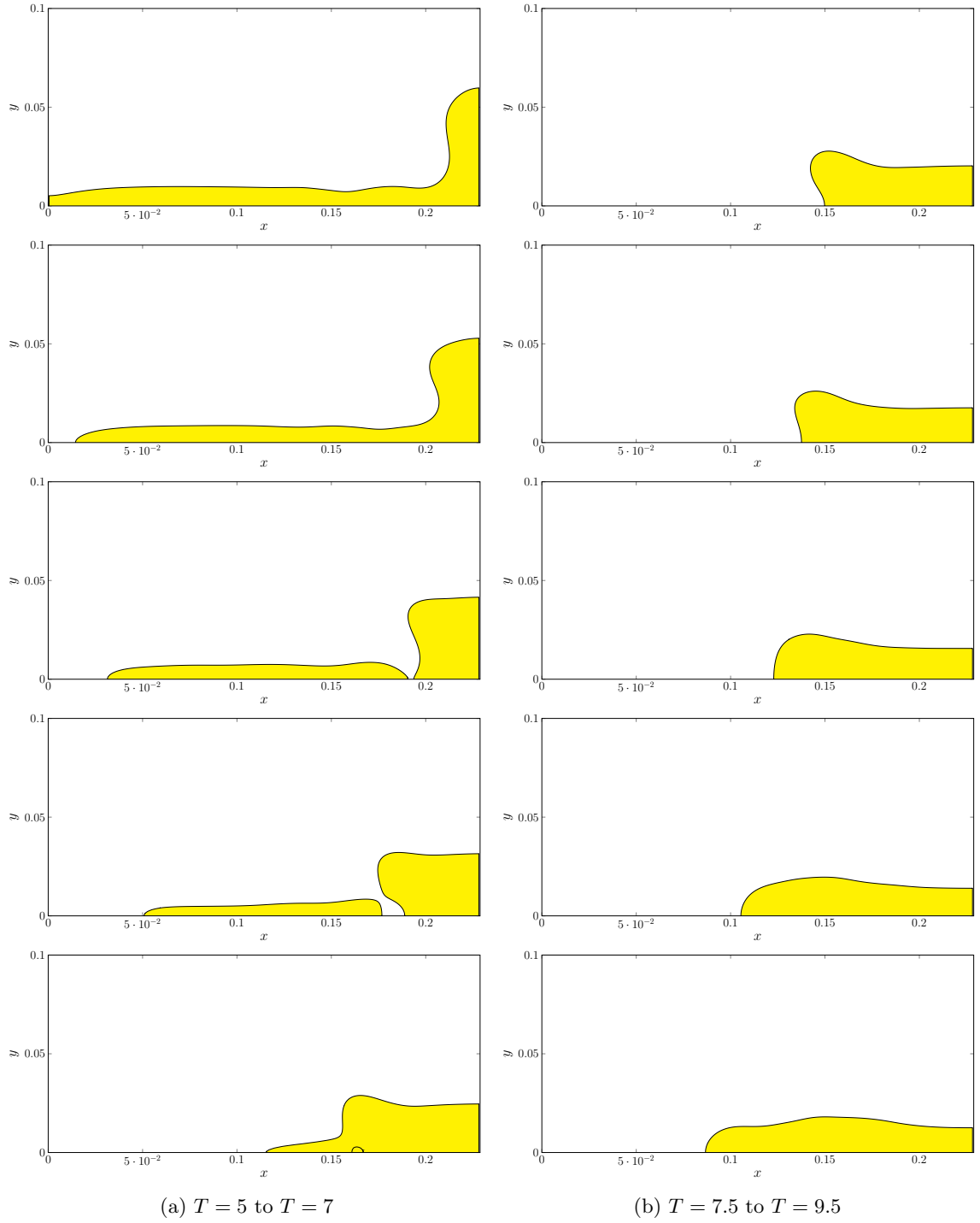


Figure 2.25: The evolution of the dam break test with water and air from $T = 5$ to $T = 9.5$ in 0.5 increments.

2.5.11 Equilibrium Drop

We demonstrate the effects of setting boundary conditions for $\hat{\gamma}^{n+1}$ based on alternate contact angles. We adapt the Neumann boundary condition given in [57] as

$$\mathbf{n} \cdot \nabla \hat{\gamma}^{n+1} = \frac{\sigma}{2\lambda} \cos(\theta) \left(\frac{\pi}{2} \cos \left(\pi \gamma^{AB} - \frac{\pi}{2} \right) \right),$$

which allows for the choice of arbitrary interface-boundary contact angles. We follow the case setup used in [26, 57], where a drop of fluid with $r = 1$ is placed centered at the origin in a $[-3, 3] \times [0, 2]$ domain with cell size $\Delta x = 1/64$. No-slip boundary conditions are enforced at the top and bottom boundaries, and the left and right boundaries are periodic. The fluids have matched density $\rho_0 = \rho_1 = 1$ and viscosity $\mu_0 = \mu_1 = 1$, with surface tension $\sigma = 100$ and interface width $\eta = 0.02$. To match the non-dimensionalized Peclet number defined in [26], we set mobility $M = 1.885 \times 10^{-7}$. No gravity is used in this example. We run the simulation with time step size $\Delta t = 5 \times 10^{-5}$ and end at time $t = 1$.

The initial placement of the drop gives an interface contact angle $\theta = 90^\circ$. By varying the angle set in the bottom boundary condition $\mathbf{n} \cdot \nabla \hat{\gamma}^{n+1}$, we can cause a mismatch between the drop placement and the boundary condition contact angle. This mismatch will cause the drop to change shape as the phase-field evolves so that the interface contact angle will match the boundary condition. In Figure 2.26 we demonstrate the enforcement of boundary contact angles on the bottom boundary with $\theta = 90^\circ, 45^\circ, 135^\circ, 60^\circ, 120^\circ$. Visually, our results are comparable to those presented in [57].

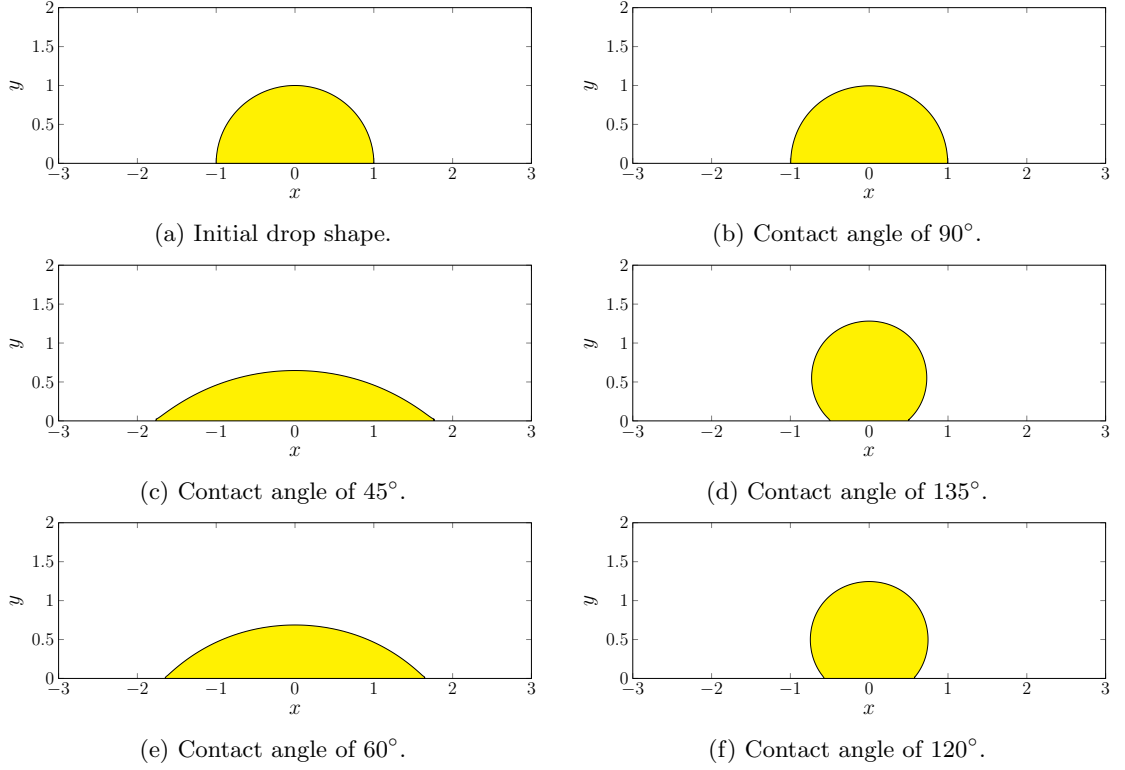


Figure 2.26: Demonstration of the effects of the contact angle θ on a drop at rest.

2.6 Conclusion

In this paper we constructed a novel, simple, and highly stable discretization of the Cahn-Hilliard equation. We coupled this to the Navier-Stokes equations to simulate phase separation in a two-phase fluid flow in two dimensions. We demonstrate that the resulting scheme is second order accurate in γ and velocity and compare it with published results on a number of tests. The scheme is mass-momentum consistent and works with large density ratios.

We also replace existing discretizations of the surface tension force on phase-field variables with one that has finite support around the transition region. The model has a parameter that allows it to transition from a smoothed continuum surface force to a fully

sharp interface formulation, in which case the surface tension applied is equivalent to a sharp interface treatment using jump conditions. We compared our formulation of surface tension against other surface treatments used for the Cahn-Hilliard Navier-Stokes equations and found that our treatment compares favorably.

We have also noticed some limitations in the proposed scheme. As noted in Section 2.5.5, shearing flows with sharp γ transitions can lead to artifacts, though these are observed to diminish with refinement. The consistent advection used for the momentum equations are generally observed to be less numerically robust than a more classical WENO-based advection scheme. We have also observed that the evolution of γ can produce values outside the range $[0, 1]$, which can lead to negative densities if the density ratio is large. We correct this using the gamma redistribution scheme from [56], though we note that this redistribution is effectively global and thus less than ideal. In our implementation, it also comes at the cost of an additional Poisson solve; a more physically plausible solution to this problem is desirable.

2.7 Acknowledgements

This work was supported in part by National Science Foundation award NSF-2006570 as well as University of California award M23PL6076.

Chapter 3

Buoyancy-driven Phase Separation in the Material Point Method

The Material Point Method (MPM) provides a unified framework for simulating multi-material systems but struggles to allow for the separation of objects. Traditionally, multi-material simulations use a single background grid, which allows for automatic interaction between materials. Since particles of different types receive their velocities by interpolating from a common background grid, they are unable to efficiently separate. This is especially problematic when simulating immiscible fluids such as oil and water; once oil particles and water particles become mixed, they are unable to separate. A common solution is to solve different objects or materials on separate background grids, which allows particles of different materials to move with different velocities and separate. Using separate background grids loses the natural interaction between materials, since materials no longer interact through the background grid. In the case of immiscible fluids, phase

separation is driven by thermodynamics and buoyancy. We present a novel method that enables natural buoyancy-driven phase separation in weakly compressible MPM. Our key observation is that compression and pressure do not depend on the type of particle. A local neighborhood of particles is compressed because particles are close together, and they exert pressure because they bump into the particles that are nearby. Correspondingly, we update the deformation gradient from a unified velocity field and use it to compute a unified pressure force. Combined with the use of separate background grids for velocity, this leads to a treatment of immiscible multiphase fluids that naturally separates due to buoyancy forces, even from a fully mixed configuration. We also propose a mixing potential that is capable of driving phase separation caused by thermodynamics even in the absence of gravity. Finally, we propose a novel algorithm for obtaining consistent per-phase level sets for rendering multiphase particle-based fluids.

3.1 Introduction

The Material Point Method (MPM) provides a unified framework for the simulation of a wide variety of solids and fluids, as well as their coupled interactions [120]. However, standard formulations of MPM struggle to achieve phase separation in immiscible multi-material systems [140]. For example, using MPM to simulate a Rayleigh-Taylor instability [104](Fig. 3.6), where a heavier fluid is initially positioned above a lighter one, results in a significant and persistent degree of mixing as the phases attempt to exchange places [64]. This unwanted mixing arises from the chaotic motion of the fluid as the phases interchange, and the underlying shared velocity grid, which prevents buoyancy from un-

mixing the particles effectively. Because particle velocities are interpolated from a shared velocity grid, nearby particles have very similar velocities, which limits relative motion between particles within a cell [98].

Extensive research has addressed limitations in multiphase material point methods [63], though most studies focus on solid-fluid coupling, in which MPM can exhibit non-physical behaviors such as fluid sticking to solids. In engineering, researchers have long employed separate motion fields or grids for each phase to mitigate such issues [87]. More recent work has developed MPM schemes specifically for simulating fluid-saturated soils undergoing large deformations [1, 19]. Similarly, in computer graphics, separate per-phase grids have been adopted to avoid sticking in solid-fluid interactions [127, 36]. However, relatively little research has focused on immiscible fluid-fluid interactions. To address this problem, Li and Liu [77] proposed a multi-species framework using both combined and phase-specific grids. Their approach partially mitigates the problem by introducing per-phase grids, yet significant unwanted mixing persists.

In this work, we identify the cause of persistent mixing in previous multi-fluid MPM approaches. We propose a physically consistent framework utilizing combined and per-phase grids, aligned with the principle of virtual work, through which buoyancy-driven phase separation emerges naturally, addressing the limitations of previous works. Our scheme achieves this through two key choices. First, we maintain strict separation between velocities. Particle velocities are transferred to phase-specific grids. Pressure forces are computed on a combined grid and distributed to phase-specific grids based on volume fractions. Gravitational forces are applied on the phase-specific grids. Velocities are transferred

back from phase-specific grids to the particles of that phase. This allows relative velocities between particles of different phases to persist across time steps and eliminates any Δt -dependent damping of relative motions. Our second key contribution is to observe that the deformation gradient should be updated using the unified velocity grid, obtained as a volume-weighted average of the phase velocity grids. This parallels the way unified pressure forces are distributed to the separate grids in accordance with the principle of virtual work. Intuitively, particles should feel compressed when the particles around them are too close, regardless of phase. Likewise, as long as rising particles are balanced by falling particles, no local compression or expansion should be felt by the fluid. Together, the above choices provide a physically consistent formulation that naturally enables buoyancy-driven phase separation in MPM.

While our principled use of combined and separate grids allows for the emergence of buoyancy-driven separation, separation of immiscible fluids due to other forces is not handled. Physically-based models such as Cahn-Hilliard equations [17] can be used to drive phase separation but require the solution of higher order partial differential equations. We introduce a *mixing potential* that generates separation forces based on geometric overlap, enabling phase separation even when fluid densities are equal or gravity is absent.

Finally, we propose a novel scheme for constructing multiple consistent level sets for rendering. We first construct implicit functions for particles of each phase separately, along with an additional implicit function for all particles together. We then use fast marching and a novel projection algorithm building on [85] to produce consistent per-phase level sets, which we use for rendering. In contrast to Losasso et al. [85], which introduced

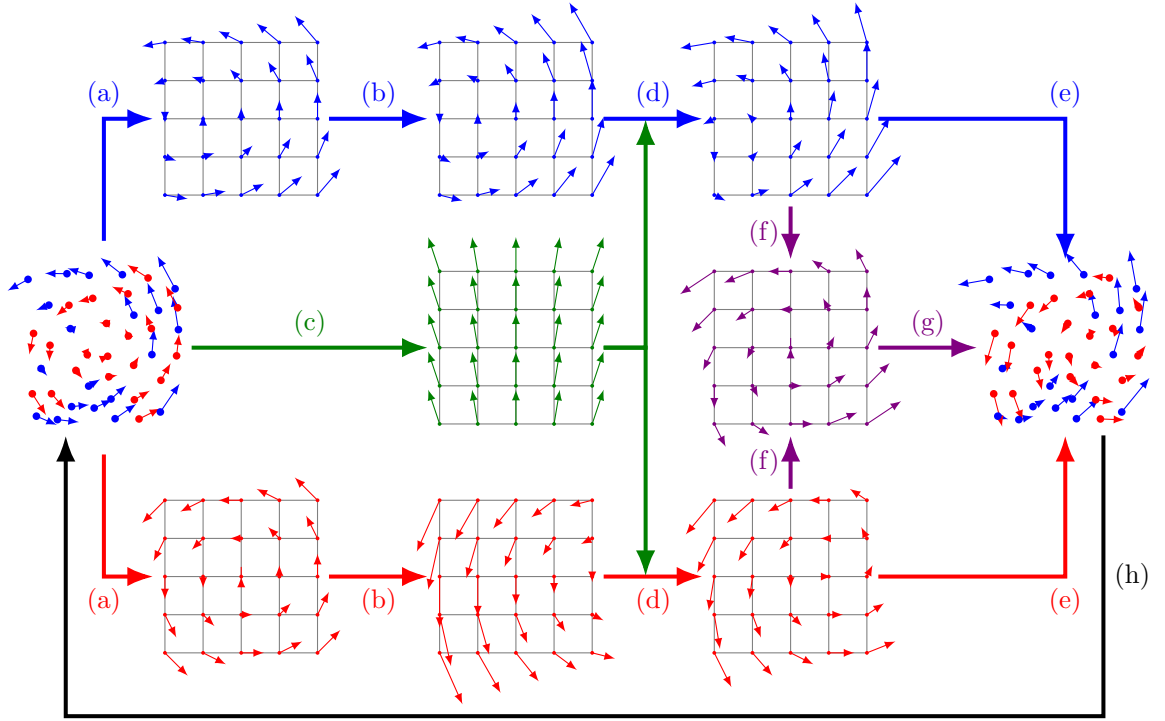


Figure 3.1: An overview of our method. (a) Masses and velocities are transferred from particles to *separate phase* grids. (b) Gravity is applied to the separate phase velocity fields. (c) Constitutive model forces are computed on a combined grid using all particles. (d) The constitutive forces are distributed to the separate phase velocity fields proportional to the phase volume. (e) Grid phase velocities are transferred back to particles. (f) Grid phase velocities are combined into a unified grid velocity. (g) The unified grid velocity is used to update deformation gradients for all particles. (h) Particles are moved and the next time step begins.

a projection scheme to eliminate gaps and overlaps for multiple fluids, our simulations do not have an air phase, and we do not modify the level set representing the total material occupancy.

The contributions of our proposed method include:

- A novel MPM scheme for multiphase fluids which allows fluids to unmix under gravity.
- A mixing potential function to drive phase separation in the absence of gravity.
- A multiphase level set projection approach for creating consistent level sets from particle-based simulations.

While our examples focus on mixtures with two phases, the proposed method naturally generalizes to an arbitrary number of phases.

3.2 Related Work

3.2.1 The material point method

MPM was first proposed as a particle-in-cell method for simulating history-dependent solid mechanics [124, 10]. It has been widely adopted and advanced in computer graphics to simulate snow [120], non-Newtonian fluids [103, 148], spatially varying materials and melting [121], granular materials [23, 73], and thermo-mechanical phenomena [25]. The affine particle-in-cell (APIC) [64] and PolyPIC [34] methods proposed improved particle-grid transfers, conserving angular momentum and reducing dissipation. These single-grid MPM formulations assume one continuum per grid node, thus restricting the relative motion of immiscible materials and preventing unmixing [140, 98]. To address the limitations

ALGORITHM 1: Phase-separating Material Point Method (MPM)
Scheme. Superscripts k indicates mixture data.

Particle to grid transfer

$$m_{k,i}^n = \sum_{p \in P_k} w_{ip}^n m_p ; \quad // \text{ phase specific mass}$$

$$\mathbf{v}_{k,i}^n = (m_{k,i}^n)^{-1} \sum_{p \in P_k} w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n (\mathbf{x}_i - \mathbf{x}_p^n)) ; \quad // \text{ velocity}$$

$$V_{k,i}^n = m_{k,i}^n / \rho_k$$

$$\alpha_{k,i} = V_{k,i}^n / \sum_k V_{k,i}^n ; \quad // \text{ compute volume fraction}$$

Compute forces

$$\bar{\mathbf{f}}_i^e = \sum_p V_p^0 \mathbf{P}_p (\mathbf{F}_p^n)^T \nabla w_{ip}^n ; \quad // \text{ combined elastic force}$$

$$\mathbf{f}_{k,i}^\Phi = - \sum_{m \in K} \sum_{p \in P_m} f'(n_{m,p}) \frac{\partial n_{m,p}}{\partial \mathbf{x}_{k,i}} ; \quad // \text{ phase separation}$$

Apply forces

$$\mathbf{f}_{k,i}^{n+1} = m_{k,i}^n \mathbf{g} + \alpha_k \bar{\mathbf{f}}_i^e + \mathbf{f}_{k,i}^\Phi ; \quad // \text{ split elastic force with } \alpha_k$$

$$\tilde{\mathbf{v}}_{k,i}^{n+1} = \mathbf{v}_{k,i}^n + \Delta t (m_{k,i}^n)^{-1} \mathbf{f}_{k,i}^{n+1}$$

Transfer grid to particle using phase-specific velocity

$$\mathbf{v}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1}, p \in P_k ; \quad // \text{ Update particle velocity}$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} ; \quad // \text{ Update particle position}$$

$$\mathbf{C}_p^{n+1} = \frac{1}{\xi} \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T, p \in P_k$$

Update deformation gradient

$$\bar{\mathbf{v}}_i^{n+1} = \sum_k \alpha_{k,i} \tilde{\mathbf{v}}_{k,i}^{n+1} ; \quad // \text{ combine phase-specific velocities}$$

$$\nabla \bar{\mathbf{v}}_p^{n+1} = \sum_i \bar{\mathbf{v}}_i^{n+1} (\nabla w_{ip}^n)^T$$

$$\mathbf{F}_p^{n+1} = (\mathbf{I} + \Delta t \nabla \bar{\mathbf{v}}_p^{n+1}) \mathbf{F}_p^n$$

of the MPM framework in allowing for relative motion, researchers have employed explicitly separate grid velocity fields, coupled through interaction forces. Early methods were developed to model frictional contact following material fracture [95, 53, 96].

3.2.2 Multiphase MPM formulations

Multiphase MPM works in graphics have primarily focused on solid-fluid coupling and sediment-laden flows, using momentum exchange terms to couple separate grid phase velocities [127, 36, 151]. Engineering applications similarly employ separate grid fields for soil-water interactions [87, 1, 19]. MPM solid-fluid coupling and immiscible fluid-fluid coupling have also been achieved with a single grid by introducing discontinuities or interface forces [55, 140], but these approaches require explicit interface detection and modeling. IQ-MPM [31] combines separate velocity integration with a unified pressure solve to achieve solid-fluid coupling. Nilles and Müller [98] adds an artificial buoyancy force to facilitate unmixing due to density differences.

Recently Li and Liu [77] introduced a mixture model relying on a combination of shared and per-phase grids to promote unmixing of immiscible fluids. However, they use a combined grid for velocities in the particle to grid step, dampening existing relative motion, and a per-phase deformation gradient update that fails to accurately capture combined strain. These choices limit the ability of the method to recover from mixed configurations. Consider the situation illustrated in Fig. 3.2(a), where fluid is initially in three regions. The top region consists of a light phase, and the bottom region consists of a heavy phase. The middle portion is a mixture of both phases, as might form during positional exchange. The

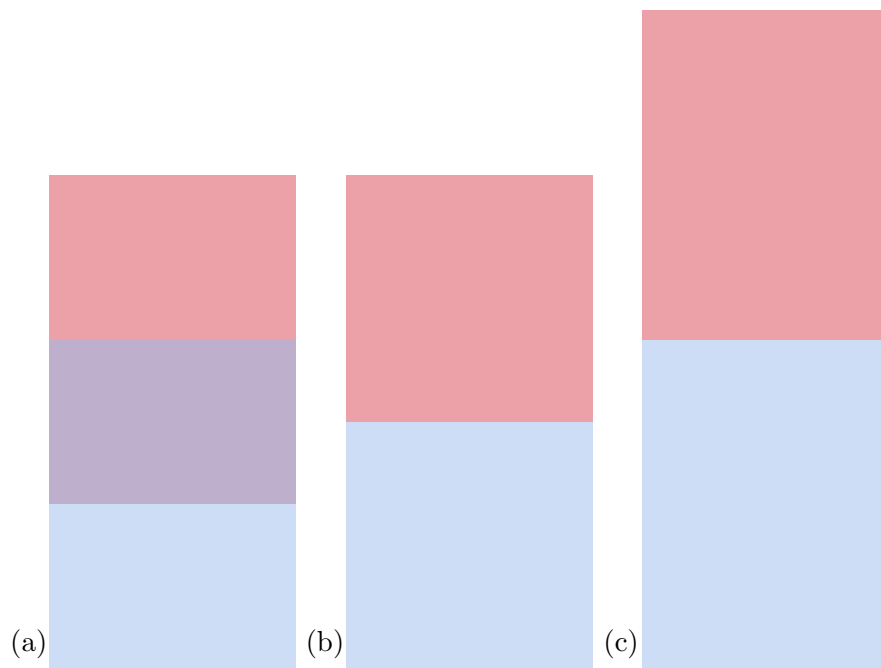


Figure 3.2: (a) Initial condition for a partially mixed domain with lighter fluid on top and denser fluid on the bottom with a mixed region in the middle. (b) Expected phase separation: lighter particles rise and denser fluids sink in the mixed regions, with little movement elsewhere. (c) Behavior observed by [77]: this method preserves the volume of each phase, causing all light particles to rise, which results in an overall volume increase.

physically correct behavior would be for the light particles in the mixed region to rise and the heavy particles to sink, leading to the configuration in Fig. 3.2(b). Compared with the initial configuration, the light particles now occupy a smaller volume of the overall fluid domain. The heavier particles also occupy a smaller volume of the overall fluid domain. Modeling deformation of each phase in isolation, both phases appear to be compressed in the region that was previously mixed. In the method of Li and Liu [77], these regions would exert pressure forces to restore their volume, erroneously leading to the configuration shown in Fig. 3.2(c). This volume expansion artifact is observed in Fig. 5 of Li and Liu [77], where the unmixed result is approximately twice the depth of the mixed result. This also limits the ability to unmix when there is not sufficient room for expansion.

3.2.3 Multi-fluid and phase separation models outside MPM

Multi-fluid simulation has seen significant development in the computer graphics community [106]. Mixture models treat multiphase flows as a single continuum with volume fractions and drift velocities for the individual phases [92, 97]. In particle methods, mixture models have been used with SPH for both miscible and immiscible flows [105, 141] and with position-based dynamics (PBD) for a variety of interacting materials [88]. In the context of SPH methods [58], multi-fluid schemes have been developed [94] and extended to support large density ratios [116], high viscosities [101], and realistic mixing [84]. Kinetic lattice Boltzmann methods have also been proposed for the simulation of turbulent multi-fluid flows [78].

Phase-field models evolve an order parameter tracking the phase volume fractions and represent fluid-fluid interfaces diffusely. The Allen-Cahn or conservative Cahn-Hilliard

equations [17] are typically coupled to the Navier-Stokes equations to evolve the dynamics [8, 41]. While these approaches can effectively model phase separation and surface tension, they typically incur significant overhead in solving the high order evolution equation for the order parameter. Nevertheless they have seen successful application with SPH and PBD [142, 143, 144] as well as MPM [130]. Our geometric mixing potential provides a lightweight alternative via a separation force when buoyancy is insufficient.

3.2.4 Multi-fluid level set reconstruction

Many works have investigated the consistent representation of multi-material systems using multiple implicit functions, with the primary problem the elimination of invalid gaps or overlaps. In solid simulation, Gascuel [38] eliminated overlaps of implicitly represented deformable objects, and works such as [134, 86, 99] address the issue in solving multi-material structural design optimization problems. In multiphase flows, methods for post-evolution correction have been proposed including simple projection [91, 85], Voronoi reconstruction [109], and projection onto an $(n - 1)$ -dimensional manifold [115]. Some approaches utilize evolution strategies that inherently maintain consistency [107, 5, 131]. Unlike prior projection methods which treat all phases symmetrically, we wish to fix our total material level set and correct only the individual phase boundaries.

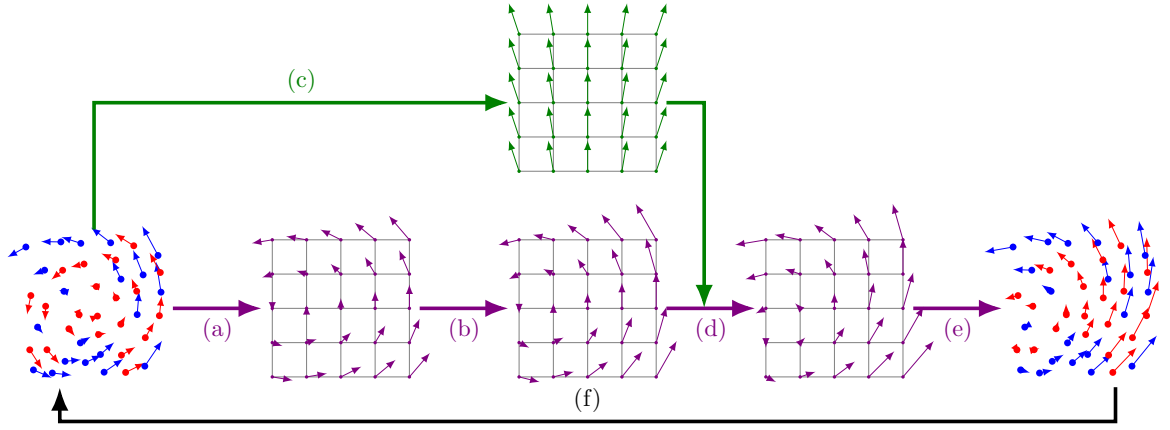


Figure 3.3: The traditional MPM algorithm (a) transfers mass and velocity from particles to grid, (b) applies gravity, (c) computes constitutive forces, (d) and applies them to velocities. Finally (e) particles are updated and (f) moved.

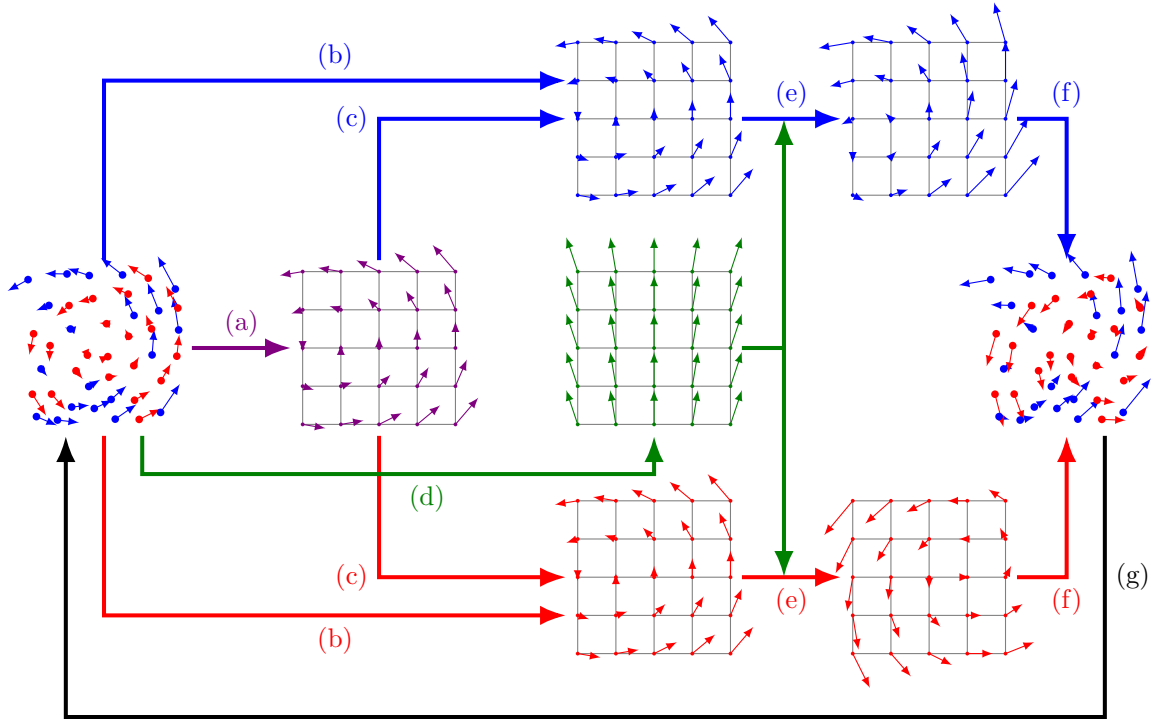


Figure 3.4: An overview of the method of [77]. (a) Velocities are transferred from particles to *unified* grids, then (c) copied to per-phase grids. Masses are transferred to (a) unified and (b) per-phase grids. (d) Constitutive model forces are computed on a combined grid and (e) distributed to the separate phase velocity grids with gravity. (f) Grid phase velocities are transferred back to particles, deformation gradient is updated, and (g) particles are moved.

3.3 Method

Fig. 3.1 shows an overview of our method. We build our approach on top of Affine Particle-in-Cell (APIC) MPM [64]. In MPM, the continuum body is discretized into a set of particles. At the beginning of each time step, all state is stored on particles. Each particle $p \in P_k$ belongs to a phase $k \in K$, where K is the set of phases and P_k is the set of particles of that phase. All particles store mass m_p , velocity $\mathbf{v}_p^n$, velocity gradient $\mathbf{C}_p^n$, and initial volume V_p^0 . Phase k has density ρ_k , so that $m_p = \rho_k V_p^0$ for $p \in P_k$. We use $\mathbf{v}_{k,i}^n$ to refer to the velocity stored on grid k at index i . Quantities on merged grids are denoted by a bar $\overline{\mathbf{v}}_i^n$.

3.3.1 Particle to Grid Transfer

We first transfer mass and momentum to the phase-specific grids, obtaining velocity by dividing off the mass.

$$m_{k,i}^n = \sum_{p \in P_k} w_{ip}^n m_p, \quad \mathbf{v}_{k,i}^n = \frac{1}{m_{k,i}^n} \sum_{p \in P_k} w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n (\mathbf{x}_i - \mathbf{x}_p^n)),$$

where w_{ip}^n are the interpolation weights. We use quadratic splines for our transfers [117]. Unlike Li and Liu [77], we never transfer velocities to a combined grid; the per-phase velocities remain fully separated throughout our algorithm.

3.3.2 Grid Forces

We apply gravity to each phase separately. We compute constitutive forces on a combined background grid, then distribute them to per-phase grids using volume weighting.

Elastic forces come from strain energy stored during deformation. In MPM, each particle carries a deformation gradient $\mathbf{F}_p$ that is evolved using the local velocity gradient. Particles feel pressure because they occupy space, not because they have mass, so volume weighting is physical. Pressure forces on a volume of fluid are proportional to the volume of fluid present. Thus, we distribute the pressure forces $\bar{\mathbf{f}}_i^e$ to each phase k using volume weights $\alpha_{k,i}$. With this, our per-phase forces and velocities are

$$\mathbf{f}_{k,i}^{n+1} = m_{k,i}^n \mathbf{g} + \alpha_k \bar{\mathbf{f}}_i^e + \mathbf{f}_{k,i}^\Phi, \quad \tilde{\mathbf{v}}_{k,i}^{n+1} = \mathbf{v}_{k,i}^n + \Delta t \frac{1}{m_{k,i}^n} \mathbf{f}_{k,i}^{n+1},$$

where $\alpha_{k,i} = V_{k,i}^n / \sum_k V_{k,i}^n$, $V_{k,i}^n = m_{k,i}^n / \rho_k$ are phase volumes, $\mathbf{g}$ is gravity, and $\bar{\mathbf{f}}_i^e$ are combined elastic forces. We also include a phase separation force $\mathbf{f}_{k,i}^\Phi$ to allow separation driven by thermodynamics rather than gravity (see Section 3.3.5).

3.3.3 Emergence of buoyancy

The force application above sufficient to achieve buoyancy-driven phase separation. Within a given region, phases receive a gravitational force proportional to their mass and a pressure force proportional to their volume. The dense phase will receive relatively more gravity, pushing it down. The lighter phase will receive relatively more pressure, causing it to rise. Note that if mass weighting were used for pressures, gravity and pressure would balance, and no phase separation would occur.

3.3.4 Grid to Particle Transfer

Next, we transfer velocities from separate per-phase grid velocities to particles using APIC transfers.

$$\mathbf{v}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1}, \quad \mathbf{C}_p^{n+1} = \frac{1}{\xi} \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T, \quad p \in P_k.$$

where $\xi = \frac{\Delta x}{4}$ for quadratic splines. Since particle velocities of different phases are interpolated from different grids, they are able to move with different velocities and efficiently separate.

We note that deformation gradients and elastic forces are complimentary. In the case of fluids, the deformation gradient tracks changes in the local particle spacing, and the pressure exerts forces to counter them. If we observe a small region of an oil-water mixture, we see patches of oil rising and patches of water sinking until eventually the phases have completely separated. Since the fluid is not locally compressing, this unmixing does not generate pressure forces. This suggests that the deformation gradient should be combined, mirroring the pressure treatment. We begin by computing a merged velocity

$$\bar{\mathbf{v}}_i^{n+1} = \sum_k \alpha_k \tilde{\mathbf{v}}_{k,i}^{n+1}.$$

We use the same weights to interpolate velocities as we used to distribute forces, following the principle of virtual work. We use these merged velocities for our deformation gradient

update.

$$\nabla \bar{\mathbf{v}}_p^{n+1} = \sum_i \bar{\mathbf{v}}_i^{n+1} (\nabla w_{ip}^n)^T, \quad \mathbf{F}_p^{n+1} = (\mathbf{I} + \Delta t \nabla \bar{\mathbf{v}}_p^{n+1}) \mathbf{F}_p^n.$$

Using a combined volume-weighted average of phase velocities allows for preservation of the total fluid volume, while still allowing each phase to change its volume. This makes phase separation possible even in fully filled domains. In contrast to our approach, Li and Liu [77] updates the deformation gradient $\mathbf{F}_p^n$ on particles using $\mathbf{C}_p^{n+1}$, computed from phase-separate grid velocities. As discussed in Section 3.2, this prevents effective phase separation.

Finally, we complete the time step by moving the particles

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}.$$

3.3.5 Phase separation force

The careful treatment of pressure and gravity on separate grids suffices to reproduce buoyancy-driven phase separation between fluids of different densities. Phase separation can also be driven by thermodynamics rather than buoyancy. In this section, we introduce a force that is able to drive phase separation in the absence of density differences or gravity. For simplicity we assume there are only two phases, which we can index by k and $\ell = 1 - k$. Although we limit ourselves to two phases in this chapter, the separation force can be naturally extended to more phases by simply applying pairwise.

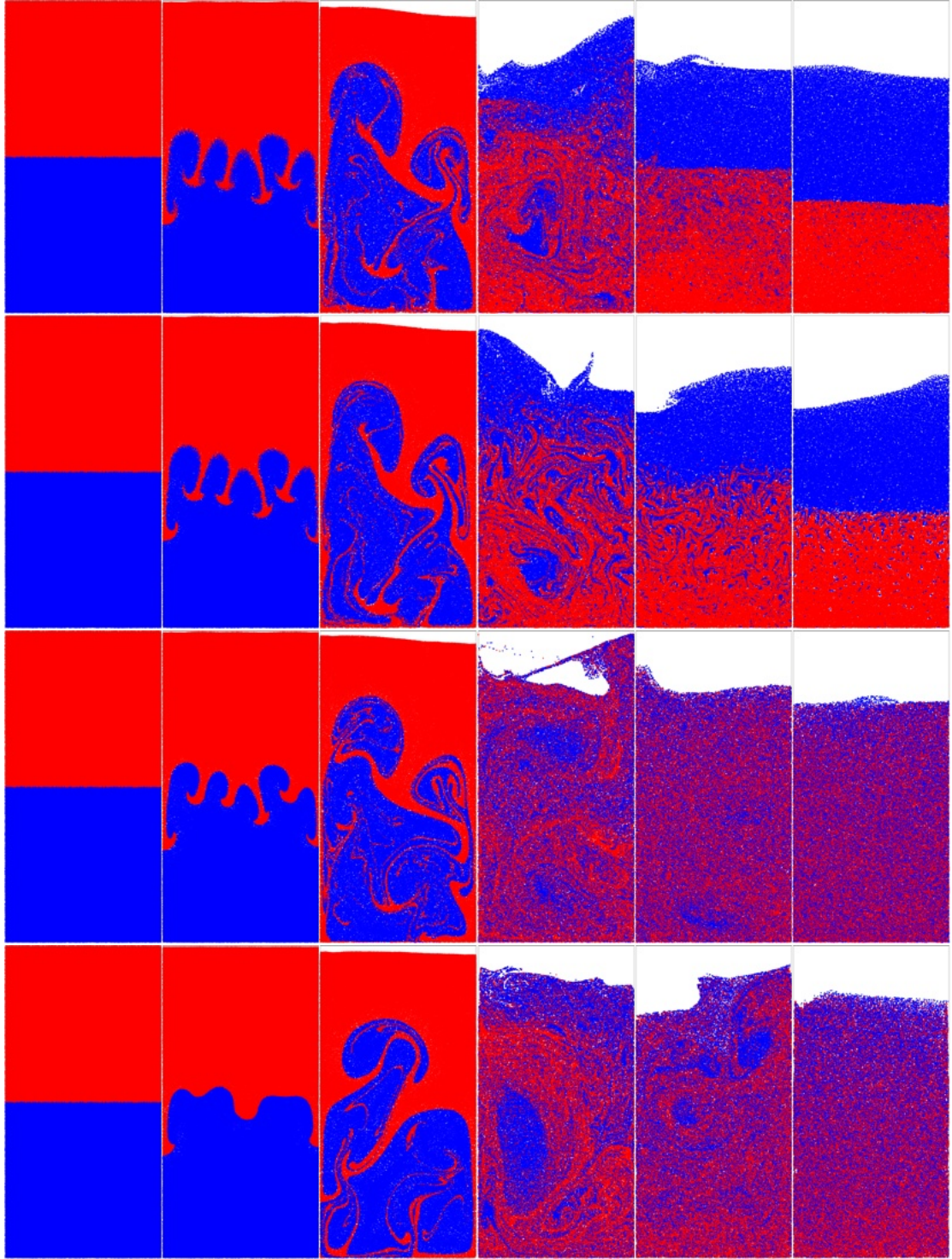


Figure 3.5: Rayleigh–Taylor instability. Rows, top to bottom: our method, our method with phase separation force, standard MPM, and Li and Liu [77]. Left to right: $t = 0, 1, 2, 8, 16, 32s$. Our method successfully unmixes the phases, while the others do not. Our phase separation force promotes small scale clumping while maintaining larger scale dynamics.

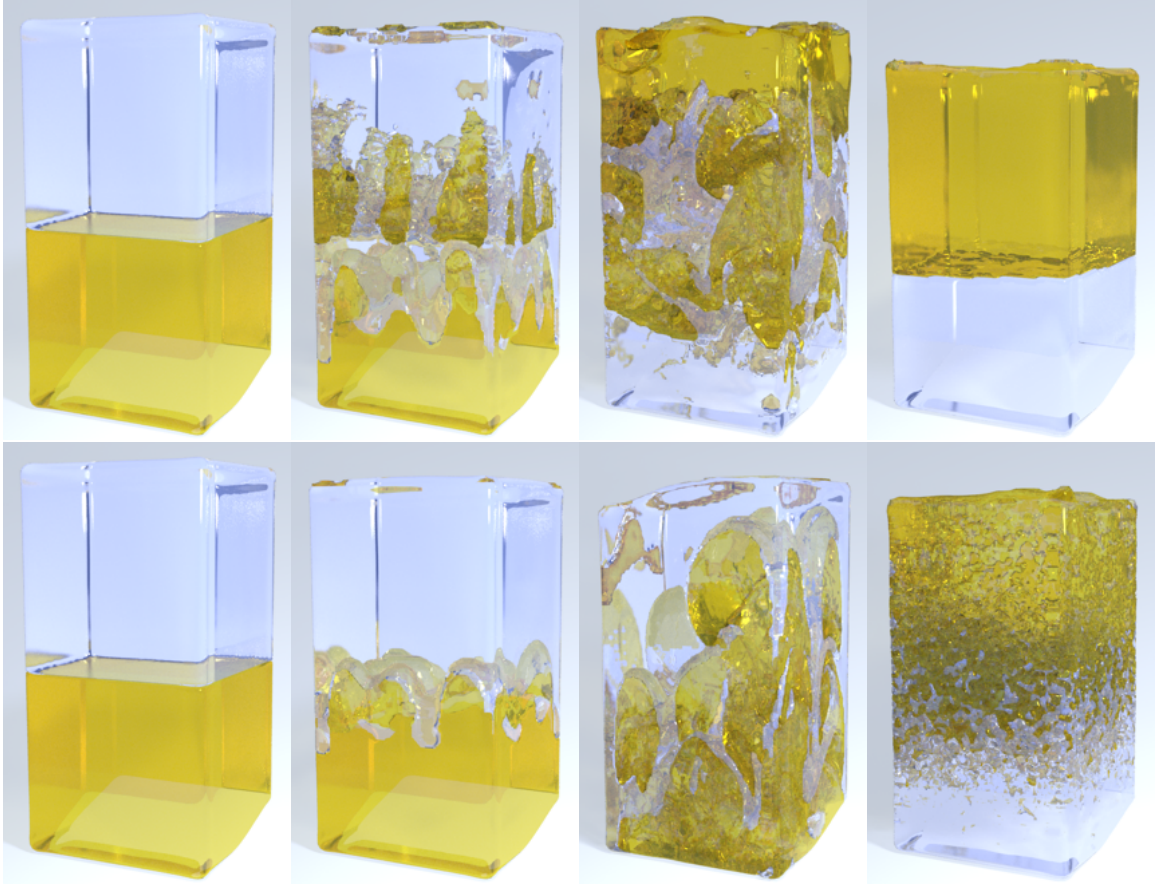


Figure 3.6: Rayleigh–Taylor instability. Initial state followed by our method (middle) and Li and Liu [77] (right) at $t = 0, 1, 2, 16s$. Our method sharply separates the two phases, alleviating the residual mixing evident with previous approach.

For certain combinations of fluids, being in a mixed configuration is energetically less favorable. We model this with a potential function that penalizes mixing. We first define a grid field that assigns each point in space to one of the phases.

$$n_{k,i} = m_i^k m_i^\ell \left(\sum_{p \in P_k} w_{ip} - \sum_{p \in P_\ell} w_{ip} \right) \quad (3.1)$$

If the local neighborhood is dominated by particles of phase k , then $n_{k,i} > 0$. If phase ℓ dominates, then $n_{k,i} < 0$. Note that $n_{k,i} = -n_{\ell,i}$. Next, we evaluate this at the particle locations

$$n_{k,p} = \sum_i w_{ip} n_{k,i}.$$

If $p \in P_k$ and $n_{k,p} > 0$, then the particle p is within its own phase, and there should be no force. If $n_{k,p} < 0$, then particle p is surrounded by particles of the other phase and is effectively mixed. The magnitude provides a measure of how mixed the particle is. We use this to define our mixing potential

$$\Phi = \sum_{k \in K} \sum_{p \in P_k} f(n_{k,p}), \quad f(x) = \begin{cases} cx^2, & x < 0, \\ 0, & x \geq 0. \end{cases} \quad (3.2)$$

The coefficient c determines the strength of the unmixing force. The unmixing force is obtained by differentiating this potential with respect to per-phase grid locations (as though the separate phase grids were Lagrangian meshes that are moving and deforming) while holding $n_{k,i}$ fixed (as the grids move, the particles move with them). The grid forces $\mathbf{f}_{k,i}^\Phi$

are

$$\mathbf{f}_{k,i}^\Phi = - \sum_{m \in K} \sum_{p \in P_m} f'(n_{m,p}) \frac{\partial n_{m,p}}{\partial \mathbf{x}_{k,i}}$$

$$\frac{\partial n_{k,p}}{\partial \mathbf{x}_{k,i}} = w_{ip} \sum_j \nabla w_{jp} n_{k,j}, \quad \frac{\partial n_{k,p}}{\partial \mathbf{x}_{\ell,i}} = -\nabla w_{ip} n_{k,i}.$$

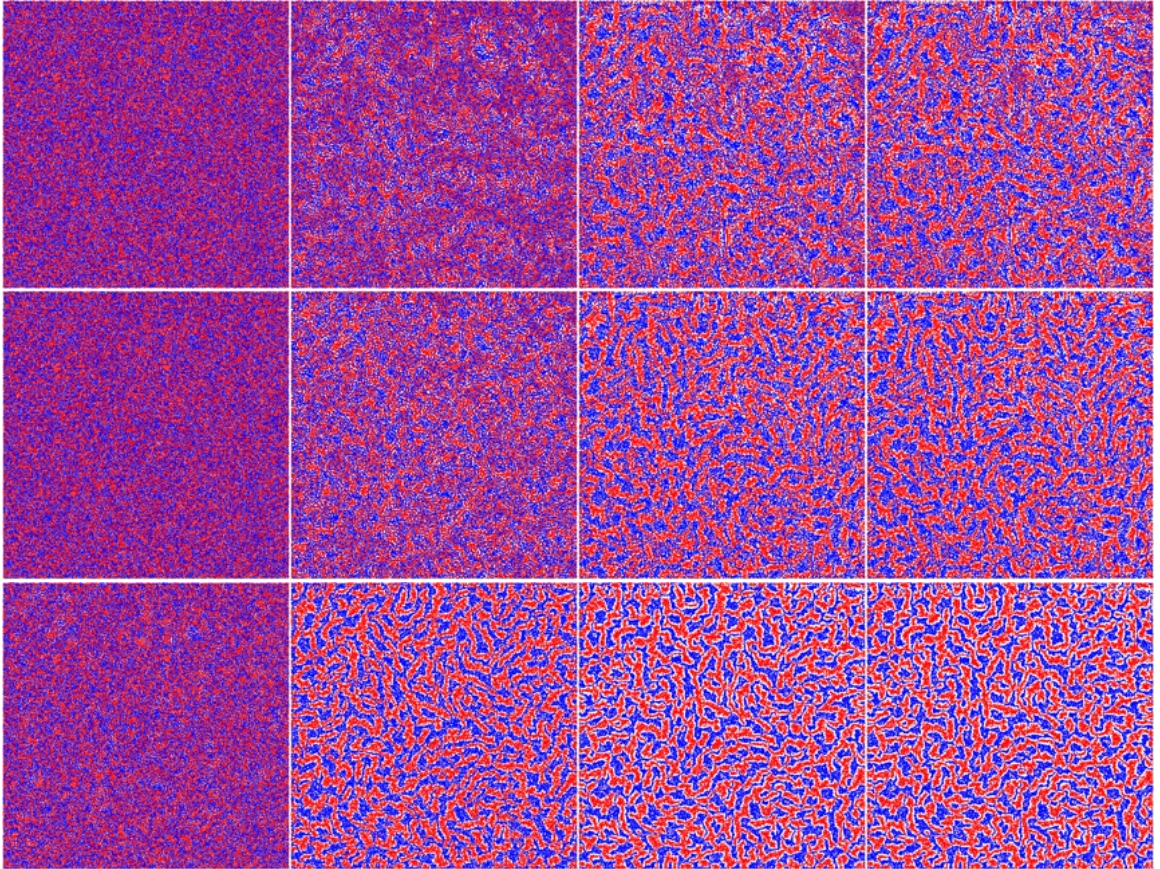


Figure 3.7: Fully mixed domain *without* gravity with density ratio 10, including our phase separation force in 2D at $t = 0, 2, 8, 10$ s. row 1: $c = 50$. row 2 : $c = 100$ and row 3: $c = 1000$ The phases clump into small subregions but do not separate vertically. Larger values for c result in faster separation.

3.3.6 Projection Method

ALGORITHM 2: Level Set Projection Method

Input: Level set values: single-phase ϕ_k and ϕ_ℓ , combined fluid ϕ .
Output: Non-overlapping level set values: ϕ'_k and ϕ'_ℓ with $\phi'_k \leq \phi'_\ell$.
Require: $\phi_k \leq \phi_\ell$

```

if  $\phi \geq 0$  then
    |  $\phi'_k = \phi, \quad \phi'_\ell = \max(\phi_\ell, \phi)$                                 // Case 1
else
    |  $s = (\phi_k - \phi_\ell)/2$ 
    | if  $\phi \leq s$  then
    | |  $\phi'_k = s, \quad \phi'_\ell = -s$                                     // Case 2
    | else if  $\phi_k \geq \phi$  then
    | |  $\phi'_\ell = \phi_\ell + (\phi - \phi_k), \quad \phi'_k = \phi$           // Case 4
    | else
    | |  $\phi'_k = \phi, \quad \phi'_\ell = \max(\phi_\ell, -\phi)$               // Case 3
    | end
end

```

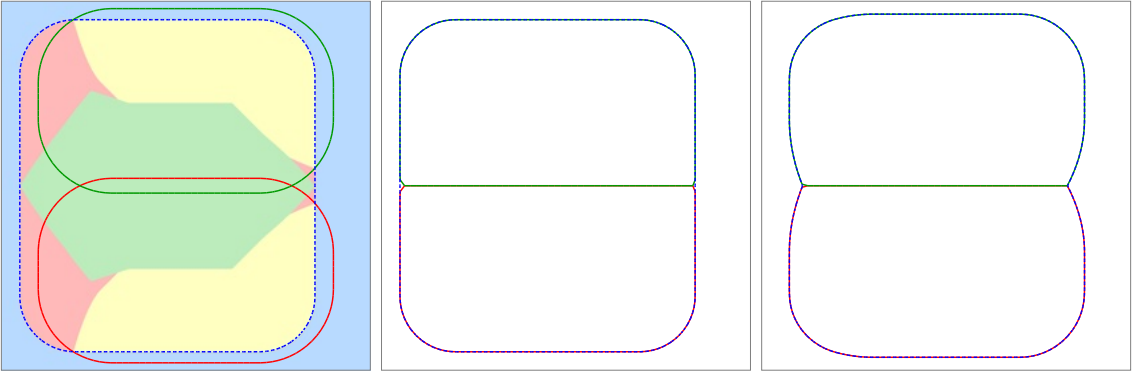


Figure 3.8: Level set projection. Per-phase level sets are shown with solid contours and the combined level set with a dashed contour. (Left) Initial overlapping state of the multiple level sets. Background colors represent cases in our level set projection method: Case 1 (blue), Case 2 (green), Case 3 (yellow), and Case 4 (red). Comparison of our level set projection method (middle) with a the method of [85] (right). Leaving the combined level set fixed during projection avoids artifacts near triple junctions.

For rendering, it is convenient to have a signed distance field representing the fluid region. In the case of multiphase fluids, we require per-phase signed distance fields

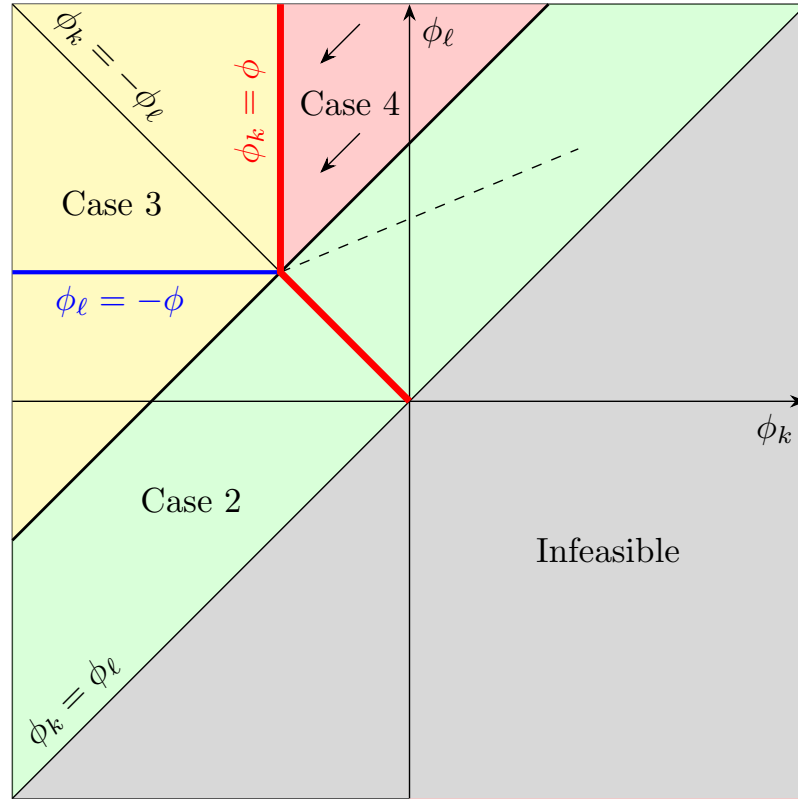


Figure 3.9: Projection map used to enforce consistency between the ordered level sets (ϕ_k, ϕ_ℓ) with $\phi_k \leq \phi_\ell$ and $\phi < 0$. The red polyline indicates the feasible set of consistent outputs (ϕ'_k, ϕ'_ℓ) . Given an input point (ϕ_k, ϕ_ℓ) , the projection algorithm computes a nearby feasible point (ϕ'_k, ϕ'_ℓ) . Selecting the nearest feasible point would produce a discontinuity at the dashed line. Instead, input space is divided into cases (Cases 2–4, indicated by the colored regions; Case 1 corresponds to $\phi \geq 0$ and is not shown). Cases 2 and 3 map to the closest feasible point. Points in Case 4 are moved to the feasible line along the direction of the arrows.

ϕ_k and ϕ_ℓ . We also construct a combined ϕ using all of the particles. In order to avoid gaps and overlaps, these signed distance fields need to be consistent near the interface. Let $\phi_0 \leq \phi_1 \leq \phi_2$ be the three level sets ϕ_k , ϕ_ℓ , and $-\phi$ in sorted order. The consistency criterion is then that $\phi_0 = -\phi_1$. Note that this is the same criterion as in Losasso et al. [85], which we use as the starting point for our own projection. Unlike Losasso et al. [85], we do not allow our third level set $-\phi$ to be changed during the projection. This prevents artifacts at the interface of the two phases near the domain boundary. In particular, it ensures that the union of the two per-phase level sets matches the level set we obtained from the full set of particles.

We construct phase-specific signed distance fields ϕ_k and ϕ_ℓ from particle sets P_k and P_ℓ , respectively, using the negative of Equation 3.1. We construct ϕ using all particles using the method of Zhao et al. [150, 149] and then we extend it to the full domain using fast marching. We cannot use Zhao et al. [150]’s method to reconstruct per-phase level sets when the two fluids jointly fill the domain since all of the reconstructions will coincide.

We propose a projection method similar to Losasso et al. [85] to make our level sets consistent (see Algorithm 2). For simplicity, we assume that $\phi_k \leq \phi_\ell$ and we will produce as output ϕ'_k and ϕ'_ℓ such that $\phi'_k \leq \phi'_\ell$. If $\phi \geq 0$, then we are outside the domain, so we must have $\phi = \phi'_k \leq \phi'_\ell$. We can enforce this by setting $\phi'_k = \phi$ and $\phi'_\ell = \max(\phi_\ell, \phi)$, which changes the inputs as little as possible while satisfying the constraints.

Otherwise, $\phi < 0$, and we are inside the domain. This leads to three cases, which are illustrated in Fig. 3.8 and Fig. 3.9 . The red line is the set of configurations that satisfy the consistency requirements. We must project each point (ϕ_k, ϕ_ℓ) to a point (ϕ'_k, ϕ'_ℓ) on

this line. Projecting all points to the closest point on the feasible line is not continuous at the dashed line. Instead, we only handle the cases 2 and 3 by projecting to the closest point on the red line. Note that for case 2, this gives us the update rule from Losasso et al. [85]. For case 4, we instead move points to the feasible region along the same direction as we did in case 2. Each of these cases leads to a very simple update rule, which we show in Algorithm 2. The resulting algorithm is idempotent and continuous everywhere.

3.4 Results

Table 3.1: Timing comparison of methods in 2D and 3D. Entries marked “-” indicate that the method was not evaluated. Initial elasticity values in dambreak 1 differ from other tests to match the results reported by Li and Liu [77].

Test Name	Particles	Res.	# Threads	E	Standard MPM	Ours	Li and Liu [77]
Stable State	121328	128×256	1	10^5	138.58 s	158.20 s	503.22 s
Partially Mixed	121152	128×256	1	10^5	141.81 s	157.48 s	300.05 s
Rayleigh-Taylor Instability	121315	128×256	1	10^5	136.70 s	153.73 s	372.57 s
Rayleigh-Taylor + Phase Separation	121315	128×256	1	10^5	-	110.29 s	-
Dam Break (Version 1)	53988	128×128	1	3×10^3	0.76 s	0.87 s	1.36 s
Dam Break (Version 2)	60696	256×256	1	10^5	139.89 s	160.37 s	654.49 s
Fully Mixed + Gravity	60624	128×128	1	10^5	-	95.16 s	-
Fully Mixed + Phase Separation	60624	128×128	1	10^5	-	106.65 s	-
Fully Mixed + Gravity + Phase Separation	60624	128×128	1	10^5	-	113.26 s	-
Rayleigh-Taylor Instability	425985	$64 \times 128 \times 64$	8	10^4	375.43 s	427.48 s	599.74 s
Dam Break	551100	$64 \times 64 \times 64$	8	10^5	60.86 s	43.51 s	43.73 s
Fully Mixed + Gravity	212972	$64 \times 64 \times 64$	8	10^4	-	240.44 s	-
Fully Mixed + Phase Separation	212972	$64 \times 64 \times 64$	8	10^4	-	284.19 s	-
Fully Mixed + Gravity + Phase Separation	212972	$64 \times 64 \times 64$	8	10^4	-	273.26 s	-

We compare our method with standard MPM and the method of Li and Liu [77]. Runtimes are provided in Table 3.1. We use the fixed corotated constitutive model [119]. We compute the Lamé coefficient λ corresponding to the specified stiffness E and Poisson’s ratio $\nu = 0.3$, and we set $\mu = 0$ to disable shear forces, leaving only a pressure force. We use quadratic B-splines [117, 64] for transfers. Following [125], we reset $\mathbf{F}_p^n \leftarrow \det(\mathbf{F}_p^n)^{1/d} \mathbf{I}$ with $d = 2, 3$ each time step; this avoids accumulation of deviatoric components in the

deformation gradient that do not affect forces. We follow the CFL criteria and separating boundary conditions from [125]. The simulations in this chapter were rendered using SideFX Houdini.

3.4.1 Stable test

We begin by evaluating our method on a stable 2D setup with lighter fluid on top and denser fluid on bottom. Results are included in the supplementary video. Overall, the methods behave similarly. The fluids should remain unmixed and stationary, though some parasitic currents are observed in all schemes, including the original MPM scheme. These are caused by residual buoyancy resulting from the nonuniform grid mass m_i^n . We note that in our method, a thin gap of approximately two grid cells forms at the interface between the phases. When simulating Li and Liu [77]’s method, we used $CFL = 0.5$ for stability.

3.4.2 Partially mixed test

Next, we introduce a mixed region between the lighter phase on top and the denser phase on the bottom as shown in Fig. 3.11. The mixed region should spontaneously separate. Our method successfully separates the two fluids, whereas both standard MPM and the method of Li and Liu [77] fail to unmix. The fully occupied domain leaves no free space for the lighter phase to rise, inhibiting the unmixing process of Li and Liu [77], as it would require compressing the phases.

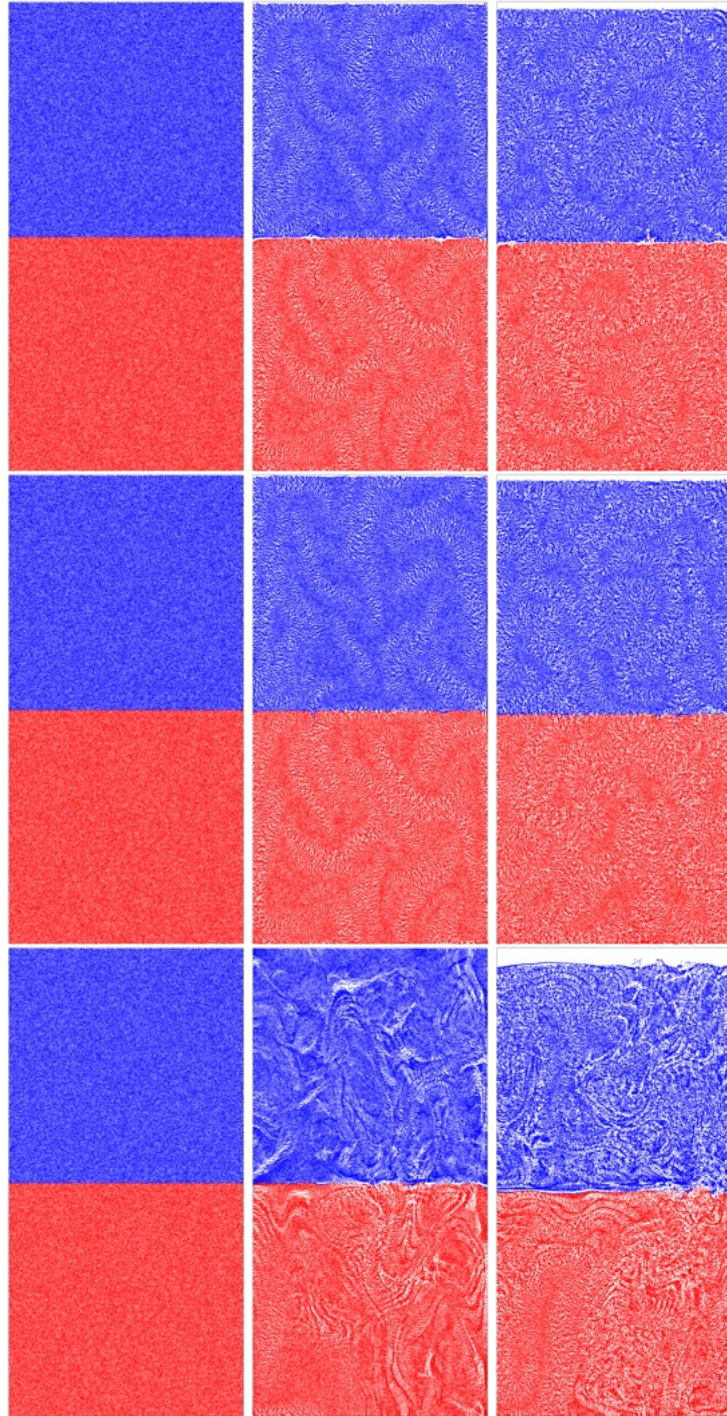


Figure 3.10: Lighter fluid on top of a denser fluid using our method (top row), standard MPM (middle row), and Li and Liu [77]’s method (bottom row) at $t = 0, 5, 10s$ from left to right.

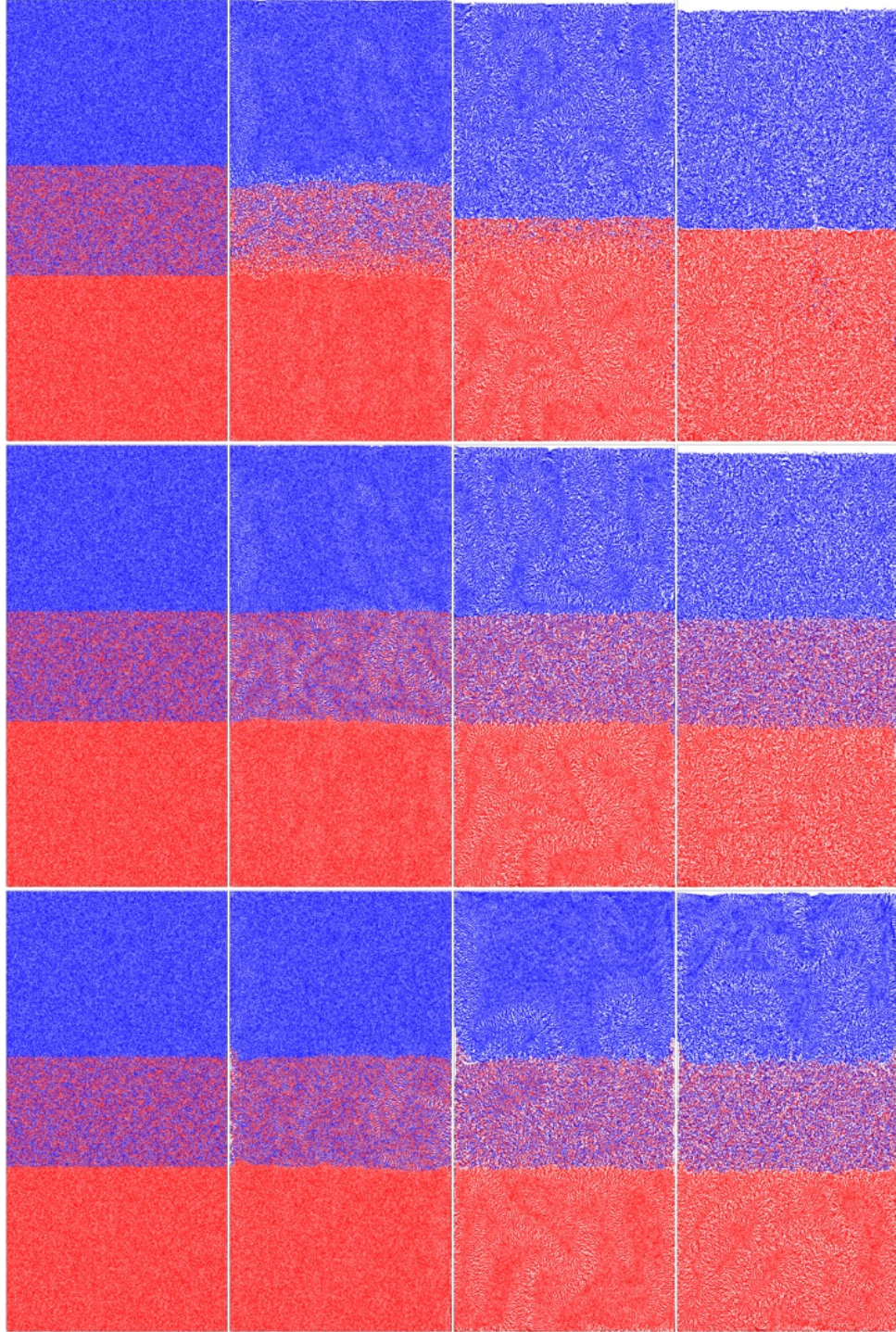


Figure 3.11: Lighter fluid on top of a denser fluid with a mixed region in the middle. Our method (top row), standard MPM (middle row), and Li and Liu [77]’s method (bottom row) at $t = 0, 2, 4, 16s$ from left to right. Our method successfully unmixes the phases, while the other two do not.

3.4.3 Rayleigh-Taylor instability

To investigate the performance under spontaneous mixing, we perform a Rayleigh-Taylor test, initialized with the heavy phase on top of the lighter phase. As the fluids change places, particles from the two phases mix. Phase separation is evident in our method by $t = 16s$ (Fig. 3.6 and Fig. 3.5), whereas minimal separation occurs with standard MPM. With Li and Liu [77], a mixed gradient forms, but phase separation is not achieved. We simulate Li and Liu [77]’s method with $CFL = 0.5$.

Fig. 3.12 shows a histogram of particle distribution for each phase at $t = 41.67s$ of the 2D test. Standard MPM and Li and Liu [77]’s method show gradual gradients, with the heavy phase more concentrated at the bottom and the light phase distributed almost uniformly. By contrast, our method shows a sharp phase separation. The 3D test results are similar to the 2D case and are shown in Fig. 3.6. This test demonstrates that turbulent flows can induce particle mixing, even from an initially fully unmixed state. The ability to recover from mixed configurations is thus critical.

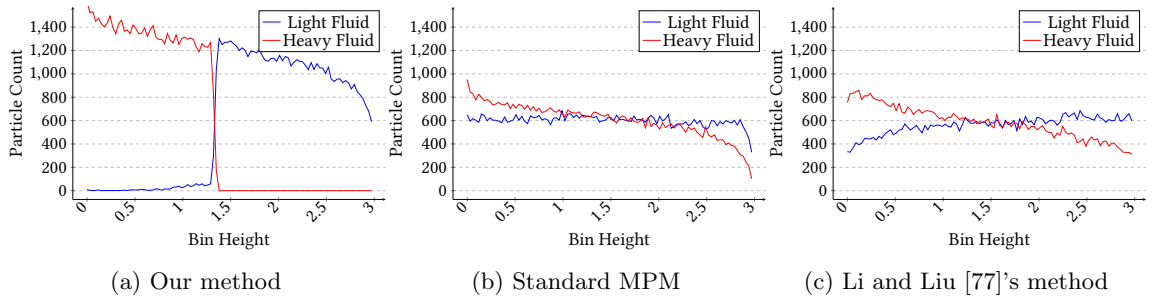


Figure 3.12: Histogram of bin height vs. phase particle count for 100 bins from $y = 0$ to $y = 3$ of the Rayleigh-Taylor instability test at time $t = 32s$. Our method (3.12a) exhibits phase separation, while mixing is apparent with standard MPM (3.12b) and Li and Liu [77]’s method (3.12c).

3.4.4 Dam break test

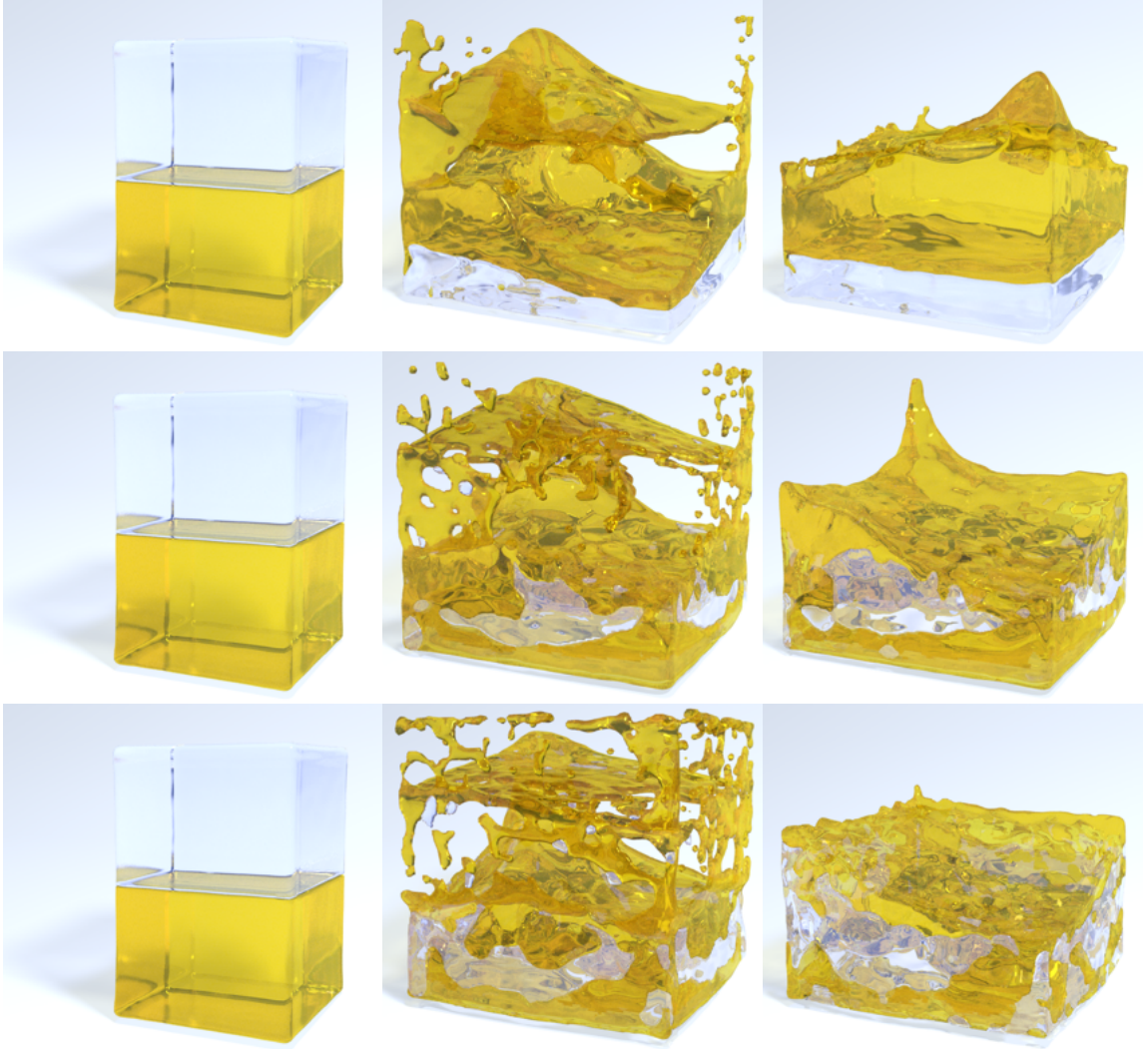


Figure 3.13: 3D dam-break test with denser fluid initially above lighter fluid. Row 1: our method at $t = 0, 4, 8$ s. Row 2: standard MPM at $t = 0, 4, 8$ s. Row 3: Li and Liu [77] at $t = 0, 4, 8$ s. Our method robustly separates the phases whereas the other two do not.

We consider two versions of the classic dam break test. The first version, from Li and Liu [77], positions the denser fluid on top of the lighter fluid with a gap and a shift between the two blocks (Fig. 3.15). Our implementation of their method reproduces results

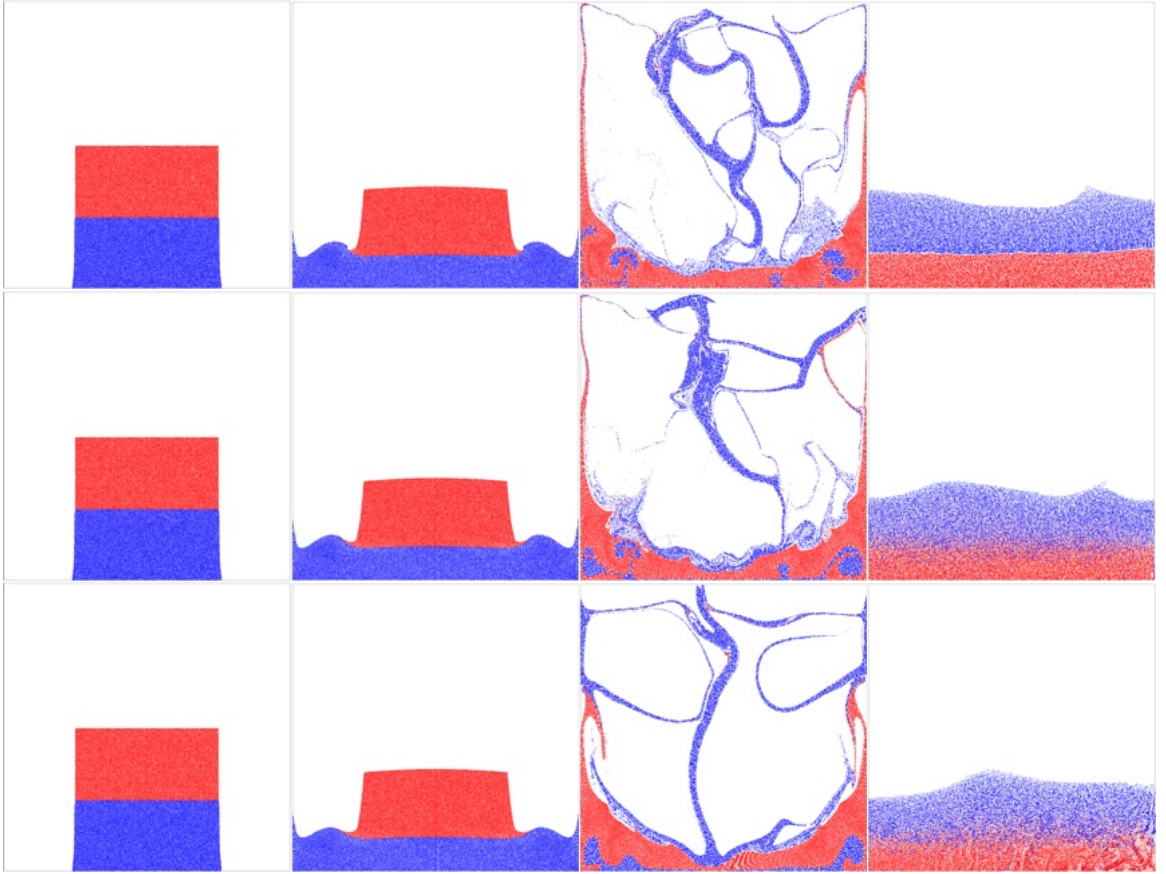


Figure 3.14: Dam break test with denser fluid on top and lighter fluid on the bottom using our method (top row), the standard MPM (middle row) and Li and Liu [77]’s method (bottom row) at $t = 0, 0.25, 1, 16s$ from left to right. Our method gives sharp separation of phases, whereas evident mixing persists using the other two approaches.

comparable to theirs. Both our method and Li and Liu [77]’s approach successfully separate the two phases, whereas standard MPM does not. We used $CFL = 0.3$ for Li and Liu [77]’s method.

We also examine a more typical dam break test consisting of a single block with separate phases. Our method achieves complete phase separation in both 2D and 3D (Fig. 3.14 and Fig. 3.13). In contrast, both standard MPM and the method of Li and Liu [77] exhibit visible mixing. This test demonstrates the robustness of our approach in achieving phase separation.

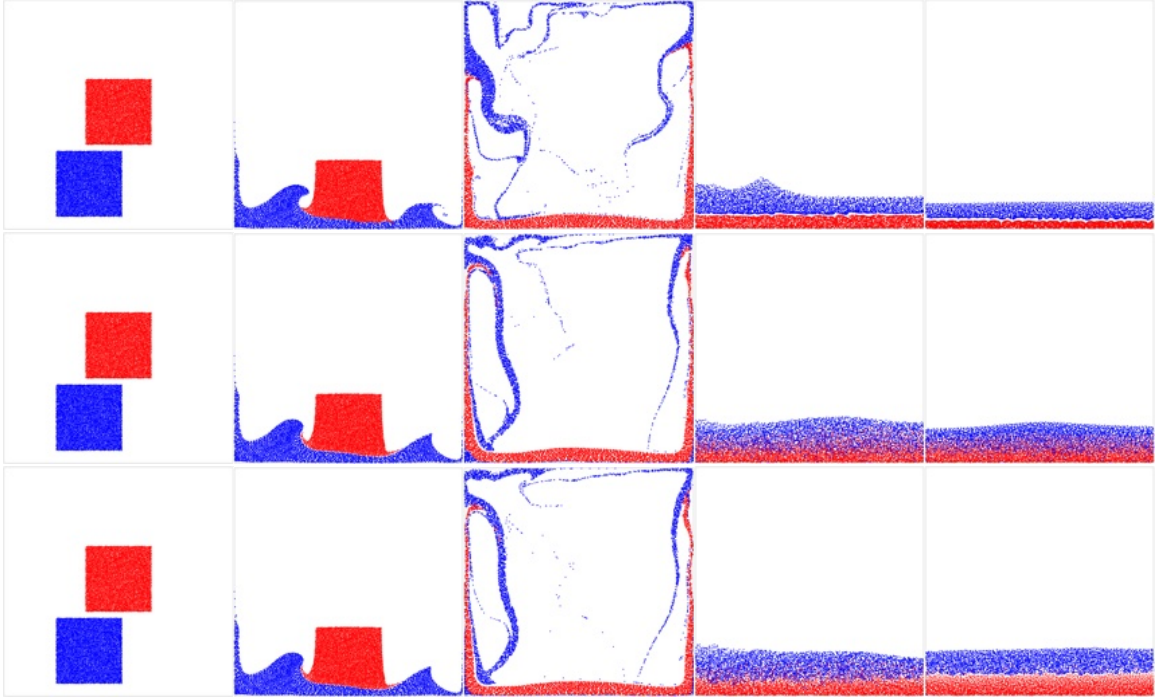


Figure 3.15: We reproduce the example from Li and Liu [77] with the phases initially offset, giving comparable results with their method. Our method (top row) shows good separation. Standard MPM (middle row) maintains significant mixing at the interface. Li and Liu [77]’s method (bottom row) exhibits some residual mixing with ongoing slow separation. Left to right: $t = 0, 0.5, 1, 16, 62.5s$.

3.4.5 Fully mixed test with phase separation force

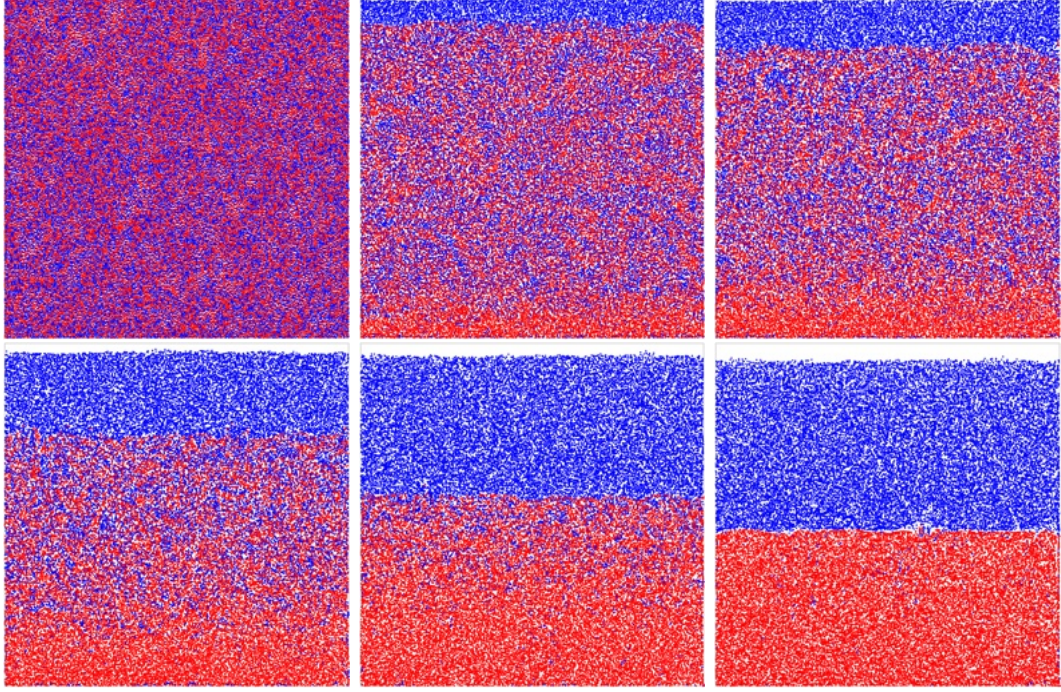


Figure 3.16: Fully mixed domain under gravity with density ratio 10 using our method at $t = 0, 2, 4, 8, 16, 32s$ left to right, top to bottom.

We examine the proposed phase separation force in 2D and 3D. We begin with the two phases uniformly mixed throughout the domain. In the first case, we run the simulation with gravity, and our method successfully separates the fluids as in previous tests (see Fig. 3.17 for 3D results and the video for 2D results). Next, we turn off gravity and add the proposed phase separation force (Fig. 3.7 and Fig. 3.18). The phase separation force causes the phases to clump into globules by phase, but in the absence of gravity the phases do not separate vertically. A larger value of c (Eq. (3.2)) accelerates the separation

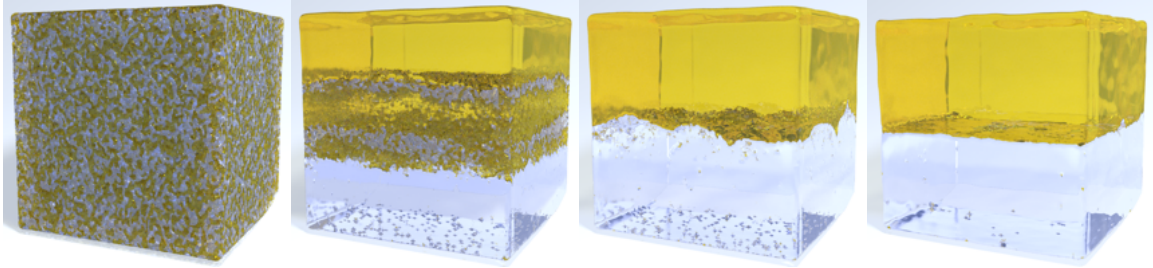


Figure 3.17: 3D Fully mixed domain test under gravity with density ratio 10, using our method at $t = 0, 0.5, 1, 8s$. Our method fully unmixes the phases.

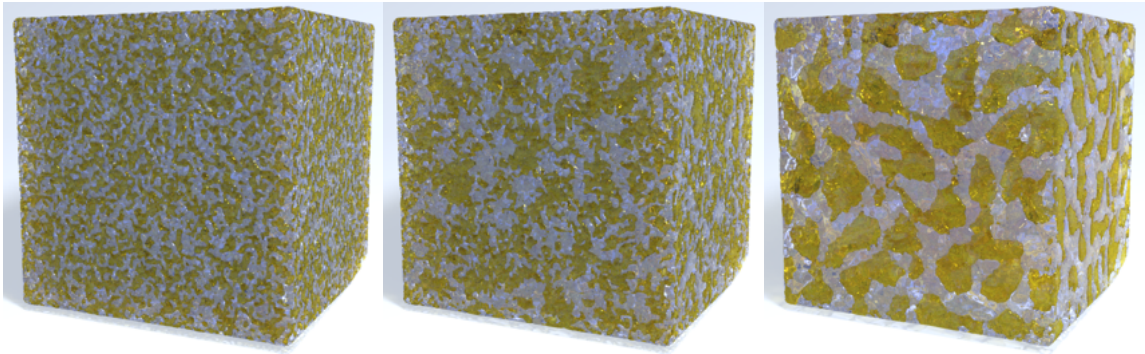


Figure 3.18: Fully mixed domain *without* gravity with density ratio 10, including our phase separation force, using our method at $t = 0, 4, 16s$ in 3D. The phases clump into small subregions but do not separate vertically.

process. Comparison of results for different values of c is shown in Figure 3.7 and in the supplemental video.

3.4.6 Rayleigh-Taylor instability with phase separation force

To evaluate the proposed phase separation force, we repeat the Rayleigh-Taylor test using our scheme and the proposed phase separation force (Section 3.3.5). The results are shown in Fig. 3.5, second row, and closely resemble those in Fig. 3.5, first row, where the phase separation force is not applied. However, the particles of each phase tend to cluster together more tightly due to the additional phase separation force. This indicates that the phase separation force promotes small scale phase separation while preserving the larger scale dynamics of the simulation.

3.4.7 Discussion

Simulating immiscible multiphase fluids is problematic for standard MPM due to the tendency of particles to become mixed during the simulation. Our method effectively and efficiently enables buoyancy-driven phase separation in immiscible fluids. This alleviates limitations of previous approaches, such as [77], where phase separation, when it occurs at all, is very slow.

The overhead of our method is quite modest; our method is only about 5% slower than standard MPM (see Table 3.1). We observed no stability problems with our own approach; we were able to run with the regular MPM time step restrictions without any problems. We found that our implementation of Li and Liu [77] suffered from reduced stability compared with standard MPM, which required us to run it with reduced time step

sizes; this caused it to be about 2.25 times slower than regular MPM. If it could be run with the same time step sizes, its performance would be similar to ours.

The phase separation force does not significantly alter the physical behavior of simulations when run with a modest value of c , since the time scale for buoyancy-driven phase separation is comparable. Increasing c causes phases to first separate in place, which can actually slow overall unmixing. Buoyancy-driven separation occurs when particles of one phase pass through regions of the other phase; the phase separation force counteracts this. The separation force behavior is qualitatively more physical on smaller length scales, but the buoyancy-driven mechanism is more physical at larger length scales.

3.4.8 Generalization to $N > 2$ phases

Though we do not directly demonstrate it here, the method described could be applied to $N > 2$ phases. The scheme depicted in Fig. 3.1 straightforwardly extends to more phases by transferring to/from additional phase grids in steps (a), (d), and (f) using volume fractions computed based on all phases. The separation force can be extended to $N > 2$ by applying it to pairs of phases, as is often done for surface tension. The projection procedure generalizes to $N > 2$ in the same manner as [85] does: the smallest two level sets become ϕ_k and ϕ_l .

As for scaling, our implementation uses dense grids, so the scaling is $O(N^2)$ for the pairwise forces and $O(N)$ elsewhere. An implementation with sparse grids would scale better in practice for the pairwise forces. $O(N^2)$ scaling would only be observed in cases where large portions of the domain contain particles of many types. Where overlap is minimal (normal usage), performance should not depend appreciably on N .

3.5 Conclusion and Limitations

We introduce two methods for phase separation. The first is a new scheme for the material point method, where phases separate through a buoyancy-driven mechanism. The second is a mixing potential that applies forces to regions of fluid containing mixed particles. The effectiveness of these methods in separating fluids is demonstrated in Section 3.4. The phase separation force is heuristic and has a tunable parameter. Moreover, using large values of c results in more noticeable gaps and necessitates a smaller time step.

In both the buoyancy-driven scheme and the mixing-potential approach, we observe the formation of a gap between phases. This gap becomes less noticeable at higher resolutions, as its width is on the order of a grid cell. We also observe volume loss in our method after a long simulation time, the larger the interface between the two phases, the more significant the volume loss. This is due to the fact that the deformation gradient is updated using a merged velocity field, which does not accurately capture the local deformation of each phase near the interface. Future work could explore more sophisticated methods for updating the deformation gradient that better account for phase-specific deformations, potentially reducing volume loss over time.

Chapter 4

An implicit time integration scheme for the multi-phase Material Point Method

The standard Material Point Method (MPM) struggles to accurately simulate multi-phase flows and buoyancy-driven phase separation. To address this, recent explicit MPM frameworks have successfully introduced phase-specific background grids to enable natural phase separation. In this chapter, we propose an implicit time integration scheme built directly upon this explicit multi-phase formulation to improve simulation stability. However, shifting to an implicit approach inherently increases volume loss compared to the explicit baseline. To mitigate this issue, we introduce a novel particle distancing potential and force term that penalizes over-dense particle regions by dynamically regularizing inter-particle spacing relative to the background grid cell size. Finally, we investigate the

underlying numerical mechanisms driving volume loss in both implicit and explicit schemes and propose targeted solutions to further preserve volume.

4.1 introduction

The Material Point Method (MPM) is a powerful method for simulating a wide range of materials, including fluids, solids, and granular materials [120, 121, 103, 65]. However the standard MPM struggles when it comes to simulating multiphase immiscible fluids[140]. This is due to the way MPM handles forces and velocities on its background grid [77, 6]. In standard MPM, particles of all phases share the same background grid in which forces are applied and velocities are updated. The shared grid is then used to update velocity, position, and deformation gradient on the particles. Therefore, nearby particles have similar velocities regardless of their phase and density.

[6] presented a scheme for integrating buoyancy-driven phase separation in MPM, where forces are computed on a shared grid similar to the standard MPM, but are then split between phases using the volume fraction of each phase. Instead of using a single shared grid, each phase has its own grid for velocities which is updated using the forces assigned to it. These phase-specific grid velocities are then used to update the particle positions and velocities. To update the particle deformation gradients, the method uses the volume-weighted average velocity of all phases. The scheme presented by [6] allows for the simulation of multiphase immiscible fluids with different densities. However, this method has limitations, particularly in terms of computational efficiency and the ability to handle large time steps. In particular, the explicit nature of the time integration requires small time

steps to maintain stability, which can make the simulation computationally expensive for large or high-resolution problems. In this chapter, we present an implicit time integration method based on [6]’s method that allows for larger time steps and faster simulations while still capturing buoyancy-driven phase separation in MPM. We demonstrate the effectiveness of our method through a series of experiments and comparisons with the explicit method, showing that it can achieve similar results with significantly reduced computational time. While facilitating buoyancy-driven phase separation, the method of Ayazi et al. [6] also observed volume loss at the multiphase interface. In their explicit method, these errors remained at levels comparable to standard MPM. However, this issue become more pronounced with the larger time steps afforded by the implicit method and lead to significant inaccuracies in the simulation results. We investigate the source of the interfacial volume loss in the method of [6] and study two proposed methods for mitigating this loss. The contributions of this chapter are as follows:

- We extend the optimization-based integration framework of [40] to handle multiphase flows by formulating the problem as a constrained optimization problem and solve this using implicit time integration.
- We identify and correct an inconsistency in the deformation gradient update of Ayazi et al. [6] to improve volume conservation.
- We introduce a particle redistancing force that maintains consistency of the particle distribution and deformation gradients and thus alleviates volume loss.

4.2 Related Work

4.2.1 Material Point Method

[123] introduced Material point method as a hybrid method using both an Eulerian grid and lagrangian particles in which particles represent continuum of material capable of handling large deformations. [10] used material point method to simulated granular material. MPM was used first in computer graphics to simulate snow[120]. [121] expanded MPM's versatility beyond solid allowing MPM to simulate incompressible material and handle melting and solidifying dynamics. [64] introduced APIC in MPM, helping with the dissipation present in earlier versions of MPM. APIC also conserves linear and angular momentum improving visual fidelity. [73] used MPM to simulate grandular material by couling MPM with Drucker-Prager plasticity model. [23] introduced semi-implicit boundary conditions allowing MPM to interact with complex materials. [138] uses MPM to to simulate cracks. [127] introduced material specific grids allowing MPM to simulate mixture of sand and water. [34] introduced PolyPIC as a generalization of APIC. [54] proposed moving least squares MPM (MLS-MPM) that naturally leads to APIC and PolyPIC. [45] used MPM to simulate cloth and thin shell by combining MPM with subdivision finite elements allowing handling large number of collisions. [140] proposed a framework to handle multi-material simulations. [137] introduced phase field fracture and continuum damage mechanics to MPM, allowing for realistic crack propagation and tearing in soft bodies.[25] used MPM to simulate thermo-mechanical phenomena. [133] pushed MPM to handle hundreds of millions of particles. [31] addresses artifacts in solid-fluid coupling[98].

4.2.2 Multiphase MPM formulations

In computer graphics, the research on multiphase MPM has been mostly on solid-fluid interaction by introducing momentum exchange terms and phase-specific grid velocities [127, 36, 151]. [148] adapted MPM to handle Herschel-Bulkley plastic flows, allowing the natural simulation of shear-thickening and shear-thinning mixtures. [140] used a uniform MPM framework to handle fully coupled solid-fluid systems natively. [77] presented a method for multiphase flows in MPM where they use phase-specific grids to update particle positions, velocity and deformation gradient. Ayazi et al. [6] also presented a scheme for buoyancy-driven phase separation in which particle positions are updated using phase-specific grids but the deformation gradient of the particles is updated using a shared grid.

4.2.3 Multiphase flow simulation outside MPM

Multiphase simulation has been well studied in computer graphics, in both Eulerian frameworks and Lagrangian methods. Early grid-based methods mostly relied on level set methods techniques to track sharp interfaces between fluids, for simulation of multiple fluids [85]. While Eulerian methods excel at enforcing incompressibility and smooth surface tracking, they struggle with numerical dissipation and topological complexities when handling highly mixed, turbulent flows. On the other hand, particle methods natively conserve mass and handle advection easily. Smoothed Particle Hydrodynamics (SPH) has been widely adopted for both miscible and immiscible multi-fluid flows [94], with subsequent advancements extending SPH to support extreme density ratios [116], realistic mixing behaviors [84], and high-viscosity phase interactions [101]. Position-Based Dynamics (PBD)

frameworks have similarly used mixture models to couple a diverse range of fluids and interacting materials [88].

To model continuous diffuse interfaces, the graphics community has increasingly turned to phase-field models and continuum mixture theory [92]. By evolving an order parameter to track phase volume fractions, phase-field approaches, often governed by the Allen-Cahn or conservative Cahn-Hilliard equations [17], capture phase separation and surface tension within the fluid dynamics. However, integrating these thermodynamic models into traditional Navier-Stokes solvers introduces severe computational bottlenecks. Solving the high-order spatial derivatives required for the order parameter evolution typically demands complex implicit solvers or restrictively small time steps [41]. While these diffuse-interface approaches have been successful in SPH and PBD [143, 144], the overhead of evaluating the phase-evolution equations limits their scalability. [41] has proposed a simplified discretization of Cahn-Hilliard-Navier-Stokes equation breaking the fourth order PDE into two second order Helmholtz equations.

4.2.4 Implicit time integration

Implicit time integration methods are widely used in numerical simulations and computer graphics for their stability and ability to take larger time steps. The implicit methods typically require solving a system of non-linear equations. These systems of equations are usually solved using iterative methods such as Newton method. [39] show that formulating the implicit time integration problem as an optimization problem is more robust than solving it as a root finding problem. [40] presented a framework for implicit time integration in MPM by solving a minimization problem that minimizes the total energy of

the system. In this work we extend this framework to handle multiphase flows.

[30] introduced an implicit MPM framework capable of efficiently simulating highly stretchable rubber and nonlinear viscoelasticity.

4.2.5 Particle spacing methods

Maintaining a uniform particle distribution is a fundamental challenge in Lagrangian simulations. When particles clump together or form unphysical empty spaces, it leads to numerical instabilities, volume loss, and visual artifacts. In SPH, maintaining regular spacing is especially difficult due to the tensile instability, a mathematical flaw where particles group together under negative pressure [126]. Early solutions addressed this by introducing an artificial pressure term to explicitly repel neighbors that drift too close [93], a concept later adapted into position-level Δq corrections for Position Based fluids to enforce smooth free surfaces [89]. To prevent particle compression at interfaces, [110] utilized layers of ghost particles to maintain density kernel accuracy near boundaries. More recent SPH frameworks have moved away from artificial forces entirely, instead utilizing Fickian concentration gradients [81] or transport-velocity formulations [3] to gently shift particles along background pressure gradients, inherently guaranteeing a regularized, clump-free distribution. Kugelstadt et al. [74] proposed an implicit density correction for incompressible FLIP and APIC simulations to correct the position-level density errors that otherwise accumulate in the particle distribution.

While MPM avoids the tensile instability inherent to pure particle methods, it introduces spatial distribution challenges rooting in interpolations between particles and the background grid. Standard linear kernels cause artificial ringing and cell-crossing noise

that forces particles into clumps. The Generalized Interpolation Material Point Method (GIMP) was introduced to smooth these crossings using contiguous particle domains [9], while subsequent analyses demonstrated that higher-order B-splines significantly suppress quadrature errors and spatial clustering [118]. Furthermore, APIC method dramatically improved the stability of particle trajectories by maintaining local affine velocity matrices, preventing the clumping seen in earlier transfer schemes [64].

Despite these improvements to grid interpolation, extreme deformations can still cause issues with particle spacing. To overcome these issues, subsequent MPM frameworks have introduced explicit particle resampling and regularization techniques. In simulations of highly sheared non-Newtonian flows like foam, a dynamic particle splitting and merging strategy was introduced to eliminate holes forming when material points drift apart [148]. Similar adaptive resampling techniques were formalized for elastoplastic solids [37]. Furthermore, MLS-MPM formulation [54] advanced the foundational stability of APIC.

4.3 Method

Our goal is to devise an optimization-based implicit integration method, as described in Gast et al. [40], for the phase separation method described in Ayazi et al. [6]. At the start of each time step, we transfer particle momenta and volume to their phase-specific grids. We then solve an optimization problem to find the grid velocity changes that minimize the incremental potential energy of the system [40]. Finally, we transfer the resulting grid velocities back to the particles. We combine the phase-specific velocities using volume-

weighted averages and use the combined velocity field to update the particle deformation gradient.

4.3.1 Particle-to-grid transfer

The particle-to-grid transfer is implemented as in Ayazi et al. [6], where the velocities of the particles of each phase are mapped to its phase-specific grid. This is achieved by transferring each particle's momentum and mass to the corresponding phase-specific grid, then dividing the momentum by the mass to obtain the grid velocity.

$$m_{k,i}^n = \sum_{p \in P_k} w_{ip}^n m_p, \quad \mathbf{v}_{k,i}^n = \frac{1}{m_{k,i}^n} \sum_{p \in P_k} w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n (\mathbf{x}_i - \mathbf{x}_p^n)),$$

where P_k is the set of particles belonging to the k -th phase, $m_{k,i}^n$ and $\mathbf{v}_{k,i}^n$ are the mass and velocity of the k -th phase at the grid node i , w_{ip}^n is the interpolation weight between particle p and grid node i , m_p is the mass of particle p , $\mathbf{v}_p^n$ is the velocity of particle p , $\mathbf{C}_p^n$ is the affine velocity matrix of particle p , $\mathbf{x}_i$ is the position of grid node i , and $\mathbf{x}_p^n$ is the position of particle p . The superscript n indicates a time-varying quantity at time step n .

4.3.2 Grid-to-particle transfer

We denote by $\tilde{\mathbf{v}}_{k,i}^{n+1}$ the phase-specific grid velocities obtained through our implicit time integration scheme. After computing $\tilde{\mathbf{v}}_{k,i}^{n+1}$, we use APIC to transfer values from the grid to particles in the same manner as the standard MPM scheme:

$$\mathbf{v}_p^{n+1} = \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1}, \quad \mathbf{C}_p^{n+1} = \frac{1}{\xi} \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T, \quad p \in P_k,$$

where $\xi = \frac{\Delta x}{4}$ for quadratic splines. To update particle deformation gradients, we combine phase-specific velocities through volume averaging. This allows per-phase volume change while attempting to maintain the overall volume of the system. We denote values on the shared grid with a bar. We update the deformation gradient on particles as

$$\bar{\mathbf{v}}_i^{n+1} = \sum_k \alpha_{k,i} \tilde{\mathbf{v}}_{k,i}^{n+1}, \quad \nabla \bar{\mathbf{v}}_p^{n+1} = \sum_i \bar{\mathbf{v}}_i^{n+1} (\nabla w_{ip}^n)^T \quad (4.1)$$

$$\mathbf{F}_p^{n+1} = (\mathbf{I} + \Delta t \nabla \bar{\mathbf{v}}_p^{n+1}) \mathbf{F}_p^n \quad (4.2)$$

4.3.3 Implicit integration of grid forces

We now describe our implicit method for computing $\tilde{\mathbf{v}}_{k,i}^{n+1}$. Following [40], implicit backward Euler integration of the system, given by

$$\mathbf{M} \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} = \mathbf{f}(\mathbf{v}^{n+1}), \quad (4.3)$$

where $\mathbf{f}$ represents all implicitly integrated forces, yields the nonlinear system

$$h(\mathbf{v}^{n+1}) = \mathbf{M}(\mathbf{v}^{n+1} - \mathbf{v}^n) - \Delta t \mathbf{f}(\mathbf{v}^{n+1}) = \mathbf{0}. \quad (4.4)$$

Rather than solving this as a root-finding problem for the roots of the function $h(\mathbf{v}^{n+1})$, Gast et al. [40] interprets Eq. (4.4) as the first-order necessary condition in minimizing an objective function $E(\mathbf{v}^{n+1})$. Minimizing E is thus mathematically equivalent to solving the nonlinear system (4.4), but formulating it as a optimization problem allows for the use of

more robust solvers (e.g., safeguarded, Newton-based line search) which can be used with large time steps and in challenging scenarios.

Optimization-based integrator

The minimization objective corresponding to Eq. (4.4) is given by

$$E(\mathbf{v}) = \frac{1}{2}(\mathbf{v} - \mathbf{v}^n)^T \mathbf{M}(\mathbf{v} - \mathbf{v}^n) + \phi \quad (4.5)$$

where ϕ is defined such that $\frac{\partial \phi}{\partial \mathbf{v}} = \Delta t \mathbf{f}(\mathbf{v})$. Note that $\frac{\partial E}{\partial \mathbf{v}} = h(\mathbf{v}) = 0$ is the first-order necessary condition for a minimizer of (4.5), and thus corresponds to a solution of Eq. (4.4).

We solve a minimization problem using a Newton method similar to [40]. In each Newton iteration, we compute the total forces as the gradient of the potential. We then split the elastic forces based on the volume fraction of each phase. The volume fractions remain fixed during the Newton solve.

After the solve, particle positions and deformation gradients are updated as in the explicit method of Ayazi et al. [6], with phase-specific velocities used to update positions and the combined velocity used to update deformation gradients.

Elastic potential

We show that for the implicit MPM phase separation scheme, we can use the same elastic potential as in standard MPM. This is achieved because of the existing symmetry in the model in which we use the same weights to split the elastic force between the phase specific grids and to combine the velocities to update the deformation gradient. The elastic

potential can be written as Stomakhin et al. [122]:

$$\phi(\mathbf{x}) = \sum_p V_p^0 \Psi(\mathbf{F}_p(\mathbf{x})),$$

where $\mathbf{x}$ are all the grid node positions, V_p^0 is the initial volume of particle p , Ψ is the elastic energy density function, and $\mathbf{F}_p(\mathbf{x})$ is the deformation gradient of particle p , which is a function of the grid node positions. To compute the elastic forces we take the gradient of the potential with respect to the grid node positions:

$$\begin{aligned} \mathbf{f}_{k,i} &= -\frac{\partial \phi}{\partial \mathbf{x}_{k,i}} = -\sum_p V_p^0 \frac{\partial \Psi(\mathbf{F}_p)}{\partial \mathbf{F}_p} : \frac{\partial \mathbf{F}_p}{\partial \mathbf{x}_{k,i}} \\ \frac{\partial \mathbf{F}_p}{\partial \mathbf{x}_{k,i}} &= \frac{\partial((\mathbf{I} + \Delta t \nabla \bar{\mathbf{v}}_p) \mathbf{F}_p^n)}{\partial \mathbf{x}_{k,i}} \\ &= \alpha_{k,i} (\mathbf{F}_p^n)^T (\nabla w_{ip}^n) = \alpha_{k,i} \frac{\partial \mathbf{F}_p}{\partial \bar{\mathbf{x}}_i} \end{aligned}$$

Note that based on Eq. (4.1), $\frac{\partial \nabla \bar{\mathbf{v}}_p}{\partial \mathbf{x}_{k,i}} = \frac{\alpha_{k,i}}{\Delta t} \nabla w_{ip}$. The above shows that the elastic forces computed on each phase-specific grid are the same as the elastic forces computed on the combined grid multiplied by the volume fraction of each phase. This means that we can use the same elastic potential as standard MPM and split the resulting forces based on the volume fraction of each phase to get the forces on each phase-specific grid. Using the same weights to split the forces and combining the velocities to update the deformation gradient maintains the symmetry of the system, allowing us to reformulate the method of [6] as an implicit time integration.

Similarly, the per-phase and combined Hessians satisfy the following relationship:

$$\begin{aligned}
\mathbf{H}_{k,i,l,j} &= \frac{\partial^2 \phi}{\partial \mathbf{x}_{k,i} \partial \mathbf{x}_{l,j}} = \sum_p V_p^0 \frac{\partial^2 \Psi(\mathbf{F}_p)}{\partial \mathbf{F}_p^2} : \frac{\partial \mathbf{F}_p}{\partial \mathbf{x}_{k,i}} \otimes \frac{\partial \mathbf{F}_p}{\partial \mathbf{x}_{l,j}} \\
&= \sum_p V_p^0 \frac{\partial^2 \Psi(\mathbf{F}_p)}{\partial \mathbf{F}_p^2} : \alpha_{k,i} \frac{\partial \mathbf{F}_p}{\partial \bar{\mathbf{x}}_i} \otimes \alpha_{l,j} \frac{\partial \mathbf{F}_p}{\partial \bar{\mathbf{x}}_j} \\
&= \alpha_{k,i} \alpha_{l,j} \bar{\mathbf{H}}_{i,j}
\end{aligned}$$

In practice, we do not compute $\mathbf{H}$ directly. We compute $\mathbf{H}\tilde{\mathbf{v}}$ where $\tilde{\mathbf{v}}$ is the updated velocity at the end of each newton iteration.

$$(\mathbf{H}\tilde{\mathbf{v}})_{k,i} = \sum_{l,j} \mathbf{H}_{k,i,l,j} \tilde{\mathbf{v}}_{l,j} = \alpha_{k,i} \sum_{l,i} \alpha_{l,j} \bar{\mathbf{H}}_{i,j} \tilde{\mathbf{v}}_j = \alpha_{k,i} (\bar{\mathbf{H}}\tilde{\mathbf{v}})_i$$

The above holds because $\sum_l \alpha_{l,j} \tilde{\mathbf{v}}_{l,i} = \bar{\mathbf{v}}$ by definition.

In Section 4.3.5 we discuss how the observed volume loss occurs as a result of this interpolation.

4.3.4 Particle distancing force

One of the limitations of the method of Ayazi et al. [6] is the volume loss in the simulation of multiphase flows. When this framework is used in an implicit time integration, we observe faster phase separation but also more pronounced volume loss which leads to undesirable simulation results. To alleviate this issue, we introduce a particle distancing potential that penalizes nodes with a volume greater than the volume of a cell. The associated force thus penalizes particles creating areas of high density, which prevents drastic

Algorithm 1 Implicit Multi-phase MPM

1. Particle to grid transfer

$$\begin{aligned}
 m_{k,i}^n &= \sum_{p \in P_k} w_{ip}^n m_p ; & // \text{ Phase specific mass} \\
 \mathbf{v}_{k,i}^n &= (m_{k,i}^n)^{-1} \sum_{p \in P_k} w_{ip}^n m_p (\mathbf{v}_p^n + \mathbf{C}_p^n (\mathbf{x}_i - \mathbf{x}_p^n)) ; & // \text{ Velocity} \\
 V_{k,i}^n &= m_{k,i}^n / \rho_k \quad \alpha_{k,i} = V_{k,i}^n / \sum_k V_{k,i}^n ; & // \text{ Compute volume fraction}
 \end{aligned}$$

2. Compute grid velocities (Implicit Solve)

 Initialize $\mathbf{v}^n$ as starting guess

while not converged do

 Compute gradient ∇E

 if $\|\nabla E\| < \epsilon$ **then**

 | **break**

 Compute Hessian $\mathbf{H}$
 $\Delta \mathbf{v} = -\mathbf{H}^{-1} \nabla E$;

 Project $\Delta \mathbf{v}$ to ensure downhill direction

 Choose step size α using line search

 $\tilde{\mathbf{v}}^{n+1} = \mathbf{v}^n + \alpha \Delta \mathbf{v}$;

3. Transfer grid to particle using phase-specific velocity

$$\begin{aligned}
 \mathbf{v}_p^{n+1} &= \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1}, \quad p \in P_k ; & // \text{ Update particle velocity} \\
 \mathbf{x}_p^{n+1} &= \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1} ; & // \text{ Update particle position} \\
 \mathbf{C}_p^{n+1} &= \frac{1}{\xi} \sum_i w_{ip}^n \tilde{\mathbf{v}}_{k,i}^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T, \quad p \in P_k;
 \end{aligned}$$

4. Update deformation gradient

$$\begin{aligned}
 \bar{\mathbf{v}}_i^{n+1} &= \sum_k \alpha_{k,i} \tilde{\mathbf{v}}_{k,i}^{n+1} ; & // \text{ Combine phase-specific velocities} \\
 \nabla \bar{\mathbf{v}}_p^{n+1} &= \sum_i \bar{\mathbf{v}}_i^{n+1} (\nabla w_{ip}^n)^T ; \\
 \mathbf{F}_p^{n+1} &= (\mathbf{I} + \Delta t \nabla \bar{\mathbf{v}}_p^{n+1}) \mathbf{F}_p^n ;
 \end{aligned}$$

total volume loss. The particle distancing potential function is defined as:

$$\phi = c \sum_i g(V_i), \quad g(V_i) = \begin{cases} \frac{1}{V_{cell}} (V_i - V_{cell})^2 & V_i > V_{cell} \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

$$V_i = \sum_p w_{ip} V_p \quad (4.7)$$

Here, c is a coefficient to change the strength of the potential, V_i is the volume of node i , V_{cell} is the cell volume on the uniform background grid, and w_{ip} is the interpolation weight between particle p and grid node i . This potential is fundamentally particle-based; its value

depends on particle positions. Because moving the grid nodes does not alter the weights w_{ip} , the grid nodes are not the true degrees of freedom. However, implicit MPM requires forces to be evaluated on the grid before updating particle velocities. To resolve this, we conceptualize two grids: a moving “particle grid” where interpolation weights remain constant, and a fixed “background grid” where weights change as particles move. We use the background grid weights to compute $\bar{V}_i$ and the particle grid weights to compute particle positions $\mathbf{x}_p$. We calculate the required grid forces by taking the gradient of the potential with respect to the particle grid nodes.

Values with a bar refer to the background grid and values without a bar on the grid refer to the particle grid.

$$\bar{V}_i = \sum_p \bar{w}_{ip} V_p$$

$$\mathbf{x}_p = \sum_j w_{jp} \mathbf{x}_j$$

$$\bar{w}_{ip} = N(\bar{\mathbf{x}}_i - \sum_j w_{jp} \mathbf{x}_j)$$

$$\nabla \bar{w}_{ip} = \nabla N(\bar{\mathbf{x}}_i - \mathbf{x}_p)$$

The potential function can be written as:

$$\phi = c \sum_i g(\bar{V}_i) = \sum_i g \left(\sum_p N \left(\bar{\mathbf{x}}_i - \sum_j w_{jp} \mathbf{x}_j \right) V_p \right)$$

Taking the gradient with respect to the particle grid nodes:

$$\frac{\partial \phi}{\partial \mathbf{x}_s} = c \sum_i \frac{\partial g(\bar{V}_i)}{\partial \mathbf{x}_s} = \sum_i g'(\bar{V}_i) \cdot \frac{\partial \bar{V}_i}{\partial \mathbf{x}_s}$$

gradient of $\bar{V}_i$ with respect to $\mathbf{x}_s$:

$$\begin{aligned} \frac{\partial \bar{V}_i}{\partial \mathbf{x}_s} &= \left[\sum_p V_p \nabla \bar{w}_{ip}^T \left(\frac{\partial \left(\bar{\mathbf{x}}_i - \sum_j w_{jp} \mathbf{x}_j \right)}{\partial \mathbf{x}_s} \right) \right]^T \\ &= \left[\sum_p V_p \nabla \bar{w}_{ip}^T (-w_{sp} \mathbf{I}) \right]^T \\ &= \sum_p -V_p w_{sp} \nabla \bar{w}_{ip} \end{aligned}$$

So the force on the grid can be written as:

$$\frac{\partial \phi}{\partial \mathbf{x}_s} = c \sum_i 2 \left(\frac{\bar{V}_i}{\bar{V}_{cell}} - 1 \right) \left(\sum_p -V_p w_{sp} \nabla \bar{w}_{ip} \right) \quad (4.8)$$

To be able to use this force in the implicit framework, as a minimization problem, using the Newton method we also need the second derivative (Hessian) of the potential function.

Hessian of the particle distancing potential

$$\begin{aligned}
H_{rs} &= \frac{\partial^2 \phi}{\partial \mathbf{x}_r \partial \mathbf{x}_s} = \frac{\partial}{\partial \mathbf{x}_r} \left(c \sum_p -V_p w_{sp} \sum_i 2 \left(\frac{\bar{V}_i}{\bar{V}_{cell}} - 1 \right) (\nabla \bar{w}_{ip}) \right) \\
&= \sum_p 2V_p w_{sp} \sum_i \left(\left(\frac{\bar{V}_i}{\bar{V}_{cell}} - 1 \right) (\nabla (\nabla \bar{w}_{ip})) (w_{rp}) + \right. \\
&\quad \left. \left(\frac{1}{\bar{V}_{cell}} \sum_q V_q w_{rq} \nabla \bar{w}_{iq} \right) \otimes (\nabla \bar{w}_{ip}) \right)
\end{aligned}$$

In a multiphase context, to compute the V_i , we use all the particles in the simulation since this force is designed to make the total volume distribution of particles uniform and should not care about the phase or other physical properties of particles. However, to take the gradient of the potential function, we consider phase-specific grid nodes as degrees of freedom, since each phase needs to maintain its own volume distribution. We can control the relative contribution of the particle distancing potential in the overall objective of the system by scaling it with a constant coefficient.

4.3.5 Volume loss

Volume loss is a common issue in MPM fluid simulations. This issue becomes persistent throughout the simulation in the multiphase MPM scheme. Below we discuss the

cause of this issue and discuss possible solutions. In a two phase flow we have

$$\dot{\mathbf{F}} = \nabla \bar{\mathbf{v}} \mathbf{F}, \quad \bar{\mathbf{v}} = \sum_k \alpha_k \tilde{\mathbf{v}}_k,$$

$$\dot{J} = \nabla \cdot \bar{\mathbf{v}} J, \quad J = \det(\mathbf{F})$$

Consider a region Ω containing particles $p \in \Omega$. We define:

$$J(\Omega) = \frac{\sum_{p \in \Omega} J_p}{\sum_{p \in \Omega} 1}$$

and the relative volume of phase k in Ω as

$$\eta_k(\Omega) = \frac{\sum_{p \in \Omega} V_0^p}{\text{vol}(\Omega)}, \quad p \in P_k$$

We note that η_k could also be interpreted as the phase mass normalized by the phase rest density, and thus it follows the conservation law:

$$\dot{\eta}_k = -\eta_k \nabla \cdot \mathbf{v}_k, \quad k = 0, 1$$

In the single phase setting, we would have $J\eta = 1$. Similarly, in the two-phase setting, we have $J(\eta_0 + \eta_1) = 1$ for any volume in space. We define $\eta = \eta_0 + \eta_1$ and $\mathbf{u}_c$ as the interpolation of $\mathbf{u}_0$ and $\mathbf{u}_1$ on particles using a weighted average α_0, α_1 on particles. Taking

the divergence of $\mathbf{u}_c$:

$$\begin{aligned}
\nabla \cdot \mathbf{u}_c &= \nabla \cdot (\alpha_0 \mathbf{u}_0 + \alpha_1 \mathbf{u}_1) \\
&= \nabla \cdot ((\eta_0 \mathbf{u}_0 + \eta_1 \mathbf{u}_1)/\eta) \\
&= \frac{1}{\eta} \nabla \cdot (\eta_0 \mathbf{u}_0 + \eta_1 \mathbf{u}_1) - \frac{1}{\eta^2} (\eta_0 \mathbf{u}_0 + \eta_1 \mathbf{u}_1) \cdot \nabla \eta \\
&= \frac{1}{\eta} \nabla \cdot (\eta_0 \mathbf{u}_0 + \eta_1 \mathbf{u}_1) - \frac{1}{\eta} \mathbf{u}_c \cdot \nabla \eta
\end{aligned}$$

Using this, we will find the correct update rule for J .

$$\eta_{kt} = \dot{\eta}_k - \mathbf{u}_k \cdot \nabla \eta_k = -\nabla \cdot (\eta_k \mathbf{u}_k), \quad k = 0, 1$$

$$(\eta_0 + \eta_1)J_t = -(\eta_{0t} + \eta_{1t})J$$

$$\eta J_t = (\nabla \cdot (\eta_0 \mathbf{u}_0) + \nabla \cdot (\eta_1 \mathbf{u}_1))J$$

$$\begin{aligned}
\eta(J_t + \bar{\mathbf{u}} \cdot \nabla J) &= (\nabla \cdot (\eta_0 \mathbf{u}_0) + \nabla \cdot (\eta_1 \mathbf{u}_1))J + \eta \bar{\mathbf{u}} \cdot \nabla J \\
&= (\nabla \cdot (\eta_0 \mathbf{u}_0) + \nabla \cdot (\eta_1 \mathbf{u}_1))J - \bar{\mathbf{u}} \cdot (\nabla \eta_0 + \nabla \eta_1)J \\
&= (\eta_0 \nabla \cdot \mathbf{u}_0 + \eta_1 \nabla \cdot \mathbf{u}_1)J \\
&\quad + ((\mathbf{u}_0 - \bar{\mathbf{u}}) \cdot \nabla \eta_0 + (\mathbf{u}_1 - \bar{\mathbf{u}}) \cdot \nabla \eta_1)J
\end{aligned}$$

where $\bar{\mathbf{u}}$ is the interpolated velocity on the particle and $\mathbf{u}_0$ and $\mathbf{u}_1$ are the interpolated velocities for each phase.

which means we can write the update for J as:

$$J_t + \bar{\mathbf{u}} \cdot \nabla J = \nabla \cdot \mathbf{u}_c J + \frac{J}{\eta} (\mathbf{u}_c - \bar{\mathbf{u}}) \cdot \nabla \eta$$

4.4 Results

Table 4.1: Timing comparison of methods in 2D and 3D. Entries marked “-” indicate that the method was not evaluated. Initial elasticity values in dambreak 1 differ from other tests to match the results reported by [6]. PD indicates that the particle distancing force was used in addition to the implicit time integration scheme.

Test Name	Particles	Res.	Threads	E	Implicit MPM (s)	Implicit Multi- phase (s)	Implicit + PD (s)	Explicit Multi-phase [6] (s)
Stable State	121180	128×256	8	10^5	32.811	19.915	34.190	45.511
Partially Mixed	121172	128×256	8	10^5	35.000	53.895	118.339	81.841
Rayleigh-Taylor Instability	121180	128×256	8	10^5	79.634	114.413	190.372	75.870
Dam Break (Version 1)	65566	256×256	8	10^5	64.821	56.069	68.421	54.394
Dam Break (Version 2)	42932	256×256	8	10^4	22.343	30.001	41.463	12.591

We compare our results to those of the baseline explicit time integration scheme.

The implicit method enables significantly larger time steps, thereby requiring fewer total time steps overall to reach the same simulation time. We observe that phase separation occurs more quickly with the larger time steps facilitated by the implicit method. Additionally, larger time steps incur increased volume loss. Suprisingly, we also observe more dynamic motion using the implicit method, despite the well-known numerical damping of the implicit time integration scheme. This could be due to the substantial reduction in total time steps, and hence the number of MPM particle-to-grid and grid-to-particle transfers, which inherently dissipate particles velocities.

We use standard MPM with implicit time integration as our baseline for comparison of the volume loss. To reduce the volume loss in the implicit method, we can adjust the coefficient of the particle distancing potential or use higher stiffness coefficients for our elastic forces. Both of the above approaches can help mitigate the volume loss issue when using an implicit time integration scheme. However they also introduce overhead and computational cost, slowing down the simulation run time. We have noticed that with a higher coefficient for the particle distancing potential, the fluid dynamics become more like sand and less like fluid. This is because as the contribution of the particle distancing force increases, the particles will be penalized more for passing each other to maintain the uniform volume distribution. If a particle needs to move upward, there must be a particle moving downward to replace it in a way that is not compromising the overall uniform volume distribution.

In all tests, we use slip boundary conditions, with a fixed corotated constitutive model [119]. We set $\mu = 0$ to prevent shear forces, and we set our Poisson ratio $\nu = 0.3$ and Young's modulus $E = 1e5$ and densities of 1 and 10 unless specified otherwise. Gravity is set to 10 in the negative y direction. We use quadratic B-splines [117, 64] for transfers between the particles and grids. Each time step, we reset $\mathbf{F}_p^n \leftarrow \det(\mathbf{F}_p^n)^{1/d} \mathbf{I}$ to prevent accumulation of deviatoric components[125]. The timings for the simulations can be found in Table 4.1. For Rayleigh-Taylor and dam break tests we noticed a slower simulation using implicit time integration than the explicit time inetegration in resolution 128. We compared the computational performance of both approaches using the Rayleigh-Taylor instability test and 2D dam break version 1 scenarios across different resolution. The comparison of the resolution

vs speed for implicit and explicit time integration can be found in Figure 4.12. Explicit time integration performs better at lower resolutions, but as the resolution increases, the implicit method becomes more computationally efficient. In both test cases, the implicit method scales better with increased resolution; this reduces the performance gap between the two methods in the dam break test and allows the implicit method to surpass the explicit method in the Rayleigh-Taylor test. In all cases, the maximum allowed time step, Δt , was restricted to 1×10^{-3} , ensuring each frame consists of at least 40 substeps. Furthermore, we observed that although the implicit method can successfully solve the minimization problem using larger time steps, it is usually computationally faster when taking smaller time steps. To further accelerate the implicit scheme, future work could utilize quasi-Newton methods to approximate the Hessian of the potential.

We have also observed significant computational overhead when using the particle distancing potential, as the computation of its gradient and Hessian require eleven particle-grid transfers in total. Although the number of Newton iterations taken by our implicit phase-separating MPM is comparable to the number taken in standard implicit MPM, the computational overhead from the particle distancing force slows the simulation. We have noticed that using a smaller Δt (about 2^{-3}) will increase the implicit time integration’s performance

4.4.1 Stable state test

We evaluate our method on a 2D test in a state of equilibrium where a lighter fluid is positioned above a heavier fluid. The expected behavior is for the fluids to have

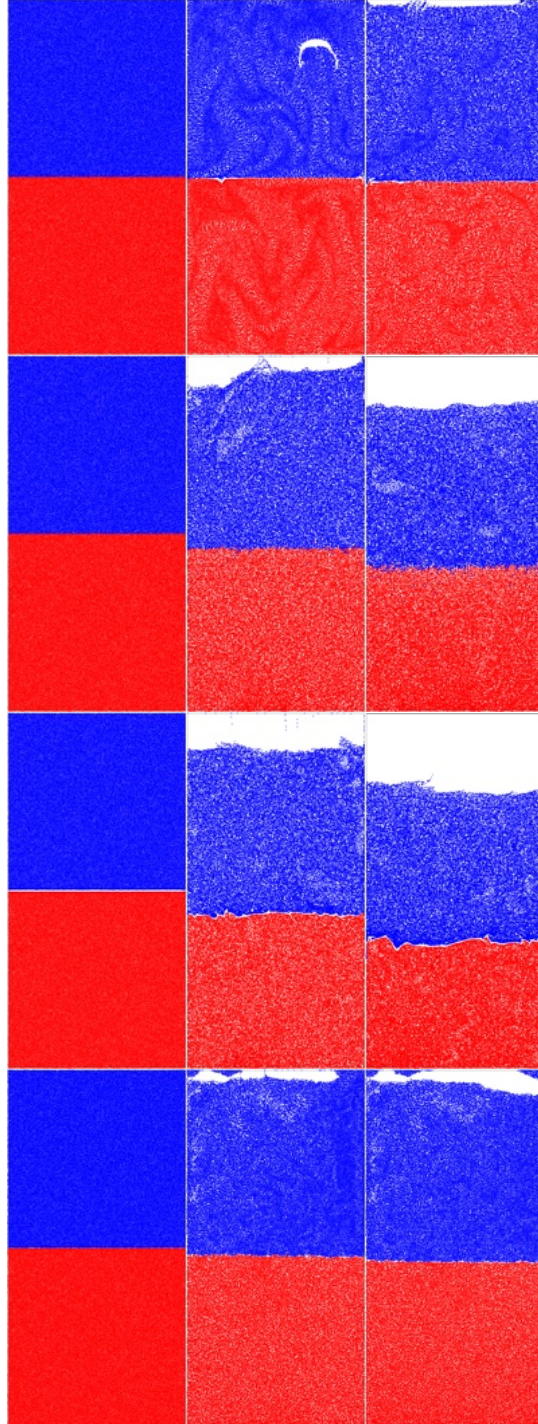


Figure 4.1: A stable state test with lighter fluid on top of a heavier fluid at times $t = 0, 5, 10s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), and phase-separating MPM with implicit time integration and the particle distancing force (fourth row).

minimal movement and remain in their initial positions. The results of the test are shown in Figure 4.1. Significant volume loss is observed in all the simulations run using the implicit time integration scheme, with the phase-separating MPM showing the most severe volume loss. Some mixing is observed in the standard MPM case. The explicit time integration scheme of the phase-separating MPM shows the least amount of volume loss. The implicit time integration scheme with the particle distancing force is the second best in terms of volume loss, showing more volume loss than the explicit scheme but less than the standard MPM and the implicit multiphase MPM scheme. Except for the implicit standard MPM scheme in which some mixing is observed, all other simulations maintain the separated state of the fluids. A coefficient of 10 was used for the particle distancing force. The coefficient was chosen to be large enough to reduce volume loss, but not so large as to disrupt the fluidity of the material.

4.4.2 Partially mixed test

We test our method on a 2D test with a lighter fluid above a denser fluid and a mixed region in the middle containing both fluids. The expected behavior is for the lighter particles to rise and for the denser particles to sink. The results of the test are shown in Figure 4.2. Similar to the stable state test, the explicit time integration scheme of the phase-separating MPM shows the least amount of volume loss. The implicit time integration scheme with the particle distancing force is the second best in terms of volume loss, showing more volume loss than the explicit scheme but less than the standard MPM and the implicit multiphase MPM scheme. The implicit time integration scheme with the

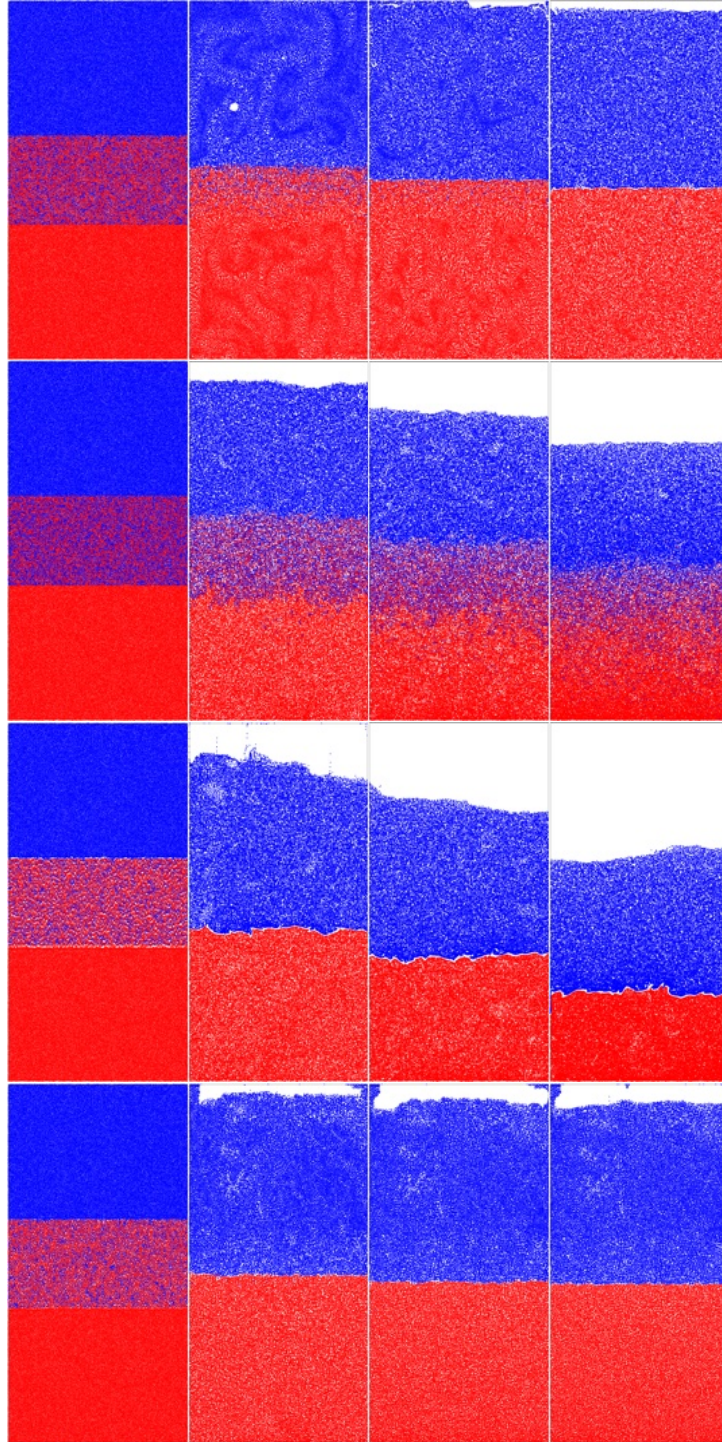


Figure 4.2: A partially mixed test with lighter fluid on top of a heavier fluid with a mixed region in the middle at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).

phase-separating MPM shows the most severe volume loss. Phase separation is observed in all the simulations except the standard MPM. Phase separation is observed in both the implicit time integration with and without the particle distancing by frame 120 whereas the explicit time integration scheme of the phase separating MPM shows phase separation by frame 240. A coefficient of 10 was used for the particle distancing force.

4.4.3 Rayleigh-Taylor test

We run a 2D Rayleigh-Taylor instability[104] test with a denser fluid on top of a lighter fluid. The results of the test are shown in Figure 4.3. Similar to the previous test, we observe faster phase separation when using the implicit time integration scheme and lower volume loss with the explicit time integration scheme and test case where particle distancing force is used. Phase separation is observed in all the simulations except the standard MPM where the phases form a gradient from denser to lighter fluid from bottom to top. Phase separation is observed with the implicit time integration scheme both with and without the particle distancing by frame 240 whereas the explicit time integration scheme of the phase-separating MPM shows phase separation by frame 480 with some mixing remaining at that frame. A coefficient of 10 was used for the particle distancing force. When running the implicit time integration scheme with a small time step size close to that of the explicit time integration scheme, we observed similar results with both methods (See Figure 4.6). However, the implicit time integration scheme without the particle distancing force forms regions of high particle density as the fluids start to mix yielding non-physical

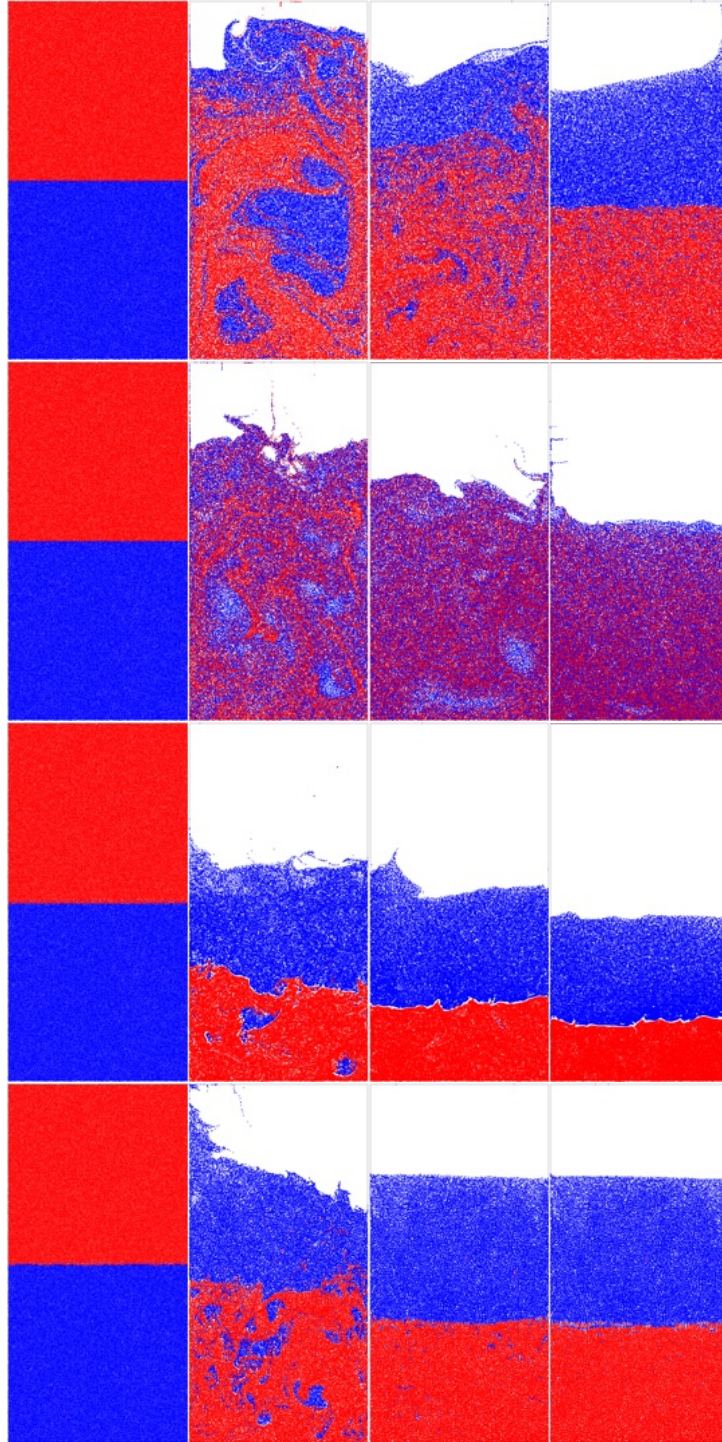


Figure 4.3: Rayleigh-Taylor test with denser fluid on top of a heavier fluid with a mixed region in the middle at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).

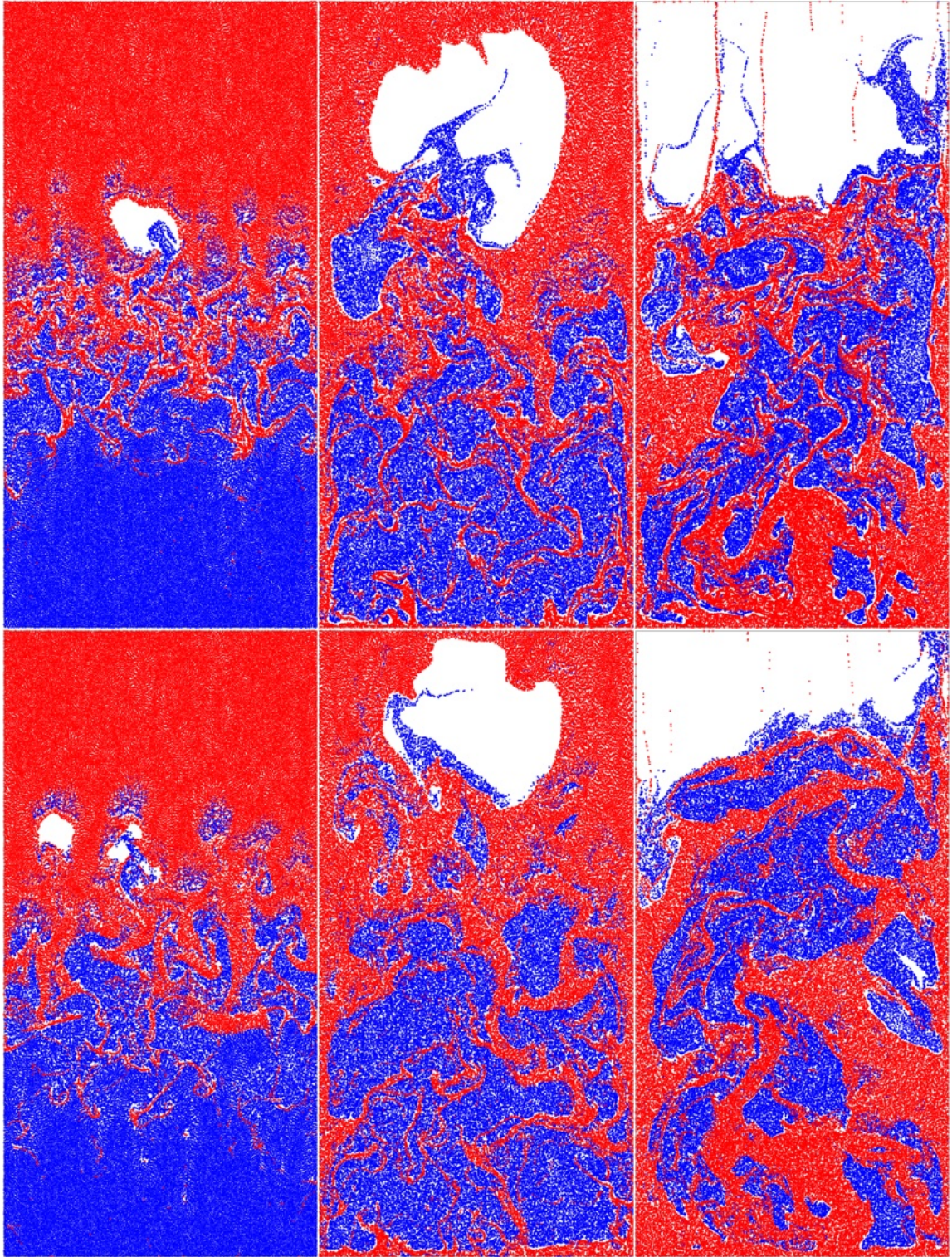


Figure 4.4: Rayleigh-Taylor test with implicit time integration scheme without the particle distancing force (first row) and with the particle distancing force (second row) at times $t = 0.8, 1.6, 2.4s$. Both form patterns of high particle density particle as the fluids start to mix creating non-physical behavior.

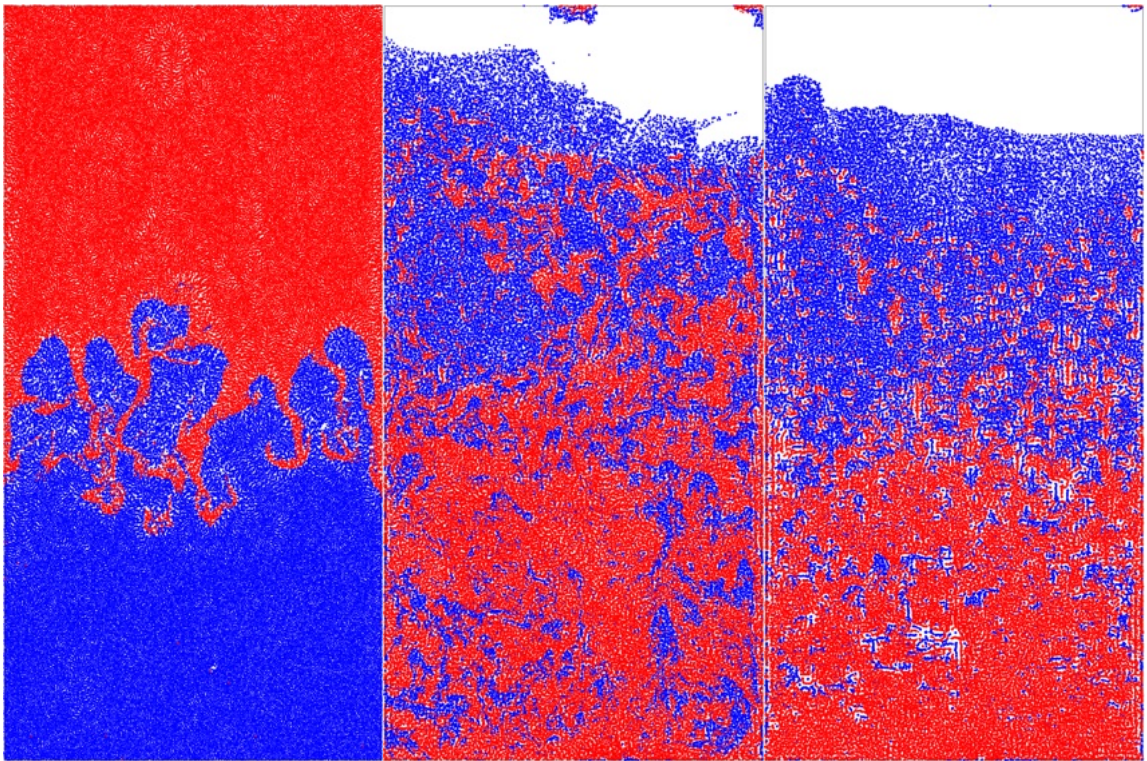


Figure 4.5: Rayleigh–Taylor instability test utilizing the implicit time integration scheme with a particle distancing coefficient of $c = 50$ at $t = 1, 5, 20s$. A distinct sand-like behavior is observed during the settling phase; the fluid motion becomes less dynamic as the particle distancing potential becomes the dominant contributor to the total potential of the system.

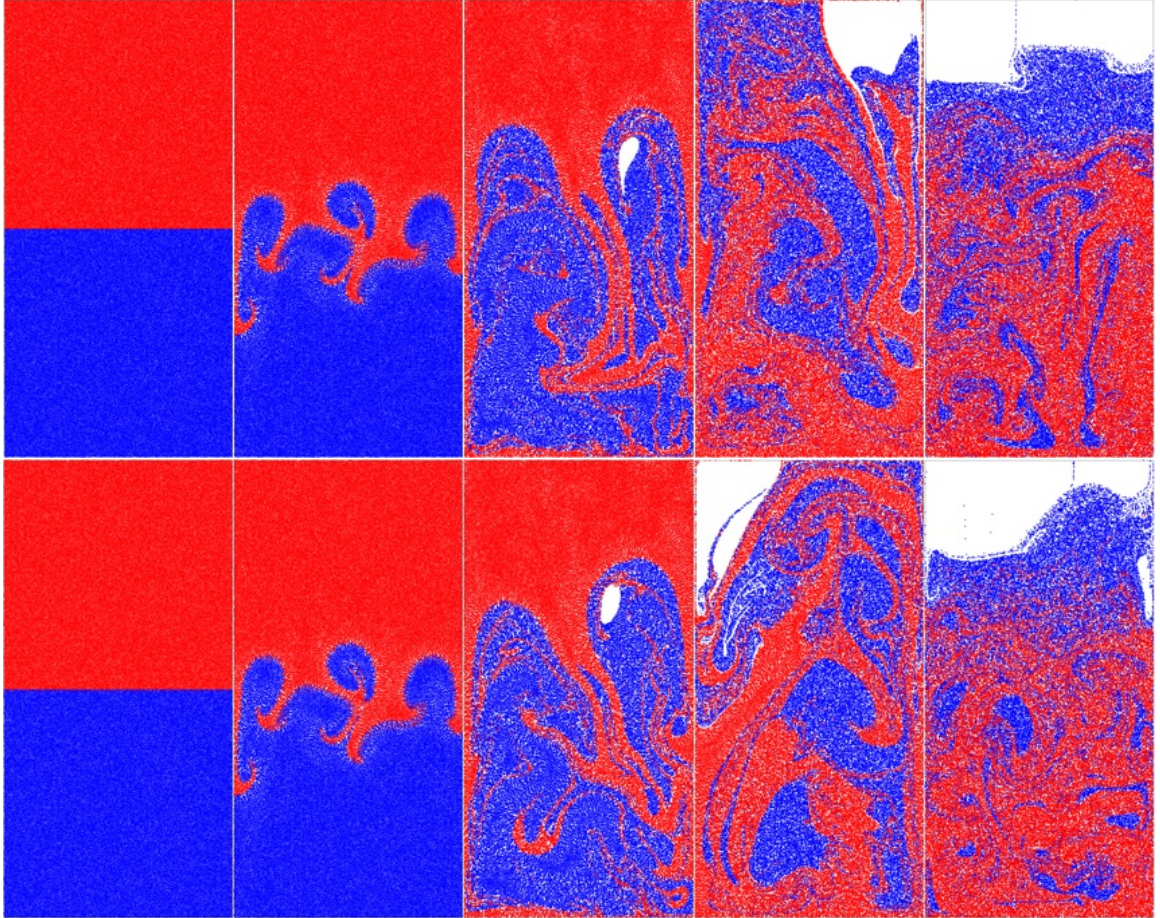


Figure 4.6: Rayleigh-Taylor test with explicit time integration scheme (first row) and with implicit time integration scheme with $\Delta t = 2.5 \times 10^{-5}$ (second row) at times $t = 0, 1, 2, 4, 8s$. $\Delta t = 2.5 \times 10^{-5}$ is the average time step size used in the explicit scheme for this specific test. Both schemes show similar development when they use similar time step sizes.

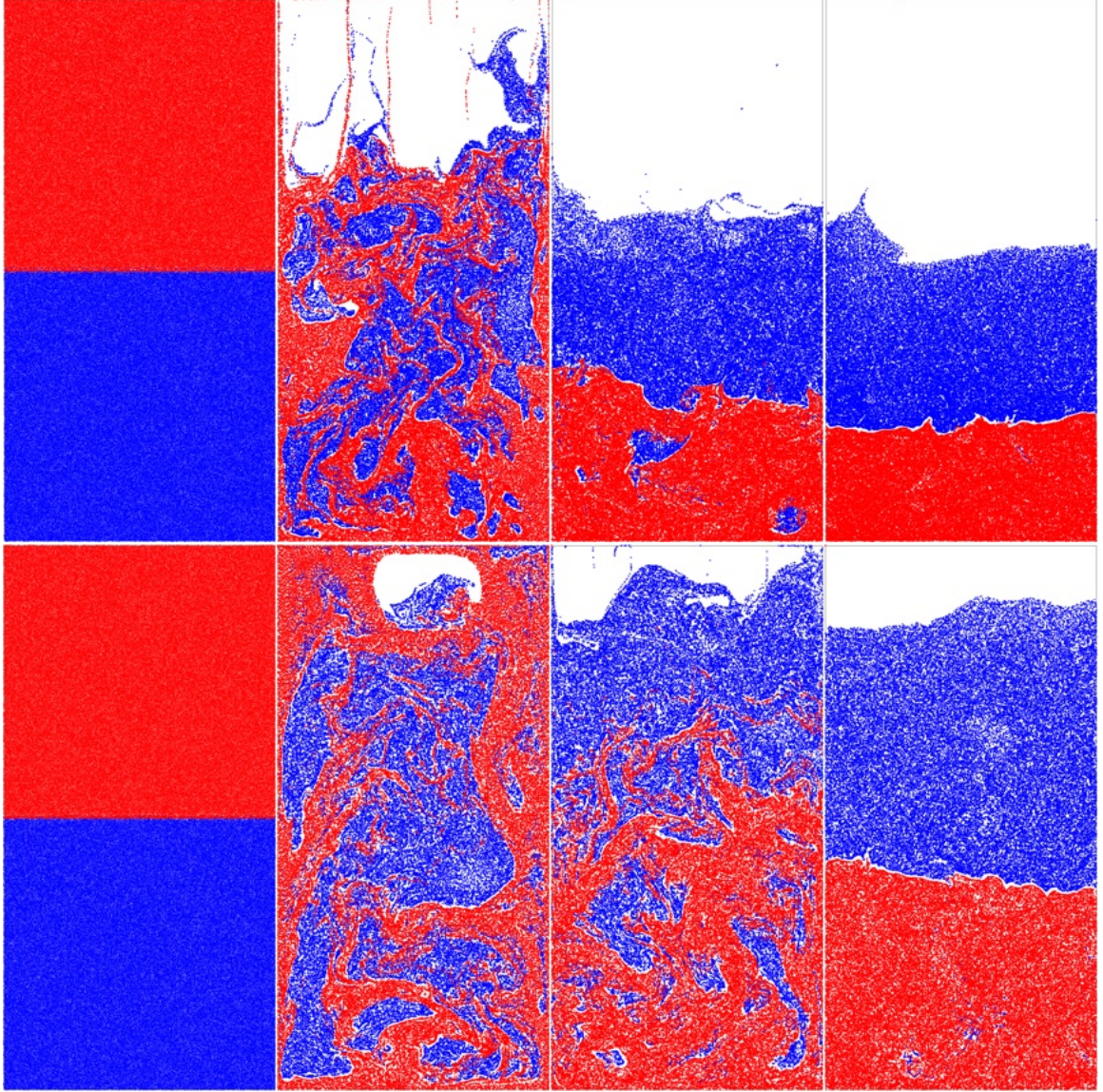


Figure 4.7: Rayleigh-Taylor test with implicit time integration scheme and Young's modulus of $E = 10^5$ (first row) and Young's modulus of $E = 10^7$ (second row) at times $t = 0, 2.5, 5, 10s$. The test with higher stiffness shows less volume loss.

behavior(Figure 4.4). The same behavior is still observed if the coefficient of the particle distancing force is smaller. Using a larger coefficient for the particle distancing force, prevents the lines from forming but creates sand like behavior once the fluids start settling and it prevents separation(Figure 4.5) This is because as the simulation starts settling, the particle distancing potential becomes the dominant term in the total potential of the systems, preventing the particles to move passed each other. As a results particle movement only happens when there is enough empty space in the vicinity of the particles resuling in movements where particles slide next to each other instead of passing each other.

We compare the affect of stiffness on volume loss in Figure 4.7. We observe slower volume loss in the Rayleigh-Taylor test with higher stiffness. However, using higher stiffness, usually requires more newton iterations to find the minimum of the potential function, resulting in a slower run time. Figure 4.7 illustrates the effect of stiffness on volume loss. We observe a slower rate of volume loss in the Rayleigh-Taylor test when higher stiffness is applied. However, increasing the stiffness typically requires more Newton iterations to minimize the potential function, resulting in longer execution times.

4.4.4 Dam break test

We run two versions of the dam break test with a denser fluid above a lighter fluid. In the first version, we have a denser fluid directly above a lighter fluid. In the second version, we have a denser fluid above and shifted to the right of a lighter fluid. We use a Young's modulus of $E = 1e4$ to match the setup of Ayazi et al. [6]. The results are shown in

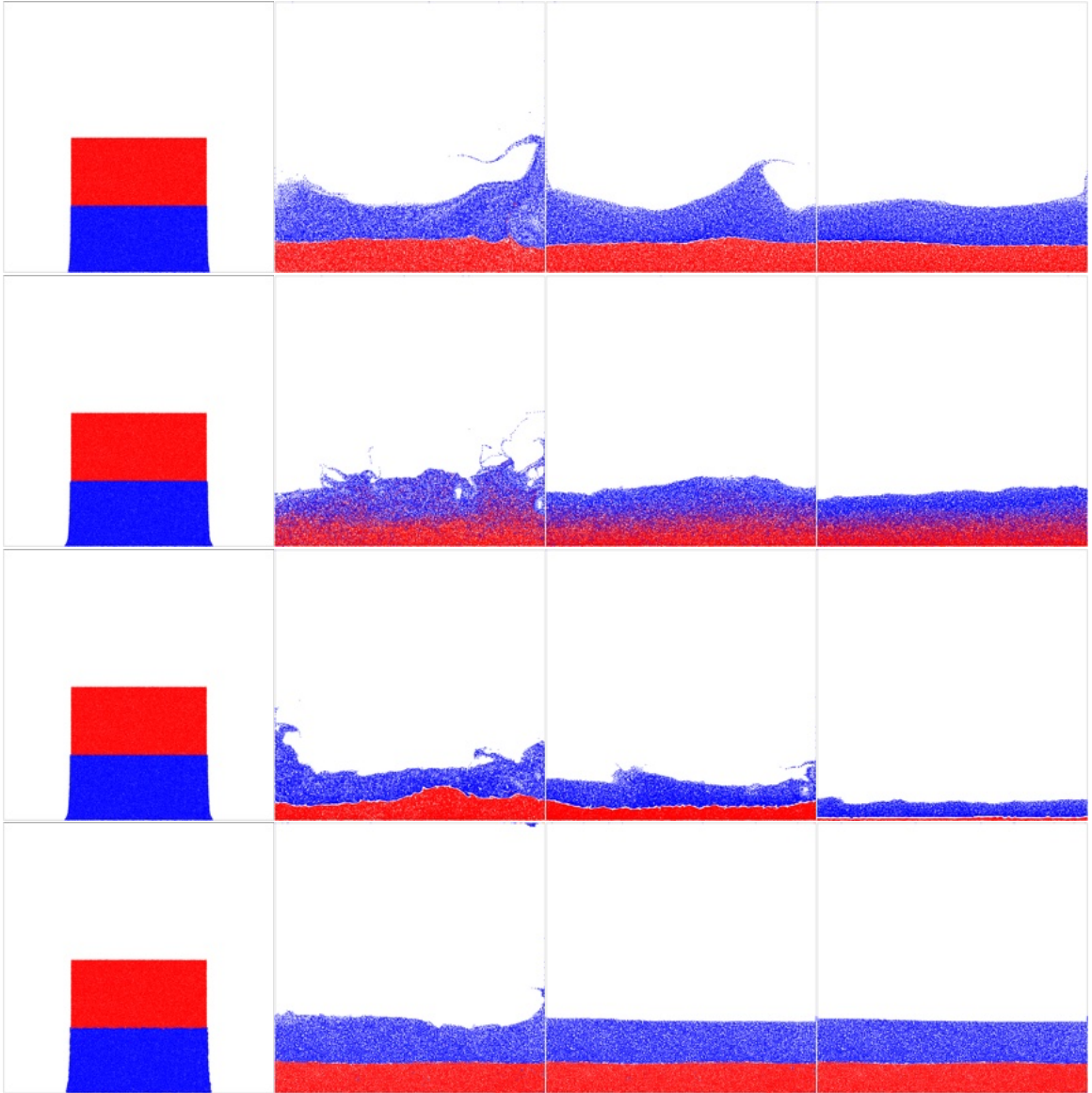


Figure 4.8: Dam break test with lighter fluid on top of a heavier fluid at times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).

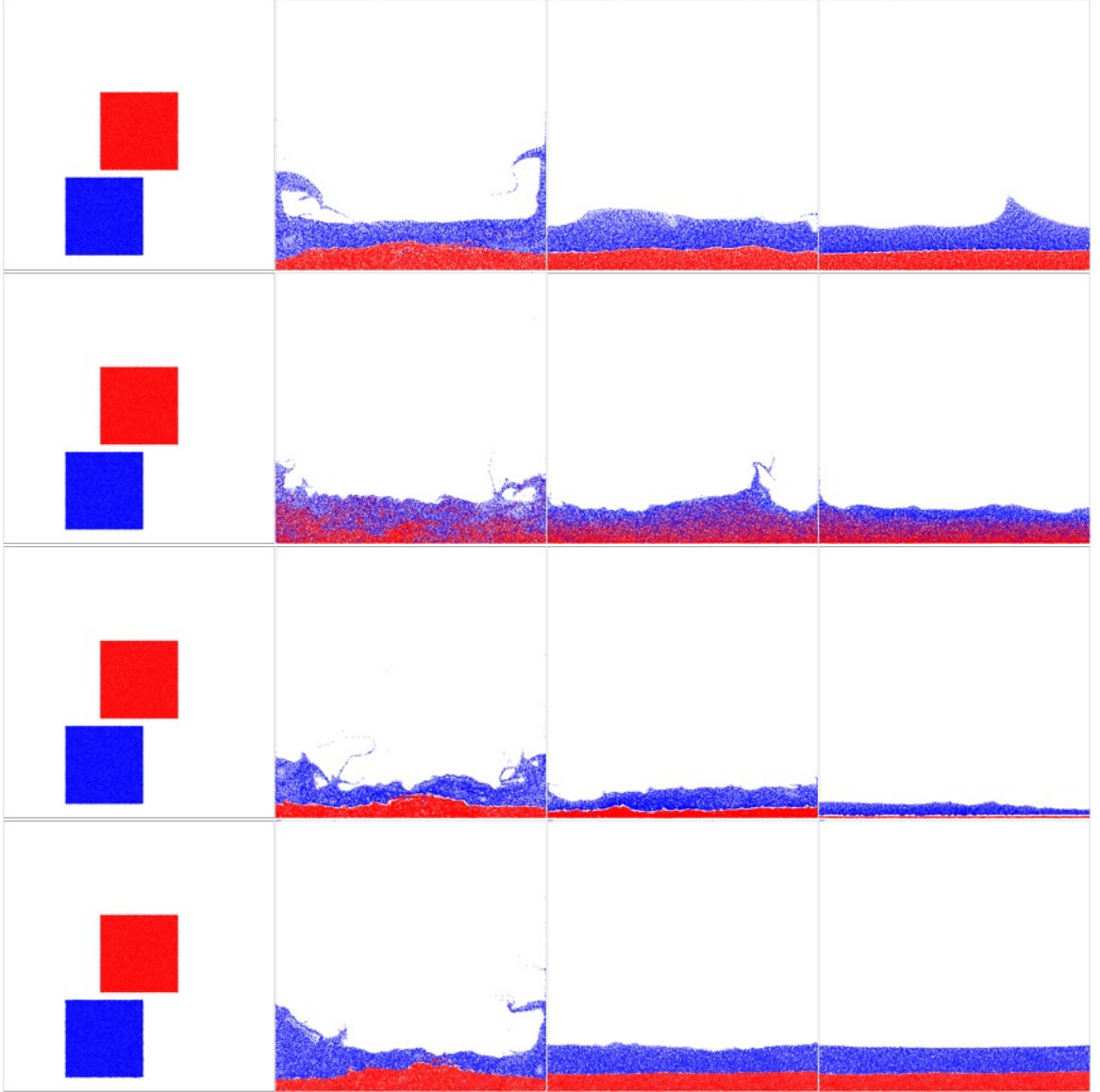


Figure 4.9: Dam break test with lighter fluid on top of a heavier fluid with an offset to the right at times $t = 0, 5, 10, 20$ s from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).

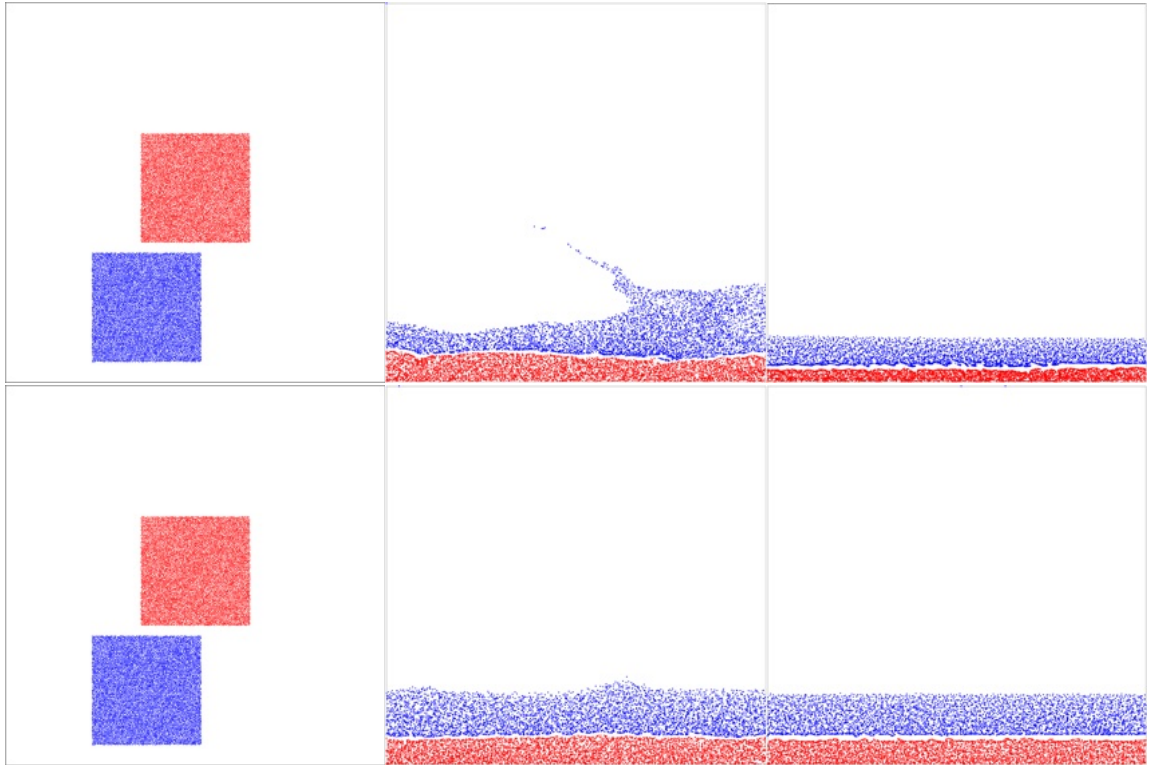


Figure 4.10: Dam break test with lighter fluid on top of a heavier fluid with an offset to the right at times $t = 0, 10, 20, 40, 60s$ from left to right using the baseline [6] method without the particles distancing force (top row), and with the particles distancing force (second row). The test case with the particle distancing force maintains its volume compared to the test case without the particle distancing force, which shows a loss of volume as the simulation progresses.

Figure 4.8 and Figure 4.9. The results for both versions are consistent with the results of the previous tests. The fluid motion is more dynamic in tests without the particle distancing force and the volume loss is more severe as a result of both the larger interface between the two fluids and the reduced stiffness. We also investigated the effect of adding a particle distancing force to the explicit multi-phase MPM. The results are shown in Figure 4.10. The test was conducted using a Young’s modulus of $E = 10^4$ and a grid resolution of 128^2 , running for an extended duration of 2,000 frames. As the simulation progresses, the baseline explicit method begins to suffer from volume loss, whereas the formulation run with the particle distancing force successfully maintains its volume. By the end of the simulation, the volumetric discrepancy between the two approaches is noticable.

4.4.5 Drop test

We run a 2D test where a drop of a lighter fluid is dropped into a pool of heavier fluid. The results are shown in Figure 4.11. The phases mix and separate in all the simulations except for the standard MPM, where some mixing persists throughout the simulation. Unlike previous tests, the least volume loss is observed with the implicit time integration scheme with the particle distancing force. The explicit and implicit time integration schemes of the phase-separating MPM and the standard MPM show similar volume loss. A coefficient of 10 was used for the particle distancing force.

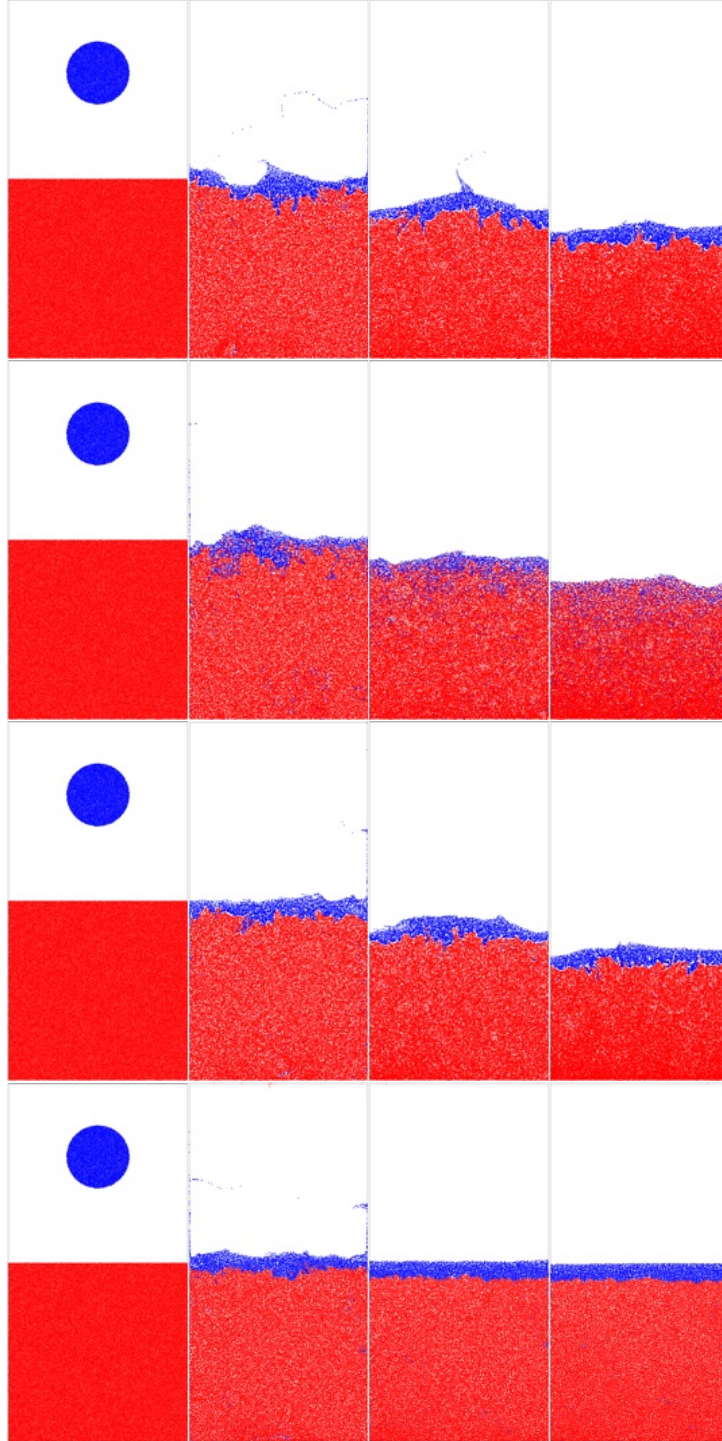


Figure 4.11: a drop of lighter fluid fall into a pool of heavier fluid. Times $t = 0, 5, 10, 20s$ from left to right using the baseline [6] method (top row), standard MPM with implicit time integration (second row), phase-separating MPM with implicit time integration (third row), phase-separating MPM with implicit phase separation and the particle distancing force (fourth row).

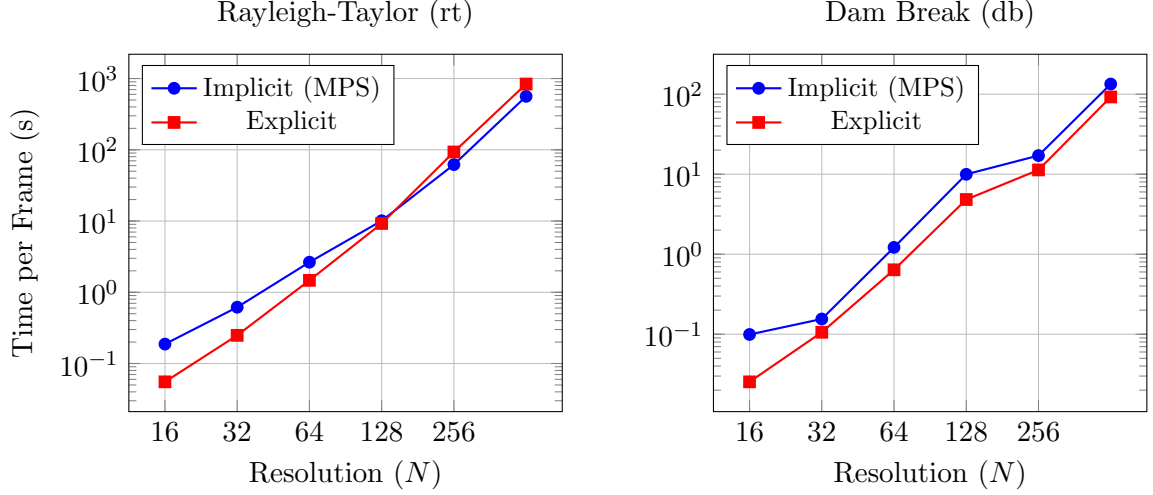


Figure 4.12: Computational efficiency comparison demonstrating execution time per frame across scaling mesh grid resolutions (16^2 to 512^2) for implicit and explicit multi-phase formulations. All simulations were run using 8 threads and for 10 frames with maximum Δt of 2×10^{-3} .

4.5 Conclusion and Limitations

In this chapter, we presented an implicit time integration scheme for the multi-phase material point method (MPM) designed to enable significantly larger time steps. To address the severe volume loss associated with this implicit scheme, we introduced a particle distancing force that is designed to maintains a uniform distribution of particles. Furthermore, to ensure physical accuracy and prevent volume loss in the implicit formulation, we derived a new deformation gradient update. This update guarantees consistency between the particles' kinematic movement, and the actual physical compression they undergo and their deformation gradient. Our evaluation of the particle distancing force revealed a notable behavioral trade-off highly dependent on the chosen penalty coefficient. Applying a sufficiently large coefficient successfully prevents volume loss; however, by heavily penalizing localized particle concentrations, it restricts particles from easily flowing past one another. Consequently, the fluid separates much more slowly and begins to exhibit cohesive,

sand-like dynamics. Utilizing a smaller coefficient allows the fluid to separate rapidly and naturally, though it results in some initial volume loss before the fluid settles into a stable state. We observed that when this particle distancing force is applied within an explicit framework, the volume loss present in the explicit scheme is also mitigated. A primary limitation of our proposed methodology is the substantial computational overhead introduced by the particle distancing force. Specifically, computing the Hessian of the particle distancing potential necessitates nine distinct state transfers between the particles and the background grid. While our Newton solver maintains a consistent convergence rate in terms of total iterations, the computational burden of these transfers makes each individual iteration exceedingly long. As a result, the overall execution time of the implicit method with the distancing force currently exceeds that of the standard explicit method. Future work will focus on optimizing these transfers and exploring alternative Hessian approximations to improve the computational efficiency of the implicit framework.

Chapter 5

Conclusion

The simulation of multiphase mixtures presents a significant challenge in computational physics, requiring numerical frameworks that can accurately resolve complex interfacial interactions and diverse separation mechanisms. The primary objective of this thesis was to develop robust, accurate, and efficient numerical methods to simulate multiphase fluid dynamics, addressing both chemical potential-driven and buoyancy-driven phase separation.

To achieve this, we addressed the mathematical and computational limitations of existing frameworks across three core contributions:

Conservative Discretization of the CHNS Equations (Chapter 2): We presented a novel, simplified discretization of the coupled Cahn-Hilliard-Navier-Stokes (CHNS) system to model phase separation in binary mixtures. Our approach ensures the strict conservation of both mass and momentum while maintaining consistency between the two. We demon-

strated that this method achieves second-order accuracy and is significantly simpler to solve computationally than traditional CHNS solvers.

Multi-Phase Material Point Method (Chapter 3): We addressed the inherent limitations of the standard Material Point Method (MPM) in simulating macroscopic, buoyancy-driven separation in immiscible fluids. Specifically, we solved the “sticking” artifact caused by the shared background grid, which artificially forces nearby particles of differing phases to move together. By introducing a scheme that updates particle velocities based on their distinct phase identities, we successfully simulated the natural separation of fluids in scenarios with fluids with different densities and in presence of gravity.

Implicit Time Integration for Multi-Phase MPM (Chapter 4): To improve the stability and computational efficiency of our multi-phase MPM framework, we developed an implicit time integration scheme. We show that the explicit multi-phase scheme allows the state of the system to be solved as an energy minimization problem. This implicit formulation permits the use of larger time steps, ensuring robust and efficient simulations for complex multiphase flows. We also introduced a particle distancing force to maintain a uniform distribution of particles and prevent volume loss. We have observed significant volume loss when simulating multi-phase simulations using the implicit scheme. We also discussed the potential causes and mitigation strategies for this issue.

In summary, the methods developed in this thesis provide a more accurate, stable, and computationally efficient toolkit for simulating multiphase mixtures. By overcoming the numerical constraints of traditional phase-field and particle-based methods, these ad-

vancements improve our ability to model both the microscopic thermodynamics of binary fluids and the macroscopic dynamics of immiscible fluid separation.

Bibliography

- [1] Keita Abe, Kenichi Soga, and Samila Bandara. Material point method for coupled hydromechanical problems. *Journal of Geotechnical and Geoenvironmental Engineering*, 140(3):04013033, 2014. doi: 10.1061/(ASCE)GT.1943-5606.0001011.
- [2] Rüyam Acar. Simulation of interface dynamics: a diffuse-interface model. *The Visual Computer*, 25(2):101–115, 2009.
- [3] S. Adami, X.Y. Hu, and N.A. Adams. A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 241:292–307, 2013. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2013.01.043>. URL <https://www.sciencedirect.com/science/article/pii/S002199911300096X>.
- [4] John D. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill, New York, 1995. ISBN 0-07-001685-2.
- [5] Tariq D Aslam and Eduardo Lozano. An efficient level set method for tracking many materials. *Journal of Computational Physics*, page 114757, 2026. stores one combined unsigned distance field and one integer indicator field, and then infers signed distance values for single level sets near interfaces during dynamics, ghost fluid type method.
- [6] Mehrnaz Ayazi, Craig Schroeder, and Tamar Shinar. Buoyancy-driven phase separation in the material point method. *ACM Transactions on Graphics*, 2026.
- [7] VE Badalassi and S Banerjee. Nano-structure computation with coupled momentum phase ordering kinetics models. *Nuclear engineering and design*, 235(10-12):1107–1115, 2005.
- [8] Vittorio E Badalassi, Hector D Cenicerros, and Sanjoy Banerjee. Computation of multiphase systems with phase field models. *Journal of computational physics*, 190(2):371–397, 2003.
- [9] SG. Bardenhagen and EM. Kober. The generalized interpolation material point method. *Comp Mod in Eng and Sci*, 5(6):477–496, 2004.
- [10] SG. Bardenhagen, JU. Brackbill, and D. Sulsky. The material-point method for granular materials. *Computer methods in applied mechanics and engineering*, 187(3):529–541, 2000.

- [11] John W Barrett, James F Blowey, and Harald Garcke. Finite element approximation of the cahn–hilliard equation with degenerate mobility. *SIAM Journal on Numerical Analysis*, 37(1):286–318, 1999.
- [12] Ted Belytschko, Wing Kam Liu, and Brian Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, New York, 2000.
- [13] James F Blowey, MIM Copetti, and Charles M Elliott. Numerical analysis of a model for phase separation of a multicomponent alloy. *IMA Journal of Numerical Analysis*, 16(1):111–139, 1996.
- [14] Jeremiah U Brackbill, Douglas B Kothe, and Charles Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992.
- [15] Christopher E. Brennen. *Fundamentals of Multiphase Flow*. Cambridge University Press, Cambridge, UK, 2005.
- [16] Markus Bussmann, Douglas B Kothe, and James M Sicilian. Modeling high density ratio incompressible interfacial flows. In *Fluids Engineering Division Summer Meeting*, volume 36150, pages 707–713, 2002.
- [17] John W. Cahn and John E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of Chemical Physics*, 28(2):258–267, 1958. doi: 10.1063/1.1744102. URL <https://doi.org/10.1063/1.1744102>.
- [18] Herbert B. Callen. *Thermodynamics and an Introduction to Thermostatistics*. John Wiley & Sons, New York, 2nd edition, 1985.
- [19] Francesca Ceccato, Lars Beuth, Pieter A Vermeer, and Paolo Simonini. Two-phase material point method applied to the study of cone penetration. *Computers and Geotechnics*, 80:440–452, 2016.
- [20] Kelong Cheng, Wenqiang Feng, Cheng Wang, and Steven M Wise. An energy stable fourth order finite difference scheme for the cahn–hilliard equation. *Journal of Computational and Applied Mathematics*, 362:574–595, 2019.
- [21] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- [22] Bart J. Daly. Numerical Study of Two Fluid Rayleigh-Taylor Instability. *The Physics of Fluids*, 10(2):297–307, 02 1967. ISSN 0031-9171. doi: 10.1063/1.1762109. URL <https://doi.org/10.1063/1.1762109>.
- [23] G. Daviet and F. Bertails-Descoubes. A semi-implicit material point method for the continuum simulation of granular materials. *ACM Trans Graph*, 35(4), jul 2016.
- [24] Hang Ding, Peter D.M. Spelt, and Chang Shu. Diffuse interface model for incompressible two-phase flows with large density ratios. *Journal of Computational Physics*, 226(2):2078–2095, 2007.

ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2007.06.028>. URL <https://www.sciencedirect.com/science/article/pii/S0021999107002793>.

- [25] Mengyuan Ding, Xuchen Han, Stephanie Wang, Theodore F Gast, and Joseph M Teran. A thermomechanical material point method for baking and cooking. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019.
- [26] S. Dong. On imposing dynamic contact-angle boundary conditions for wall-bounded liquid–gas flows. *Computer Methods in Applied Mechanics and Engineering*, 247–248: 179–200, 2012. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2012.07.023>. URL <https://www.sciencedirect.com/science/article/pii/S004578251200254X>.
- [27] Suchuan Dong and Jie Shen. A time-stepping scheme involving constant coefficient matrices for phase-field simulations of two-phase incompressible flows with large density ratios. *Journal of Computational Physics*, 231(17):5788–5804, 2012.
- [28] Charles M Elliott. The cahn-hilliard model for the kinetics of phase separation. In *Mathematical models for phase change problems*, pages 35–73. Springer, 1989.
- [29] Charles M Elliott and Donald A French. A nonconforming finite-element method for the two-dimensional cahn–hilliard equation. *SIAM Journal on Numerical Analysis*, 26(4):884–903, 1989.
- [30] Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. Silly rubber: an implicit material point method for simulating non-equilibrated viscoelastic and elastoplastic solids. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.
- [31] Yu Fang, Ziyin Qu, Minchen Li, Xinxin Zhang, Yixin Zhu, Mridul Aanjaneya, and Chenfanfu Jiang. Iq-mpm: an interface quadrature material point method for non-sticky strongly two-way coupled nonlinear solids and fluids. *ACM Transactions on Graphics (TOG)*, 39(4):51–1, 2020.
- [32] Xiaobing Feng and Andreas Prohl. Error analysis of a mixed finite element method for the cahn-hilliard equation. *Numerische Mathematik*, 99:47–84, 2004.
- [33] Marianne M. Francois, Sharen J. Cummins, Edward D. Dendy, Douglas B. Kothe, James M. Sicilian, and Matthew W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, March 2006. ISSN 00219991. doi: [10.1016/j.jcp.2005.08.004](https://doi.org/10.1016/j.jcp.2005.08.004). URL <https://linkinghub.elsevier.com/retrieve/pii/S0021999105003748>.
- [34] C. Fu, Q. Guo, T. Gast, C. Jiang, and J. Teran. A polynomial particle-in-cell method. *ACM Transactions on Graphics (TOG)*, 36(6):222, 2017.
- [35] Daisuke Furihata. A stable and conservative finite difference scheme for the cahn-hilliard equation. *Numerische Mathematik*, 87(4):675–699, 2001.

- [36] M. Gao, A. Pradhana, X. Han, Q. Guo, G. Kot, E. Sifakis, and C. Jiang. Animating fluid sediment mixture in particle-laden flows. *ACM Transactions on Graphics (TOG)*, 37(4):149, 2018.
- [37] Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. An adaptive generalized interpolation material point method for simulating elastoplastic materials. *ACM Trans. Graph.*, 36(6), November 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130879. URL <https://doi.org/10.1145/3130800.3130879>.
- [38] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 313–320, 1993. deformable object representation, corrects overlaps only.
- [39] T. Gast and C. Schroeder. Optimization integrator for large time steps. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 31–40, 2014.
- [40] T. Gast, C. Schroeder, A. Stomakhin, C. Jiang, and J. Teran. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics*, 21(10):1103–1115, 2015.
- [41] Jason Goulding, Mehrnaz Ayazi, Tamar Shinar, and Craig Schroeder. Simplified conservative discretization of the cahn-hilliard-navier-stokes equations. *Journal of Computational Physics*, 519:113382, 2024. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2024.113382>. URL <https://www.sciencedirect.com/science/article/pii/S0021999124006302>.
- [42] J.-L. Guermond and L. Quartapelle. A projection fem for variable density incompressible flows. *Journal of Computational Physics*, 165(1):167–188, 2000. ISSN 0021-9991. doi: <https://doi.org/10.1006/jcph.2000.6609>. URL <https://www.sciencedirect.com/science/article/pii/S0021999100966099>.
- [43] Francisco Guillén-González and Giordano Tierra. On linear schemes for a cahn–hilliard diffuse interface model. *Journal of Computational Physics*, 234:140–171, 2013.
- [44] Francisco Guillén-González and Giordano Tierra. Second order schemes and time-step adaptivity for allen–cahn and cahn–hilliard models. *Computers & Mathematics with Applications*, 68(8):821–846, 2014.
- [45] Q. Guo, X. Han, C. Fu, T. Gast, R. Tamstorf, and J. Teran. A material point method for thin shells with frictional contact. *ACM Transactions on Graphics (TOG)*, 37(4): 147, 2018.
- [46] Zhenlin Guo, Ping Lin, J Lowengrub, and Steven M Wise. Mass conservative and energy stable finite difference methods for the quasi-incompressible navier–stokes–cahn–hilliard system: Primitive variable and projection-type schemes. *Computer Methods in Applied Mechanics and Engineering*, 326:144–174, 2017.

- [47] Morton E Gurtin, Debra Polignone, and Jorge Vinals. Two-phase binary fluids and immiscible fluids described by an order parameter. *Mathematical Models and Methods in Applied Sciences*, 6(06):815–831, 1996.
- [48] F. Harlow and E. Welch. Numerical calculation of time dependent viscous flow of fluid with a free surface. *Phys Fluid*, 8(12):2182–2189, 1965.
- [49] Qiaolin He, Roland Glowinski, and Xiao-Ping Wang. A least-squares/finite element method for the numerical solution of the navier–stokes-cahn–hilliard system modeling the motion of the contact line. *Journal of computational physics*, 230(12):4991–5009, 2011.
- [50] Qunwu He and Nobuhide Kasagi. Phase-field simulation of small capillary-number two-phase flow in a microtube. *Fluid dynamics research*, 40(7-8):497, 2008.
- [51] Yinnian He, Yunxian Liu, and Tao Tang. On large time-stepping methods for the cahn–hilliard equation. *Applied Numerical Mathematics*, 57(5-7):616–628, 2007.
- [52] Pierre C Hohenberg and Bertrand I Halperin. Theory of dynamic critical phenomena. *Reviews of Modern Physics*, 49(3):435, 1977.
- [53] Michael A Homel and Eric B Herbold. Field-gradient partitioning for fracture and frictional contact in the material point method. *International Journal for Numerical Methods in Engineering*, 109(7):1013–1044, 2017.
- [54] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201293. URL <https://doi.org/10.1145/3197517.3201293>.
- [55] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4):150, 2018.
- [56] Ziyang Huang, Guang Lin, and Arezoo M Ardekani. Consistent and conservative scheme for incompressible two-phase flows using the conservative allen-cahn model. *Journal of Computational Physics*, 420:109718, 2020.
- [57] Ziyang Huang, Guang Lin, and Arezoo M. Ardekani. Consistent, essentially conservative and balanced-force phase-field method to model incompressible two-phase flows. *Journal of Computational Physics*, 406:109192, 2020. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2019.109192>. URL <https://www.sciencedirect.com/science/article/pii/S0021999119308976>.
- [58] Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. Sph fluids in computer graphics. 2014.

- [59] Mamoru Ishii and Takashi Hibiki. *Thermo-Fluid Dynamics of Two-Phase Flow*. Springer, New York, 2nd edition, 2011.
- [60] Jacob N. Israelachvili. *Intermolecular and Surface Forces*. Academic Press, Burlington, MA, 3rd edition, 2011.
- [61] David Jacqmin. Calculation of two-phase navier–stokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96–127, 1999. ISSN 0021-9991. doi: <https://doi.org/10.1006/jcph.1999.6332>. URL <https://www.sciencedirect.com/science/article/pii/S0021999199963325>.
- [62] David Jacqmin. Calculation of two-phase navier–stokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96–127, 1999. ISSN 0021-9991. doi: <https://doi.org/10.1006/jcph.1999.6332>. URL <https://www.sciencedirect.com/science/article/pii/S0021999199963325>.
- [63] Issam Jassim, Dieter Stolle, and Pieter Vermeer. Two-phase dynamic analysis by material point method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(15):2502–2522, October 2012. ISSN 1096-9853. doi: 10.1002/nag.2146. URL <http://dx.doi.org/10.1002/nag.2146>.
- [64] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. The affine particle-in-cell method. *ACM Trans Graph*, 34(4):51:1–51:10, 2015.
- [65] Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *ACM siggraph 2016 courses*, pages 1–52. 2016.
- [66] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996.
- [67] Myungjoo Kang, Ronald P Fedkiw, and Xu-Dong Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of scientific computing*, 15: 323–360, 2000.
- [68] Junseok Kim. A continuous surface tension force formulation for diffuse-interface models. *Journal of computational physics*, 204(2):784–804, 2005.
- [69] Junseok Kim. Phase field computations for ternary fluid flows. *Computer methods in applied mechanics and engineering*, 196(45-48):4779–4788, 2007.
- [70] Junseok Kim. A generalized continuous surface tension force formulation for phase-field models for multi-component immiscible fluid flows. *Computer Methods in Applied Mechanics and Engineering*, 198(37-40):3105–3112, 2009.
- [71] Junseok Kim. Phase-field models for multi-component fluid flows. *Communications in Computational Physics*, 12(3):613–661, 2012.
- [72] Junseok Kim and John Lowengrub. Phase field modeling and simulation of three-phase flows. *Interfaces and free boundaries*, 7(4):435–466, 2005.

- [73] G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and T. Teran. Drucker-prager elastoplasticity for sand animation. *ACM Transactions on Graphics (SIGGRAPH 2016)*., 2016.
- [74] Tassilo Kugelstadt, Andreas Longva, Nils Thuerey, and Jan Bender. Implicit density projection for volume conserving liquids. *IEEE Transactions on Visualization and Computer Graphics*, 27(4):2385–2395, 2019.
- [75] Benjamin Lallanne, Lucia Rueda Villegas, Sébastien Tanguy, and Frédéric Risso. On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method. *Journal of Computational Physics*, 301:289–307, 2015.
- [76] Dongsun Lee, Joo-Youl Huh, Darae Jeong, Jaemin Shin, Ana Yun, and Junseok Kim. Physical, mathematical, and numerical derivations of the cahn–hilliard equation. *Computational Materials Science*, 81:216–225, 2014.
- [77] Bo Li and Shiguang Liu. A multi-species material point method with a mixture model. *Computer Animation and Virtual Worlds*, 35, 05 2024. doi: 10.1002/cav.2239.
- [78] Wei Li, Kui Wu, and Mathieu Desbrun. Kinetic simulation of turbulent multifluid flows. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024.
- [79] Yibao Li, Hyun Geun Lee, Binhu Xia, and Junseok Kim. A compact fourth-order finite difference scheme for the three-dimensional cahn–hilliard equation. *Computer Physics Communications*, 200:108–116, 2016.
- [80] Yibao Li, Yongho Choi, and Junseok Kim. Computationally efficient adaptive time step method for the cahn–hilliard equation. *Computers & Mathematics with Applications*, 73(8):1855–1864, 2017.
- [81] S.J. Lind, R. Xu, P.K. Stansby, and B.D. Rogers. Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4):1499–1523, 2012. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2011.10.027>. URL <https://www.sciencedirect.com/science/article/pii/S0021999111006279>.
- [82] Chun Liu and Jie Shen. A phase field model for the mixture of two incompressible fluids and its approximation by a fourier-spectral method. *Physica D: Nonlinear Phenomena*, 179(3-4):211–228, 2003.
- [83] Haihu Liu and Yonghao Zhang. Phase-field modeling droplet dynamics with soluble surfactants. *Journal of Computational Physics*, 229(24):9166–9187, 2010.
- [84] Shiguang Liu, Qiguang Liu, and Qunsheng Peng. Realistic simulation of mixing fluids. *The Visual Computer*, 27(3):241–248, 2011.

- [85] Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. Multiple interacting liquids. *ACM Trans. Graph.*, 25:812–819, July 2006. ISSN 0730-0301. doi: <http://doi.acm.org/10.1145/1141911.1141960>. URL <http://doi.acm.org/10.1145/1141911.1141960>.
- [86] Zhen Luo, Liyong Tong, Junzhao Luo, Peng Wei, and Michael Yu Wang. Design of piezoelectric actuators using a multiphase level set method of piecewise constants. *Journal of computational physics*, 228(7):2643–2659, 2009. topology optimization method for multi-material structural design, n level sets for 2 to the n regions.
- [87] P. Mackenzie-Helnwein, P. Arduino, W. Shin, J. A. Moore, and G. R. Miller. Modeling strategies for multiphase drag interactions using the material point method. *International Journal for Numerical Methods in Engineering*, 83(3):295–322, June 2010. ISSN 1097-0207. doi: 10.1002/nme.2823. URL <http://dx.doi.org/10.1002/nme.2823>.
- [88] M. Macklin, M. Müller, N. Chentanez, and T. Kim. Unified particle physics for real-time applications. *ACM Trans Graph*, 33(4):153:1–153:12, 2014.
- [89] Miles Macklin and Matthias Müller. Position based fluids. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.
- [90] JC Martin and WJ Moyce. An experimental study of the collapse of fluid columns on a rigid horizontal plane, in a medium of lower, but comparable, density. 5. *Philosophical Transactions of the Royal Society of London Series A-Mathematical and Physical Sciences*, 244(882):325–334, 1952.
- [91] Barry Merriman, James K Bence, and Stanley J Osher. Motion of multiple junctions: A level set approach. *Journal of computational physics*, 112(2):334–363, 1994. levelset function for each phase with projection similar to losasso but breaks signed distance property.
- [92] Manninen Mikko, Taivassalo Veikko, and Kallio Sirpa. On the mixture model for multiphase flow. *Vtt Publications*, 288:3–67, 1996.
- [93] J.J. Monaghan. Sph without a tensile instability. *Journal of Computational Physics*, 159(2):290–311, 2000. ISSN 0021-9991. doi: <https://doi.org/10.1006/jcph.2000.6439>. URL <https://www.sciencedirect.com/science/article/pii/S0021999100964398>.
- [94] Matthias Müller, Barbara Solenthaler, Richard Keiser, and Markus Gross. Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIG-GRAPH/Eurographics Symposium on Computer Animation*, SCA '05, page 237–244, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931988. doi: 10.1145/1073368.1073402. URL <https://doi.org/10.1145/1073368.1073402>.
- [95] John A Nairn. Material point method calculations with explicit cracks. *Computer Modeling in Engineering and Sciences*, 4(6):649–664, 2003.

- [96] John A Nairn, Chad C Hammerquist, and Grant D Smith. New material point method contact algorithms for improved accuracy, large-deformation problems, and proper null-space filtering. *Computer Methods in Applied Mechanics and Engineering*, 362:112859, 2020.
- [97] Michael B Nielsen and Ole Østerby. A two-continua approach to eulerian simulation of water spray. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.
- [98] Alexander Maximilian Nilles and Stefan Müller. A moving least squares material point method for varied porous material interactions and non-sticky coupling of phases, 2022.
- [99] Masaki Noda, Yuki Noguchi, and Takayuki Yamada. Extended level set method: A multiphase representation with perfect symmetric property, and its application to multi-material topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 393:114742, 2022. topology optimization method for multi-material structural design.
- [100] Mark Owkes and Olivier Desjardins. A mass and momentum conserving unsplit semi-lagrangian framework for simulating multiphase flows. *Journal of Computational Physics*, 332:21–46, 2017.
- [101] Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. An implicit viscosity formulation for sph fluids. *ACM Transactions on Graphics (TOG)*, 34(4): 1–10, 2015.
- [102] Mehdi Raessi and Heinz Pitsch. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. *Computers & Fluids*, 63:70–81, 2012.
- [103] D. Ram, T. Gast, C. Jiang, C. Schroeder, A. Stomakhin, J. Teran, and P. Kavehpour. A material point method for viscoelastic fluids, foams and sponges. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 157–163. ACM, 2015.
- [104] Rayleigh. Investigation of the Character of the Equilibrium of an Incompressible Heavy Fluid of Variable Density*. *Proceedings of the London Mathematical Society*, s1-14(1):170–177, 11 1882. ISSN 0024-6115. doi: 10.1112/plms/s1-14.1.170. URL <https://doi.org/10.1112/plms/s1-14.1.170>.
- [105] Bo Ren, Chenfeng Li, Xiao Yan, Ming C. Lin, Javier Bonet, and Shi-Min Hu. Multiple-fluid sph simulation using a mixture model. *ACM Trans. Graph.*, 33(5), September 2014. ISSN 0730-0301. doi: 10.1145/2645703. URL <https://doi.org/10.1145/2645703>.
- [106] Bo Ren, Xu-Yun Yang, Ming C. Lin, Nils Thuerey, Matthias Teschner, and Chenfeng Li. Visual simulation of multiple fluids in computer graphics: A state-of-the-art report. *Journal of Computer Science and Technology*, 33(3):431–451, 2018.

ISSN 1000-9000(Print) /1860-4749(Online). doi: 10.1007/s11390-018-1829-0. URL <https://jcst.ict.ac.cn/en/article/doi/10.1007/s11390-018-1829-0>.

- [107] Steven J Ruuth. A diffusion-generated approach to multiphase motion. *Journal of Computational Physics*, 145(1):166–192, 1998. n level set functions for n regions, dynamics retain valid state and don’t require projection.
- [108] Somnath Santra, Shubhadeep Mandal, and Suman Chakraborty. Phase-field modeling of multicomponent and multiphase flows in microfluidic systems: A review. *International Journal of Numerical Methods for Heat & Fluid Flow*, 31(10):3089–3131, 2020.
- [109] Robert I Saye and James A Sethian. The voronoi implicit interface method for computing multiphase physics. *Proceedings of the National Academy of Sciences*, 108(49):19498–19503, 2011. uses a single function, Voronoi reconstruction after evolution step.
- [110] Hagit Schechter and Robert Bridson. Ghost sph for animating water. *ACM Trans. Graph.*, 31(4), July 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185557. URL <https://doi.org/10.1145/2185520.2185557>.
- [111] K.M. Roessig J.U. Brackbill W.M. Witzel J.C.Foster S.G. Bardenhagen, J.E. Guilkey. An improved contact algorithm for the material point method and application to stress propagation in granular material. *Computer Modeling in Engineering & Sciences*, 2(4):509–522, 2001. ISSN 1526-1506. doi: 10.3970/cmes.2001.002.509. URL <http://www.techscience.com/CMES/v2n4/24749>.
- [112] Jie Shen and Xiaofeng Yang. Energy stable schemes for cahn-hilliard phase-field model of two-phase incompressible flows. *Chinese Annals of Mathematics, Series B*, 31(5):743–758, 2010.
- [113] Jie Shen and Xiaofeng Yang. A phase-field model and its numerical approximation for two-phase incompressible flows with different densities and viscosities. *SIAM Journal on Scientific Computing*, 32(3):1159–1179, 2010.
- [114] Jie Shen and Xiaofeng Yang. Decoupled, energy stable schemes for phase-field models of two-phase incompressible flows. *SIAM Journal on Numerical Analysis*, 53(1):279–296, 2015.
- [115] Kurt A Smith, Francisco J Solis, and David Chopp. A projection method for motion of triple junctions by level sets. *Interfaces and free boundaries*, 4(3):263–276, 2002. symmetric projection of n level set functions onto n-1 dimensional manifold.
- [116] Barbara Solenthaler and Renato Pajarola. Density contrast sph interfaces. 2008.
- [117] M. Steffen, R. Kirby, and M. Berzins. Analysis and reduction of quadrature errors in the material point method (MPM). *Int J Numer Meth Eng*, 76(6):922–948, 2008.
- [118] Michael Steffen, Robert M. Kirby, and Martin Berzins. Analysis and reduction of quadrature errors in the material point method

- (mpm). *International Journal for Numerical Methods in Engineering*, 76(6):922–948, 2008. doi: <https://doi.org/10.1002/nme.2360>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2360>.
- [119] A. Stomakhin, R. Howes, C. Schroeder, and J.M. Teran. Energetically consistent invertible elasticity. In *Proc. Symp. Comp. Anim.*, pages 25–32, 2012.
 - [120] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. A material point method for snow simulation. In *ACM Transactions on Graphics (SIGGRAPH 2013)*, pages 102:1–10, 2013.
 - [121] A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle. Augmented mpm for phase-change and varied materials. *ACM Transactions on Graphics (SIGGRAPH 2014)*, pages 1–11, 2014.
 - [122] Alexey Stomakhin, Craig Schroeder, Lingjie Chai, Joseph Teran, and Andrew Selle. Material point method for snow simulation. *ACM Transactions on Graphics (SIGGRAPH)*, 32(4):102:1–10, 2013.
 - [123] D. Sulsky, Z. Chen, and H. L. Schreyer. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994.
 - [124] D. Sulsky, S. Zhou, and H. Schreyer. Application of a particle-in-cell method to solid mechanics. *Comp Phys Comm*, 87(1):236–252, 1995.
 - [125] Yunxin Sun, Tamar Shinar, and Craig Schroeder. Effective time step restrictions for explicit mpm simulation. In *Computer Graphics Forum*, volume 39, pages 55–67. Wiley Online Library, 2020.
 - [126] J.W. Swegle, D.L. Hicks, and S.W. Attaway. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116(1):123–134, 1995. ISSN 0021-9991. doi: <https://doi.org/10.1006/jcph.1995.1010>. URL <https://www.sciencedirect.com/science/article/pii/S0021999185710108>.
 - [127] AP. Tampubolon, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth. Multi-species simulation of porous sand and water mixtures. *ACM Transactions on Graphics (SIGGRAPH 2017)*, 36(4):105, 2017.
 - [128] Knut Erik Teigen, Peng Song, John Lowengrub, and Axel Voigt. A diffuse-interface method for two-phase flows with soluble surfactants. *Journal of computational physics*, 230(2):375–393, 2011.
 - [129] Grétar Tryggvason. Numerical simulations of the rayleigh-taylor instability. *Journal of Computational Physics*, 75(2):253–282, 1988. ISSN 0021-9991. doi: [https://doi.org/10.1016/0021-9991\(88\)90112-X](https://doi.org/10.1016/0021-9991(88)90112-X). URL <https://www.sciencedirect.com/science/article/pii/002199918890112X>.

- [130] Zaili Tu, Chen Li, Zipeng Zhao, Long Liu, Chenhui Wang, Changbo Wang, and Hong Qin. A unified mpm framework supporting phase-field models and elastic-viscoplastic phase transition. *ACM Trans. Graph.*, 43(2), January 2024. ISSN 0730-0301. doi: 10.1145/3638047. URL <https://doi.org/10.1145/3638047>.
- [131] Luminita A Vese and Tony F Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision*, 50(3):271–293, 2002. n values for 2 to the n regions, variational approach.
- [132] Bao-Shan Wang, Peng Li, Zhen Gao, and Wai Sun Don. An improved fifth order alternative weno-z finite difference scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 374:469–477, 2018.
- [133] Xinlei Wang, Yuxing Qiu, Stuart R. Slattery, Yu Fang, Minchen Li, Song-Chun Zhu, Yixin Zhu, Min Tang, Dinesh Manocha, and Chenfanfu Jiang. A massively parallel and scalable multi-gpu material point method. *ACM Trans. Graph.*, 39(4), August 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392442. URL <https://doi.org/10.1145/3386569.3392442>.
- [134] Peng Wei and Michael Yu Wang. Piecewise constant level set method for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 78(4):379–402, 2009. topology optimization method for multi-material structural design.
- [135] Garth N Wells, Ellen Kuhl, and Krishna Garikipati. A discontinuous galerkin method for the cahn–hilliard equation. *Journal of Computational Physics*, 218(2):860–877, 2006.
- [136] Steven M Wise, John S Lowengrub, Hermann B Frieboes, and Vittorio Cristini. Three-dimensional multispecies nonlinear tumor growth—i: model and numerical method. *Journal of theoretical biology*, 253(3):524–543, 2008.
- [137] Joshua Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. Cd-mpm: continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.
- [138] Joel Wretborn, Rickard Armiento, and Ken Museth. Animation of crack propagation by means of an extended multi-body solver for the material point method. *Comput. Graph.*, 69(C):131–139, December 2017. ISSN 0097-8493. doi: 10.1016/j.cag.2017.10.005. URL <https://doi.org/10.1016/j.cag.2017.10.005>.
- [139] Qing Xia, Junxiang Yang, and Yibao Li. On the conservative phase-field method with the n-component incompressible flows. *Physics of Fluids*, 35(1), 2023.
- [140] X. Yan, C-F. Li, X-S. Chen, and S-M. Hu. Mpm simulation of interacting fluids and solids. *Computer Graphics Forum*, 37(8):183–193, September 2018. ISSN 1467-8659. doi: 10.1111/cgf.13523. URL <http://dx.doi.org/10.1111/cgf.13523>.

- [141] Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R Martin, and Shi-Min Hu. Multi-phase sph simulation for interactive fluids and solids. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016.
- [142] Tao Yang, Jian Chang, Bo Ren, Ming C Lin, Jian Jun Zhang, and Shi-Min Hu. Fast multiple-fluid simulation using helmholtz free energy. *ACM Transactions on Graphics (TOG)*, 34(6):1–11, 2015.
- [143] Tao Yang, Jian Chang, Bo Ren, Ming C. Lin, Jian Jun Zhang, and Shi-Min Hu. Fast multiple-fluid simulation using helmholtz free energy. *ACM Trans. Graph.*, 34(6), nov 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818117. URL <https://doi.org/10.1145/2816795.2818117>.
- [144] Tao Yang, Jian Chang, Ming C Lin, Ralph R Martin, Jian J Zhang, and Shi-Min Hu. A unified particle system framework for multi-phase, multi-material visual simulations. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.
- [145] Xiaofeng Yang, James J Feng, Chun Liu, and Jie Shen. Numerical simulations of jet pinching-off and drop formation using an energetic variational phase-field method. *Journal of Computational Physics*, 218(1):417–428, 2006.
- [146] Pengtao Yue, James J Feng, Chun Liu, and Jie Shen. A diffuse-interface method for simulating two-phase flows of complex fluids. *Journal of Fluid Mechanics*, 515: 293–317, 2004.
- [147] Pengtao Yue, James J Feng, Chun Liu, and Jie Shen. Diffuse-interface simulations of drop coalescence and retraction in viscoelastic fluids. *Journal of Non-Newtonian Fluid Mechanics*, 129(3):163–176, 2005.
- [148] Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun. Continuum foam: a material point method for shear-dependent flows. *ACM Trans Graph*, 34(5):160:1–160:20, 2015.
- [149] Chen Zhao, Tamar Shinar, and Craig Schroeder. Reconstruction of implicit surfaces from fluid particles using convolutional neural networks. In *Computer Graphics Forum*, volume 43, page e15181. Wiley Online Library, 2024.
- [150] Chen Zhao, Tamar Shinar, and Craig Schroeder. Fast reconstruction of implicit surfaces using convolutional neural networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 8(4):1–15, 2025.
- [151] Kai-Li Zhao, Liu chao Qiu, and Yi Liu. Two-phase mpm modeling of dry granular fronts and watery tails formed in debris flows. *Communications in Nonlinear Science and Numerical Simulation*, 128:107624, 2024. ISSN 1007-5704. doi: <https://doi.org/10.1016/j.cnsns.2023.107624>. URL <https://www.sciencedirect.com/science/article/pii/S1007570423005452>.
- [152] Guangpu Zhu, Jisheng Kou, Shuyu Sun, Jun Yao, and Aifen Li. Numerical approximation of a phase-field surfactant model with fluid flow. *Journal of Scientific Computing*, 80:223–247, 2019.

- [153] Jingzhi Zhu, Long-Qing Chen, Jie Shen, and Veena Tikare. Coarsening kinetics from a variable-mobility cahn-hilliard equation: Application of a semi-implicit fourier spectral method. *Physical Review E*, 60(4):3564, 1999.