Title

Advances in Volume Penalization Methods for Simulating Multiphase Fluid-Structure Interaction and Phase-Change Phenomena

Author

Thirumalaisamy, Ramakrishnan

Publication Date

2024

UNIVERSITY OF CALIFORNIA SAN DIEGO

SAN DIEGO STATE UNIVERSITY

Advances in Volume Penalization Methods for Simulating Multiphase Fluid-Structure Interaction and Phase-Change Phenomena

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical and Aerospace Engineering)

by

Ramakrishnan Thirumalaisamy

Committee in charge:

University of California San Diego:

Professor David Saintillan, Co-Chair
Professor Alicia Kim
Professor Abhishek Saha

San Diego State University:

Professor Amneet Pal Singh Bhalla, Co-Chair
Professor Gustaaf Jacobs

2024

The dissertation of Ramakrishnan Thirumalaisamy is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

Co-chair

Co-chair

University of California San Diego

San Diego State University

2024

DEDICATION

Dedicated to my Amma and Appa!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am grateful to everyone who supported me throughout my dissertation journey with their valuable comments and ideas.

A special thanks to my advisor, Prof. Amneet Pal Singh Bhalla, for providing me with the invaluable opportunity to collaborate on this research. His unwavering support, guidance, and encouragement were instrumental in completing this thesis. Prof. Bhalla patiently taught me coding practices, software development, and GitHub usage during my early Ph.D. years. His expertise in fluid-structure problems accelerated my research progress in the beginning. His feedback on my writing enhanced my communication skills and improved the effectiveness of my research. Throughout my studies, Prof. Bhalla has been available for insightful discussions and has shared innovative ideas to enhance our work. His constructive criticism has been invaluable in developing important technical and problem-solving skills. The passion he carries for research is inspiring. I have gained valuable technical skills that will benefit me throughout my career. I look forward to continuing our collaboration for many years to come. Prof. Bhalla has also provided valuable career advice, showing genuine care for my professional development. He has been a constant presence in every part of my Ph.D. journey. Thank you, Prof. Bhalla !

I extend my heartfelt gratitude to the committee members Prof. David Saintillan, Prof. Gustaaf Jacobs, Prof. Abhishek Saha, and Prof. Alicia H. Kim for their insightful suggestions and guidance on my research. Special thanks to Prof. David Saintillan for serving as my co-chair and for the valuable input provided during our yearly review meetings. I am also thankful to all the professors for their impressive courses, which greatly enriched my academic experience. A special acknowledgment goes to the IBAMR developers for their extensive code reviews and invaluable suggestions that enhanced the quality of my work. Additionally, I would like to express my appreciation to everyone I collaborated with during my Ph.D. journey. I am especially grateful to Pieter Ghysels (LBNL) and Hiroshi Otomo (Dassault Systemes) for providing me with internship opportunities. I would like to thank Prof. Ganesh Natarajan (IIT Palakkad) and Prof. Amaresh Dalal (IIT Guwahati) for their encouragement and support that motivated me to pursue research in the first place.

I express my gratitude to San Diego State University (SDSU) startup fund and the National Science Foundation awards OAC 1931368 and CBET, United States CAREER 2234387 (awarded to Prof. Bhalla) for supporting my research. Additionally, I am thankful to San Diego State University for providing me with a university graduate fellowship during my second year, which covered my stipend and health insurance for the year.

I am grateful to my colleague and good friend, Kaustubh Khedkar, who has been by my side for the past 7 years. Our late-night discussions about research, career, and personal life have been truly memorable. From studying for exams together to conducting research and traveling through life's ups and downs, I deeply appreciate our journey together. Thank you, my brother.

I am fortunate to have made great friends in San Diego. I cherish the countless lifetime memories we have created together, although it's difficult to name each one individually. To all my friends in San Diego, thank you. Cricket has been my most beloved sport. I extend my gratitude to the SD Warriors cricket team. Winning competitions and lifting trophies with the team are cherished moments I will always remember. I also want to express my heartfelt gratitude to all my friends from my childhood through my master's program at IIT Guwahati, as well as my relatives. Special thanks to my friend Keerthi Mohan for his constant encouragement and support.

Finally, I owe everything to my parents. Without them, I couldn't have accomplished anything. They allowed me to make choices in life. For all the hard work they have put in, I hope I have made them proud. Being the first generation student and receiving a Ph.D., especially from a prestigious universities like SDSU and UCSD, should make them proud.

Chapter 2, in part, is a reprint of the material as it appears in Critique on "Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry". Thirumalaisamy, Ramakrishnan; Nangia, Nishant; Bhalla, Amneet Pal Singh, Journal of Computational Physics, vol. 433, 110163, 2021. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 2, in part, is a reprint of the material as it appears in Handling Neumann and Robin boundary conditions in a fictitious domain volume penalization framework. Thirumalaisamy, Ramakrishnan; Patankar, Neelesh A.; Bhalla, Amneet Pal Singh, Journal of Computational Physics, vol. 448, 110726, 2022. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in An effective preconditioning strategy for volume penalized incompressible/low Mach multiphase flow solvers. Thirumalaisamy, Ramakrishnan; Khedkar, Kaustubh; Ghysels, Pieter; Bhalla, Amneet Pal Singh, Journal of Computational Physics, vol. 490, 112325, 2023. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in A low Mach enthalpy method to model non-isothermal gas–liquid–solid flows with melting and solidification. Thirumalaisamy, Ramakrishnan; Bhalla, Amneet Pal Singh, International Journal of Multiphase Flow, vol. 169, 104605, 2023. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4, in part is currently being prepared for submission for publication of the material. Thirumalaisamy, Ramakrishnan; Bhalla, Amneet Pal Singh. The dissertation/thesis author was the primary investigator and author of this paper.

VITA

2010–2013	Bachelor of Engineering, Mechanical Engineering, Dr. Mahalingam College of Engineering and Technology, Anna University, India.
2015–2017	Master of Technology, Mechanical Engineering (Fluids and Thermal Engineering), Indian Institute of Technology (IIT) Guwahati, India.
2017–2019	Senior Research Fellow, AnuPravaha Laboratory, Department of Mechanical Engineering, Indian Institute of Technology (IIT) Guwahati, India.
2022	Summer internship at Lawrence Berkeley National Laboratory, Berkeley.
2023	Software engineer Internship at Dasaault Systemes, MA.
2023	Summer internship at Lawrence Berkeley National Laboratory, Berkeley.
2019–2024	Doctor of Philosophy in Engineering Sciences (Mechanical and Aerospace Engineering), San Diego State University and University of California, San Diego.

PUBLICATIONS

1. **Ramakrishnan Thirumalaisamy**, Amneet Pal Singh Bhalla (2023), A low Mach enthalpy method to model non-isothermal gas–liquid–solid flows with melting and solidification, International Journal of Multiphase Flow, vol. 169, 104605.

2. **Ramakrishnan Thirumalaisamy**, Kaustubh Khedkar, Pieter Ghysels, Amneet Pal Singh Bhalla (2023), An effective preconditioning strategy for volume penalized incompressible/low Mach multiphase flow solvers, Journal of Computational Physics, Vol. 490, 112325.

3. **Ramakrishnan Thirumalaisamy**, Neelesh A. Patankar, Amneet Pal Singh Bhalla (2022), Handling Neumann and Robin boundary conditions in a fictitious domain volume penalization framework, Journal of Computational Physics, Vol. 448, 110726.

4. **Ramakrishnan Thirumalaisamy**, Nishant Nangia, Amneet Pal Singh Bhalla (2021), Critique on Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry, Journal of Computational Physics, Vol. 433, 110163.

5. **Ramakrishnan Thirumalaisamy**, Amneet Pal Singh Bhalla (2024), A consistent, volume preserving, and adaptive mesh refinement-based framework for modeling non-isothermal gas-liquid-solid flows, under preparation.

ABSTRACT OF THE DISSERTATION

Advances in Volume Penalization Methods for Simulating Multiphase Fluid-Structure Interaction and Phase-Change Phenomena

by

Ramakrishnan Thirumalaisamy

Doctor of Philosophy in Engineering Sciences (Mechanical and Aerospace Engineering)

San Diego State University, 2024
University of California San Diego, 2024

Professor David Saintillan, Co-Chair
Professor Amneet Pal Singh Bhalla, Co-Chair

The volume penalization method (VP), a type of Fictitious Domain Method, is a widely used technique for solving partial differential equations (PDEs) in complex domains. Its applications span various fields, from fluid-structure interactions like wave energy converters, bird and insect flight, fish swimming, and cardiovascular flows, to phase change applications such as glacier melting and additive manufacturing processes. This thesis presents robust and adaptive VP techniques for simulating non-isothermal phase-changing flows, as well as isothermal multiphase fluid-structure interaction problems. Using the numerically constructed flux-forcing functions for arbitrarily complex boundaries, we extend the flux-based volume penalization (VP) method to handle more general boundary conditions, including spatially varying inhomogeneous Neumann and Robin boundary conditions. Several two- and three-dimensional test examples, including flux-driven thermal convection in a concentric annular domain, are considered to assess the spatial accuracy of the numerical solutions. In addition, we propose a projection method-based preconditioning strategy for solving VP incompressible and low-Mach Navier-Stokes equations. The solver converges faster as the penalty coefficient decreases, contrary to prior experience. The developed preconditioning strategy is used in a novel low Mach enthalpy method to solve solidification and melting problems with variable thermophysical properties, including density. The proposed method captures the density change-induced flow during phase change material (PCM) melting and solidification. A gas phase is also incorporated and coupled to the solid-liquid PCM region in this formulation. The new low Mach enthalpy method is validated against analytical solutions for a PCM undergoing a large density change during its phase transition. Furthermore, we propose a set of simple sanity checks to serve as benchmarks for evaluating computational fluid dynamics (CFD) algorithms that aim to capture the volume change effects of PCMs. Adaptive mesh refinement is employed to achieve fine grid resolution in domains requiring more accuracy, such as PCM-gas and liquid-solid interfaces.

Chapter 1

Introduction

Since its introduction in the 1960s, computational fluid dynamics (CFD) has become a primary tool with widespread applications across academia and industry. Initially, CFD was predominantly used in traditional sectors such as automotive, aerospace, and power generation. However, over the years, significant research conducted by academia and industry has enhanced and popularized CFD techniques, leading to its expansion into cutting-edge technologies and processes. For example, CFD has improved the design of electronic chips through thermal analysis and cooling strategies. Additionally, CFD plays a critical role in advanced manufacturing processes such as 3D printing by predicting part properties and performance [5]. Other notable cutting-edge applications include battery simulation with CFD and optimization of Heating, Ventilation, and Air Conditioning (HVAC) systems.

Fluid-structure interaction (FSI) involves the interplay between a structure and the fluid surrounding it. Fluid forces induce movement in the structure, leading to changes in fluid flow. FSI has numerous applications across various scales, including the flow of blood in the heart and circulation, the flight of birds and insects, and the dispersal of seeds and pollen by wind [6]. Additionally, FSI plays a crucial role in engineering applications such as wave energy converters and ship design. Simulating FSI applications poses a challenge due to the potential for significant deformations and displacements at the interface. Therefore, body-fitted methods require frequent remeshing and incur high computational costs for solving linear systems of equations. To overcome the limitations of body-fitted grid methods, fictitious domain (FD) methods have been proposed. In fictitious domain methods, an irregular region of interest is embedded within a larger, simpler computational domain, and the original partial differential equation (PDE) is reformulated over the entire domain. FD methods typically employ regular Cartesian grids to mesh the computational domain, allowing for simpler discretization of PDEs and faster linear solvers to solve the discrete system of equations. However, since the regular grid does not conform to the irregular interface, incorporating original boundary conditions into the reformulated equation is not straightforward.

The volume penalization (VP) method was initially introduced by Angot et al. [7] and initially considered only Dirichlet boundary conditions. Subsequently, the VP method was extended to include Neumann and Robin boundary conditions within a finite element framework by Ramière, Angot, and Belliard [8]. In their formulation, inhomogeneous Neumann and Robin boundary conditions were incorporated by introducing a surface delta function into the reformulated equation, with regularization of the singular delta function during numerical implementation. More recently, Kadoch et al. [9] presented a volume penalization method for enforcing homogeneous Neumann boundary conditions. They implemented their method using a pseudo-spectral code to simulate moving domain problems involving chemical mixers. Since homogeneous Neumann boundary conditions were considered in this work, the need for a surface delta function kernel was circumvented. Sakurai and colleagues [1] introduced the flux-based VP method, which extends the approach of Kadoch et al. [9] to handle inhomogeneous Neumann boundary conditions. The flux-based VP method utilizes a flux-forcing function to impose the inhomogeneous Neumann conditions at the interface. Sakurai et al. employed second-order central finite differences and interpolation to implement the flux-based VP method and assessed its spatial convergence rate by solving one- and two-dimensional Poisson problems. The authors in [1] considered simple interfaces in two-spatial dimensions (circles and rectangles) in their study, which allowed them to construct flux-forcing functions analytically. Moreover, the imposed flux values were considered spatially constant on the interface. The analytical construction of flux-forcing functions limits the feasibility of the flux-based VP method for practical applications. The objectives of this thesis include generalizing the flux-based VP method to handle arbitrarily complex interfaces in two and three spatial dimensions. This is achieved through numerical construction of flux-forcing functions, as described in Chapter 2 of this thesis. Furthermore, the thesis extends the flux-based VP method to incorporate spatially varying Robin boundary conditions.

While VP methods have been extensively studied for their accuracy in penalized solutions and used for simulating FSI problems [10, 11, 12, 13, 14, 15, 16], to our knowledge, there has been no study proposing efficient solvers for the volume penalized Navier-Stokes equations, particularly under stiff system conditions. Projection methods are typically employed to solve the volume penalized incompressible Navier-Stokes (INS) equations [15, 16]. However, the projection algorithms used by [15, 16] do not lead to robust convergence of the solver, particularly when the permeability is very low. One objective of this thesis is to propose a correct projection algorithm by including the Brinkman penalty term into the pressure Poisson equation, as described in Chapter 3 of this thesis. In this work, we solve the volume penalized Navier-Stokes equations using a monolithic velocity-pressure solver with the projection method employed as a preconditioner. Results indicate that the proposed preconditioner performs well for low values of the penalization coefficient. Contrary to prior experience, the solver's convergence rate improves as the permeability parameter decreases.

VP method is also utilized to model phase change problems under a different name, known as the Carman-Kozeny drag model [17, 18]. In melting/solidification problems, a volumetric penalty/drag force is applied to retard solid phase motion [18, 19]. Melting and solidification are primary physical phenomena in the phase change materials (PCMs). The numerical modeling and simulation of PCMs is a very active area of research due to their critical role in energy systems, including concentrated solar power plants and latent thermal energy storage units [20, 21, 22, 23, 24], as well as in geophysical processes such as sea ice formation and glacier melting [25, 26], and manufacturing technologies like casting, welding, and metal

3D printing [27, 28, 29, 30].

The numerical modeling of PCMs is difficult because the energy equation is nonlinear, and most problems involve liquid flows, with some also including gas flows and solid motion. CFD models for simulating the phase change of materials began to be developed in the late 80s and early 90s. These models can be categorized into two main groups based on how they handle the moving phase boundary: deforming and fixed grid schemes. Fixed grid schemes offer greater flexibility for incorporating additional physics into heat transfer problems (e.g., fluid flow) and are easier to implement than deforming grid schemes. They also naturally handle complex topological changes of the interface (merging, pinching, break-up, self-folding), which the deforming grid methods cannot. Despite significant improvements in numerical methodologies, volume change in PCM with melting or solidification was often ignored and the variable density in the liquid was described using the Boussinesq approximation [31, 32, 33, 34, 35, 36, 37]. In this thesis, as described in Chapter 4, a fixed-grid low Mach enthalpy method is developed to capture density change-induced flow during PCM melting and solidification. In this formulation a gas phase is also incorporated and coupled to the solid–liquid PCM region. Our ultimate goal is to develop a simulation method that can handle simultaneous occurrences of evaporation, condensation, melting, and solidification.

Finally in Chapter 5, ongoing and future research topics are described, including extending the framework to accommodate condensation and evaporation, adding new features such as recoil pressure and radiation models to simulate full-scale additive manufacturing (AM) processes, and addressing turbulent fluid-structure interaction problems.

Chapter 2

Handling Neumann and Robin boundary conditions in a fictitious domain volume penalization framework

This chapter presents a flux-based volume penalization (VP) approach for imposing inhomogeneous Neumann and Robin boundary conditions on embedded interfaces. The flux-based VP method modifies the diffusion coefficient of the original elliptic (Poisson) equation and uses a flux-forcing function as a source term in the equation to impose the Neumann boundary conditions. As such, the flux-based VP method can be easily incorporated into existing fictitious domain codes. Previous studies relied on an analytical construction of flux-forcing functions, which limits the practicality of the approach. In this chapter, we present a numerical technique for constructing flux-forcing functions for arbitrarily complex boundaries. The imposed flux values are also allowed to vary spatially in our approach. Furthermore, the flux-based VP method is extended to include (spatially varying) Robin boundary conditions, which makes the flux-based VP method even more general. The numerical construction of the flux-forcing functions relies only on a signed distance function that describes the distance of a grid point from the interface and can be constructed for any irregular boundary. We consider several two- and three-dimensional test examples to access the spatial accuracy of the numerical solutions. The method is also used to simulate flux- driven thermal convection in a concentric annular domain.

2.1 Introduction

Partial differential equations (PDEs) in complex domains describe many natural and engineering processes. Examples include heat and mass transfer across melting/solidifying fronts, aquatic locomotion, cellular phenomena like cellular blebbing and cell crawling, flow in internal combustion engines or left ventricular assist devices, energy harvesting using wind turbines and wave energy converters, etc. In order to obtain meaningful solutions to PDEs, appropriate boundary conditions are required on the domain boundaries. Traditionally, body-fitted grid approaches, in which a complex domain is triangulated using sophisticated grid generation software, have been employed to solve PDEs numerically. Although body-fitted grid approaches allow imposing various types of boundary conditions accurately, they pose a serious challenge when the solution domain changes its topology over time. Issues like constant remeshing of the computational domain, the high aspect ratio of the elements, etc., limit the feasibility of body-fitted grid methods for modeling challenging moving domain problems.

To overcome the limitations of the body-fitted grid methods, fictitious domain (FD) methods have been proposed. In fictitious domain methods, an irregular region of interest is embedded into a larger, simpler computational domain and the original PDE is reformulated on the entire domain. FD methods typically employ regular Cartesian grids to mesh the computational domain. This allows simpler discretization of PDEs and fast linear solvers to solve the discrete system of equations. Since the regular grid no longer adheres to the irregular interface, incorporating original boundary conditions in the reformulated equation is not straightforward. Nevertheless, several techniques to incorporate Dirichlet boundary conditions have been proposed for various variants of the FD method. Dirichlet boundary conditions are particularly relevant for modeling fluid-structure interaction (FSI) problems, where velocity matching condition on the fluid-structure interface is required. Fictitious domain methods such as the immersed boundary (IB) method [38] and the volume penalization (VP) method [7] have been successfully used to model several FSI problems, including wave energy converters [39, 40, 13], water entry/exit problems [41], fish swimming [11, 42], esophageal transport [43], cardiovascular flows [44], etc. The IB method was introduced by Peskin to model flow in a human heart [45] and is a two-grid approach to FSI modeling: Lagrangian mesh for describing the moving structure and an Eulerian grid for describing the fluid flow. In contrast, the VP method introduced by Angot et al. [7] is a single grid approach in which all quantities related to fluid and structure are described on the Eulerian grid. The moving structure in the VP method is typically tracked using an indicator function. Since all quantities are described on a single grid, parallel implementation of VP methods on distributed memory systems is relatively easier compared to the two-grid IB methods.

The original VP method introduced by Angot et al. [7] considered only Dirichlet boundary conditions. Later the VP method was generalized to Neumann and Robin boundary conditions by Ramière, Angot, and Belliard [8]. The authors in [8] implemented their VP technique within a finite element framework. In their formulation, inhomogeneous Neumann and Robin boundary conditions were incorporated by introducing a surface delta function in the reformulated equation; the singular delta function was regularized in the numerical implementation. Recently, in Kadoch et al. [9], a volume penalization method for imposing homogeneous Neumann boundary conditions was presented. The authors in [9] implemented their method within a pseudo-spectral code and used it to simulate moving domain problems involving chemical mixers. Since homogenous (Neumann) boundary conditions were considered in Kadoch et al. [9], the need for a surface delta function kernel was circumvented. More recently, Sakurai and co-workers [1] introduced the so-called flux-based VP method, which extends Kadoch et al.'s approach to imposing inhomogeneous Neumann boundary conditions. The flux-based VP approach uses a flux-forcing function to impose the inhomogeneous Neumann boundary conditions on the interface. Sakurai et al. used second-order central finite differences and interpolation to implement the flux-based VP method and solved several one- and two-dimensional Poisson problems to assess the spatial convergence rate of the numerical solutions. Sakurai et al. considered simple interfaces in two-spatial dimensions (circles and rectangles) in their study, which allowed them to construct flux-forcing functions analytically. Moreover, the imposed flux values were considered spatially constant on the interface. The analytical construction of flux-forcing functions limits the feasibility of the flux-based VP method for practical applications. Recently, Thirumalaisamy et al. [46] critiqued Sakurai et al. for some inconsistencies in their results and conclusions, following which the authors of [1] published a corrigendum [47] to their original work. Similar to Sakurai et al., Thirumalaisamy et al. also relied on the analytical construction of flux-forcing functions for the flux-based VP method.

One of the objectives of this work is to generalize the flux-based VP method to handle arbitrarily complex interfaces in two and three spatial dimensions. This is achieved through numerical construction of flux-forcing functions, as described in Sec. 2.5.1 of this thesis. Moreover, the imposed flux values are allowed to vary spatially on the interface. The proposed numerical approach for constructing flux-forcing functions requires only a signed distance function that describes the distance of a grid point from the interface. The signed distance function can be constructed analytically for simple geometries, or through computational geometry techniques for complex interfaces [48]. Another objective of this work is to extend the flux-based VP method to include (spatially varying) Robin boundary conditions. This allows imposing both types of boundary conditions (Neumann and Robin) through similar (numerical) flux-forcing functions.

Similar to Sakurai et al., we also discretize the volume penalized equations using second-order finite differences. Using the method of manufactured solution, we assess the accuracy of the proposed approach by solving two- and three-dimensional Poisson problems with constant and spatially varying Neumann/Robin boundary conditions. We compare the performance of our approach using both continuous and discontinuous indicator functions in the test problems considered in Sec. 2.7. It is observed that the continuous indicator function performs better (in terms of order of accuracy and uniformity of convergence rate) for imposing the spatially constant Neumann/Robin boundary condition, whereas the discontinuous one performs better for the spatially varying Neumann/Robin problem.

We also provide a formal derivation of the flux-based VP Poisson equation, which was not provided in Sakurai et al. [1], where the equation first appeared for the Neumann problem. The derivation reveals that the flux-based volume penalization method also uses a surface delta function to impose inhomogeneous Neumann/Robin boundary conditions. Interestingly, explicit construction of the delta function is not required in the flux-based approach, which is in contrast to the volume penalization approach of Ramière et al. [8]. We remark that on a formulation level the volume penalization approaches of Ramière et al., Kadoch et al., and Sakurai et al. (and the present work) are equivalent; minor differences in these works arise from the definition of the surface delta function. This insight is gained from Li et al. [49] who derived phase field-based diffuse domain equations satisfying Dirichlet, Neumann, and Robin boundary conditions. Li et al. used the method of matched asymptotic expansions to provide different diffuse domain approximations for the Neumann/Robin problem [1]; these approximations differ in the way how surface delta function is defined.

Characteristic-based approaches to impose Neumann and Robin boundary conditions for the volume penalized PDEs have also been proposed in the literature; see, for example, Brown-Dymkoski et al. [50] and Hardy et al. [51] who used characteristic-based VP approach to model the energy transport equation satisfying Neumann and Robin boundary conditions in the context of compressible flows and low Mach formulation of compressible flows, respectively. The main limitation of the characteristic-based VP method is that it relies on having a time-derivative term in the PDE and as such cannot be applied to steady-state (i.e., having no temporal derivative term) elliptic equations. In addition to the volume penalization methods [8, 9, 1, 50, 51, 52, 53, 54], other fictitious domain techniques have also been proposed to impose flux boundary conditions on embedded interfaces. Notable ones include the flux-correction technique (FCT) of Ren et al. [55], Wang et al. [56], and Guo et al. [57] and the direct forcing method of Lou et al. [58]. FCT is a predictor-corrector scheme and is implemented using the Lagrangian-Eulerian machinery of the IB method. In the prediction step of FCT, an intermediate scalar field is computed on the

---

[1] Different diffuse domain approximations for the Dirichlet problem are also provided in Li et al. [49].

Eulerian grid, which in general does not satisfy the flux boundary condition on the interface defined by the Lagrangian markers. Next, in the correction step, a Lagrangian forcing term is computed either implicitly [56, 57] or explicitly [55] that corrects the intermediate scalar field to satisfy the Neumann boundary condition. In an essence, FCT is a time-splitting approach (similar to the characteristic-based VP approach), which requires having a time-derivative term in the scalar transport equation. Therefore, unlike the flux-based VP method, FCT cannot be used for time-independent elliptic equations. In the direct forcing method, the scalar field near the interface is reconstructed locally using second- or third-degree polynomials in order to satisfy the flux boundary condition. This is achieved by identifying "forcing" points on the fictitious (solid) side of the interface, on which the reconstructed scalar field value is directly imposed. Direct forcing methods are also typically implemented as a predictor-corrector scheme, which avoids modifying the system of linear equations.

In the following sections, we first describe the continuous form of the volume penalized equations and thereafter describe the numerical construction of the flux-forcing functions. Finally, various test cases are considered in two- and three-spatial dimensions to access the accuracy of the numerical solutions.

2.2 Mathematical formulation

2.2.1 The Neumann problem

Consider an irregular fluid domain Ω_{f} embedded into a larger, regular computational domain Ω, as shown in Fig. 2.1. Define $\Omega \setminus \Omega_{\text{f}} = \Omega_{\text{s}}$ as the fictitious solid domain and $\mathbf{n}$ as a unit outward normal of the fluid-solid interface $\partial\Omega_{\text{s}}$ (or $\partial\Omega_{\text{f}}$). With q as the scalar quantity of interest, κ as the diffusion coefficient, and f as a source term, Sakurai et al. [1] extended the Poisson equation defined in the fluid region Ω_{f}

$$-\nabla \cdot \kappa \, \nabla q = f, \tag{2.1}$$

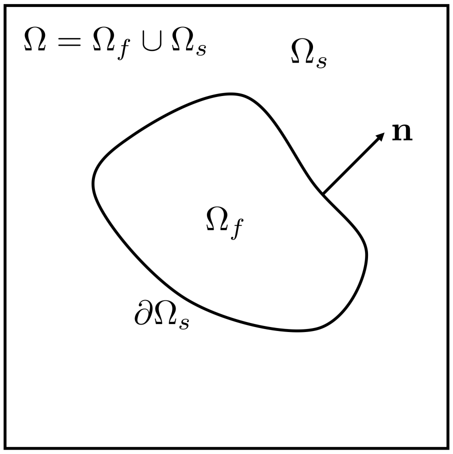


Figure 2.1. Schematic of a regular computational domain Ω with an embedded irregular fluid region Ω_{f}. The solid domain is defined as $\Omega_{\text{s}} = \Omega \setminus \Omega_{\text{f}}$. The fluid-solid interface $\partial\Omega_{\text{s}}$ (or $\partial\Omega_{\text{f}}$) has the unit normal vector $\mathbf{n}$ pointing out from the fluid and into the solid.

satisfying inhomogeneous Neumann/flux boundary conditions on $\partial\Omega_{\text{s}}$

$$-\kappa\, \mathbf{n} \cdot \nabla q = g, \tag{2.2}$$

to the entire computational domain Ω using the flux-based VP approach. The extended domain Poisson equation satisfying the inhomogeneous flux boundary conditions on the interface reads as

$$-\nabla \cdot\, [\{\kappa\,(1-\chi) + \eta\chi\}\,\nabla q] = (1-\chi)\, f + \nabla \cdot\, (\chi\boldsymbol{\beta}) - \chi\nabla \cdot \boldsymbol{\beta}. \tag{2.3}$$

Here, η is the penalization parameter, $\chi(\mathbf{x})$ is an indicator function whose value is 1 in the solid region and 0 in the fluid region, and $f_{\text{b}} = \nabla \cdot\, (\chi\boldsymbol{\beta}) - \chi\nabla \cdot \boldsymbol{\beta}$ is an additional forcing term required to impose the flux boundary conditions on $\partial\Omega_{\text{s}}$. The vector-valued flux-forcing function $\boldsymbol{\beta}(\mathbf{x})$ is selected such that $\boldsymbol{\beta} \cdot \mathbf{n} = -g$ on the interface. In the limit of $\eta \to 0$, the solution to the volume penalized (VP) Poisson equation converges to the solution of non-penalized Poisson equation (Eqs. (2.1) and (2.2)). A formal derivation of Eq. (2.3) is provided in the next section. As noted in Thirumalaisamy et al. [46], the flux-based VP approach allows κ and g to vary spatially as well.

2.3 Derivation of the flux-based volume penalized Poisson equation: the Neumann problem

In this section, we derive Eq. (2.3) by following the diffuse domain equation derivation provided in Li et al. [49]. Similar derivation appeared in Ramière et al. [8]. To begin, multiply Eq. (2.1) by a test function ψ and integrate it over the fluid domain Ω_f to obtain

$$\int_{\partial\Omega_\text{f}} \psi g \;\, \text{dS} + \int_{\Omega_\text{f}} \kappa \; \nabla q \cdot \nabla \psi \;\, \text{dV} = \int_{\Omega_\text{f}} \psi f \;\, \text{dV}. \tag{2.4}$$

In the above equation we used the vector identity

$$\nabla \cdot (a\mathbf{b}) = (\nabla a) \cdot \mathbf{b} + a \; (\nabla \cdot \mathbf{b}), \tag{2.5}$$

with the scalar field $a = \psi$ and vector field $\mathbf{b} = \kappa \; \nabla q$, along with the Neumann boundary condition on the fluid-solid interface $\partial\Omega_\text{f}$ as written in Eq. (2.2). Note that Eq. (2.4) is the weak form of the Poisson Eq. (2.1) defined in the fluid domain Ω_f. Next, extend the integration domain from region Ω_f to Ω in the integral Eq. (2.4) by introducing the indicator function χ ($\chi = 0$ in Ω_f and $\chi = 1$ in Ω_s) and a surface delta function $\delta_{\partial\Omega_\text{f}}$ to obtain

$$\int_{\Omega} \delta_{\partial\Omega_\text{f}} \, \psi g \;\, \text{dV} + \int_{\Omega} (1-\chi) \kappa \; \nabla q \cdot \nabla \psi \;\, \text{dV} = \int_{\Omega} (1-\chi) \psi f \;\, \text{dV}. \tag{2.6}$$

Again invoking the vector identity defined in Eq. (2.5), but with $\mathbf{b} = (1-\chi)\kappa \; \nabla q$ this time, the second integrand in the left-hand side of the above equation can be written as

$$(1-\chi)\kappa \; \nabla q \cdot \nabla \psi = \nabla \cdot [\psi (1-\chi) \kappa \; \nabla q] - \psi \; \nabla \cdot [(1-\chi)\kappa \; \nabla q].$$

This allows us to simplify the second integral in the left-hand side of Eq. (2.6) as

$$\int_{\Omega}(1-\chi)\kappa\ \nabla q\cdot\nabla\psi\ \mathrm{dV} = \underbrace{\int_{\partial\Omega}\psi(1-\chi)\kappa\ (\nabla q\cdot\mathbf{n}_{\partial\Omega})\ \mathrm{dS}}_{=\,0\text{ as }\chi\,=\,1\text{ on }\partial\Omega} - \int_{\Omega}\psi\ \nabla\cdot[(1-\chi)\kappa\ \nabla q]\ \mathrm{dV}.$$

Therefore, the weak form of the Poison equation in the extended domain can be written as

$$\int_{\Omega}\psi\left(-\nabla\cdot[(1-\chi)\kappa\ \nabla q]+\delta_{\partial\Omega_{\mathrm{f}}}\, g-(1-\chi)f\right)\ \mathrm{dV}=0. \tag{2.7}$$

Since ψ is an arbitrary test function, the collective term multiplying ψ in Eq. (2.7) should evaluate to zero at each point in the domain. This gives the strong form of the extended domain Poisson equation as

$$-\nabla\cdot[(1-\chi)\kappa\ \nabla q]+\delta_{\partial\Omega_{\mathrm{f}}}\, g=(1-\chi)f. \tag{2.8}$$

Next, we show that the flux-based volume penalized Poisson Eq. (2.3) can be obtained from Eq. (2.8) using a specific definition of the surface delta function $\delta_{\partial\Omega_{\mathrm{f}}}$. First, simplify the forcing term f_{b} in the right hand-side of Eq. (2.3) to

$$\nabla\cdot\ (\chi\boldsymbol{\beta})-\chi\nabla\cdot\boldsymbol{\beta}=\boldsymbol{\beta}\cdot\nabla\chi. \tag{2.9}$$

Next, noticing that $\nabla\chi=\delta_{\partial\Omega_{\mathrm{f}}}\,\mathbf{n}$, the forcing term of VP poisson equation becomes

$$\boldsymbol{\beta}\cdot\nabla\chi=\delta_{\partial\Omega_{\mathrm{f}}}(\boldsymbol{\beta}\cdot\mathbf{n})=-\delta_{\partial\Omega_{\mathrm{f}}}\, g. \tag{2.10}$$

Substituting $\delta_{\partial\Omega_{\mathrm{f}}}\ g$ term from Eq. (2.10) into the extended domain Poisson equation (2.8) eliminates the explicit representation of the delta function and the extended domain equation reads as

$$-\nabla\cdot[(1-\chi)\kappa\ \nabla q]=(1-\chi)f+\nabla\cdot\ (\chi\boldsymbol{\beta})-\chi\nabla\cdot\boldsymbol{\beta}. \tag{2.11}$$

The flux-based VP Poisson equation is obtained from Eq. (2.11) by introducing a small amount of diffusion in the solid domain which is controlled by the penalization parameter η. For an easy reference, the VP equation is re-written below

$$-\nabla \cdot [\{\kappa(1-\chi) + \eta\chi\}\nabla q] = (1-\chi)f + \nabla \cdot\ (\chi\boldsymbol{\beta}) - \chi\nabla \cdot \boldsymbol{\beta}.$$

2.4 Derivation of the flux-based volume penalized Poisson equation: the Robin problem

The flux-based volume penalization method can be easily extended to include Robin boundary conditions of the type

$$\zeta\, q + \kappa\, \mathbf{n} \cdot \nabla q = -g \tag{2.12}$$

on the irregular boundary $\partial\Omega_{\mathrm{f}}$ (or $\partial\Omega_{\mathrm{s}}$). First, it can be easily verified that the weak form of the Poisson equation defined in the fluid domain and satisfying Robin boundary conditions written in Eq. (2.12) is

$$\int_{\partial\Omega_{\mathrm{f}}} \psi(\zeta q + g)\ \mathrm{dS} + \int_{\Omega_{\mathrm{f}}} \kappa\ \nabla q \cdot \nabla\psi\ \mathrm{dV} = \int_{\Omega_{\mathrm{f}}} \psi f\ \mathrm{dV}. \tag{2.13}$$

Next, following the procedure to reformulate the PDE on the entire domain as described above, the strong form of the Poisson equation reads as

$$-\nabla \cdot [(1-\chi)\kappa\ \nabla q] + \delta_{\partial\Omega_{\mathrm{f}}}\ (\zeta q + g) = (1-\chi)f. \tag{2.14}$$

Defining a flux function $\boldsymbol{\beta}$ that satisfies the property of $\boldsymbol{\beta} \cdot \mathbf{n} = -g$ on $\partial\Omega_{\text{f}}$, the above equation can be written as

$$-\nabla \cdot [(1-\chi)\kappa \, \nabla q] + \nabla\chi \cdot (\zeta q \, \mathbf{n} - \boldsymbol{\beta}) = (1-\chi) f. \tag{2.15}$$

The flux-based VP Poisson equation satisfying the Robin boundary conditions is obtained from Eq. (2.15) by adding a small amount of diffusion in the solid domain

$$(\zeta \, \nabla\chi \cdot \mathbf{n}) q - \nabla \cdot [\{\kappa \, (1-\chi) + \eta\chi\} \;\; \nabla q] = (1-\chi) f + \nabla \cdot (\chi\boldsymbol{\beta}) - \chi \nabla \cdot \boldsymbol{\beta}. \tag{2.16}$$

In order to avoid computing the gradient of a possible discontinuous indicator function χ, the above equation is re-written as

$$\zeta[\nabla \cdot (\chi\mathbf{n}) - \chi\nabla \cdot \mathbf{n}] q - \nabla \cdot [\{\kappa \, (1-\chi) + \eta\chi\} \;\; \nabla q] = (1-\chi) f + \nabla \cdot (\chi\boldsymbol{\beta}) - \chi \nabla \cdot \boldsymbol{\beta}. \tag{2.17}$$

The normal vector appearing in the first term of Eq. (2.17) can be computed numerically using the signed distance function as $\mathbf{n} = -\nabla\phi$.

2.4.1 The Robin problem

Next, we consider the inhomogeneous Robin boundary conditions of the type

$$\zeta \, q + \kappa \, \mathbf{n} \cdot \nabla q = -g \tag{2.18}$$

on the fluid-solid interface $\partial\Omega_{\text{s}}$. The flux-based VP Poisson equation for the Robin problem, which reads as

$$\zeta[\nabla \cdot (\chi\mathbf{n}) - \chi\nabla \cdot \mathbf{n}] q - \nabla \cdot [\{\kappa \, (1-\chi) + \eta\chi\} \;\; \nabla q] = (1-\chi) f + \nabla \cdot (\chi\boldsymbol{\beta}) - \chi \nabla \cdot \boldsymbol{\beta}. \tag{2.19}$$

In the equation above, the flux-forcing function satisfies the requirement of $\boldsymbol{\beta} \cdot \mathbf{n} = -g$. The unit normal vector $\mathbf{n}$ appearing in the first term of Eq. (2.19) can be computed numerically using a signed distance function as explained later in Sec. 2.4.3. In our formulation, ζ, κ, and g are allowed to vary spatially.

2.4.2 Multiple interfaces and coupled volume penalized equations

The VP Poisson equations (Eqs. (2.3) and (2.19)) can also be generalized to handle multiple interfaces within the computational domain Ω. For some of these interfaces, Dirichlet boundary conditions may also be prescribed. Following Thirumalaisamy et al. [46], the generalized form of the VP Poisson equation satisfying Neumann and Dirichlet boundary conditions reads as

$$\begin{aligned} -\nabla \cdot \left[\left\{\kappa\left(1-\sum_{j=1}^{N}\chi_j^{\mathrm{n}}\right)+\sum_{j=1}^{N}\eta\chi_j^{\mathrm{n}}\right\}\nabla q\right] &= \sum_{j=1}^{N}\left\{\nabla \cdot \left(\chi_j^{\mathrm{n}}\boldsymbol{\beta}_j\right)-\chi_j^{\mathrm{n}}\nabla\cdot\boldsymbol{\beta}_j\right\} \\ &\quad -\sum_{i=1}^{D}\frac{\chi_i^{\mathrm{d}}\left(q-q_i^{\mathrm{d}}\right)}{\eta}+\left(1-\sum_{j=1}^{N}\chi_j^{\mathrm{n}}\right)f. \end{aligned} \tag{2.20}$$

For the above equation to hold true, the computational domain Ω is assumed to consist of disjoint volumetric regions Ω_i^{d} (for $i = 1, 2, \ldots, D$) and Ω_j^{n} (for $j = 1, 2, \ldots, N$), with imposed Dirichlet ($q = q_i^{\mathrm{d}}$) and Neumann ($-\kappa\ \nabla q \cdot \mathbf{n}_j = g_j^{\mathrm{n}}$) boundary conditions, respectively. Furthermore, the union of Ω_i^{d} and Ω_j^{n} regions defines the total solid domain, i.e., $\Omega_{\mathrm{s}} = \Omega_1^{\mathrm{d}} \cup \Omega_2^{\mathrm{d}} \cup \cdots \Omega_D^{\mathrm{d}} \cup \Omega_1^{\mathrm{n}} \cup \Omega_2^{\mathrm{n}} \cup \cdots \Omega_N^{\mathrm{n}}$. In Eq. (2.20) the indicator function $\chi^{\mathrm{n}}(\mathbf{x})(\text{respectively}, \chi^{\mathrm{d}}(\mathbf{x}))$ is 1 if $\mathbf{x} \in \Omega^{\mathrm{n}}(\text{respectively}, \Omega^{\mathrm{d}})$ and 0 if $\mathbf{x} \in \Omega \setminus \Omega^{\mathrm{n}}(\text{respectively}, \Omega^{\mathrm{d}})$. Note that Robin boundary conditions can be easily included in Eq. (2.20), as their form is very similar to the Neumann problem. We omit Robin boundary conditions in the generalized equation written above for brevity.

The volume penalization approach can also be extended to other governing equations that describe conservation of momentum, energy, species, etc. For example, the VP incompressible Navier-Stokes equations coupled to the flux-based VP advection-diffusion equation satisfying

Neumann boundary condition reads as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla p + \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right] + \frac{\chi}{\eta}(\mathbf{u}_\text{b} - \mathbf{u}) + \mathbf{f}(\mathbf{x}, q, t), \quad (2.21)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.22)$$

$$\frac{\partial q}{\partial t} + (1-\chi)(\mathbf{u} \cdot \nabla q) = \nabla \cdot \left[\{\kappa (1-\chi) + \eta \chi\} \nabla q\right] + (1-\chi) f + \nabla \cdot (\chi \boldsymbol{\beta}) - \chi \nabla \cdot \boldsymbol{\beta}. \quad (2.23)$$

In the equations above, $\mathbf{u}(\mathbf{x}, t)$ is the fluid velocity, $\mathbf{u}_\text{b}(\mathbf{x}, t)$ is the structure velocity, $p(\mathbf{x}, t)$ is the hydrodynamic pressure, $\mathbf{f}$ denotes the momentum body force, $\rho(\mathbf{x})$ is the mass density, and $\mu(\mathbf{x})$ is the dynamic viscosity. The equation set (5.6)-(2.23) is written considering only a single interface in the domain; generalization to handle multiple interfaces is also possible following Eq. 2.20. We remark that in the context of fluid-structure interaction (FSI) problems, only velocity matching condition on the fluid-structure interface is required, i.e., $\mathbf{u} = \mathbf{u}_\text{b}$ on $\partial\Omega_\text{s}$ is essential, whereas $\mathbf{u} = \mathbf{u}_\text{b}$ in Ω_s is optional. In the volume penalization approach to FSI, both these conditions are imposed through the penalization term $\frac{\chi}{\eta}(\mathbf{u}_\text{b} - \mathbf{u})$. Therefore, in the VP momentum equation (5.6), only Dirichlet boundary conditions have been considered.

2.4.3 Interface capturing

We use a signed distance function $\phi(\mathbf{x})$ to implicitly define the fluid-solid interface $\partial\Omega_\text{s}$. The scalar field $\phi(\mathbf{x})$ is defined to satisfy the following property: $\phi(\mathbf{x}) > 0$ if $\mathbf{x} \in \Omega_\text{f}$, $\phi(\mathbf{x}) < 0$ if $\mathbf{x} \in \Omega_\text{s}$ and $\phi(\mathbf{x}) = 0$ is $\mathbf{x} \in \partial\Omega_\text{s}$. Moreover, the negative gradient of the signed distance function $\phi(\mathbf{x})$ gives the unit outward normal vector of the interface, i.e., $\mathbf{n} = -\nabla\phi$. The signed distance function can also be used to define the indicator function $\chi(\mathbf{x})$. In this work we use

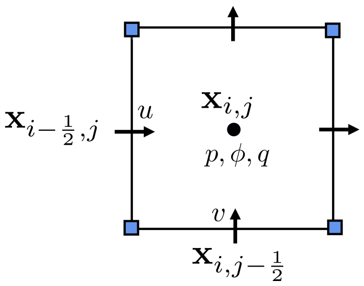


Figure 2.2. Schematic of a single Cartesian grid cell along with the placement of various variables: the velocity components are stored at the edge centers (black, →); the fluid pressure p, the transported quantity q and the signed distance function ϕ are stored at the cell centers (black, $\circ$).

$\phi(\mathbf{x})$ to define two types of indicator functions: one is smooth and continuous and written as

$$\chi(\mathbf{x}) = \begin{cases} 1, & \phi(\mathbf{x}) < -n_{\text{smear}}\, h, \\ 1 - \frac{1}{2}\left(1 + \frac{1}{n_{\text{smear}} h}\phi(\mathbf{x}) + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{smear}} h}\phi(\mathbf{x})\right)\right), & |\phi(\mathbf{x})| \leq n_{\text{smear}}\, h, \\ 0, & \text{otherwise}, \end{cases} \tag{2.24}$$

and the other one is discontinuous, which reads as

$$\chi(\mathbf{x}) = \begin{cases} 1, & \phi(\mathbf{x}) < 0, \\ \frac{1}{2}, & \phi(\mathbf{x}) = 0, \\ 0, & \text{otherwise}. \end{cases} \tag{2.25}$$

In the Eq. (2.24) above, $n_{\text{smear}} \in \mathbb{R}$ is the number of grid cells over which the indicator function is smoothed on either side of the interface and h is the grid cell size.

2.5 Discrete equations

We use second-order finite difference stencils to discretize the spatial derivative terms of cell-centered Poisson and face-centered momentum equations (Eqs. (2.3) and (5.6), respectively) on a Cartesian grid. Fig. 2.2 shows a schematic representation of a two-dimensional Cartesian grid cell, in which the velocity components are stored on edge centers (face centers in three spatial dimensions), whereas the transported variable q, the fluid pressure p, and the signed distance function ϕ are stored at the cell center. The computational domain Ω is discretized into $N_x \times N_y$ Cartesian grid cells with mesh spacing Δx and Δy in the x- and y-direction, respectively. In this work we use equal mesh spacing in the two directions, i.e., $\Delta x = \Delta y = h$. In what follows next, we primarily focus on the discretization of the VP Poisson Eq. (2.3) for the Neumann problem; details on the spatiotemporal discretization of the VP incompressible Navier-Stokes equations can be found in our prior works [59, 41, 40].

Referring to Fig. 2.2, let (i, j) denote the cell index, $(i - \frac{1}{2}, j)$ denote the lower x edge index and $(i, j - \frac{1}{2})$ denote the lower y edge index. Then the discretized form of the VP Poisson Eq. 2.3 in two spatial dimensions reads as

$$\begin{aligned}\{\psi_{i+\frac{1}{2},j} + \psi_{i-\frac{1}{2},j} + \psi_{i,j+\frac{1}{2}} + \psi_{i,j-\frac{1}{2}}\}\, q_{i,j} - \psi_{i+\frac{1}{2},j}\, q_{i+1,j} - \psi_{i-\frac{1}{2},j}\, q_{i-1,j} - \psi_{i,j+\frac{1}{2}}\, q_{i,j+1} \\ - \psi_{i,j-\frac{1}{2}}\, q_{i,j-1} = S_{i,j},\end{aligned} \tag{2.26}$$

in which

$$\psi_{i+\frac{1}{2},j} = \frac{1}{\Delta x^2}\{\kappa\,(1 - \chi + \eta\chi)\}_{i+\frac{1}{2},j} \tag{2.27a}$$

$$\psi_{i-\frac{1}{2},j} = \frac{1}{\Delta x^2}\{\kappa\,(1 - \chi + \eta\chi)\}_{i-\frac{1}{2},j} \tag{2.27b}$$

$$\psi_{i,j+\frac{1}{2}} = \frac{1}{\Delta y^2}\{\kappa\,(1 - \chi + \eta\chi)\}_{i,j+\frac{1}{2}} \tag{2.27c}$$

$$\psi_{i,j-\frac{1}{2}} = \frac{1}{\Delta y^2}\{\kappa\,(1 - \chi + \eta\chi)\}_{i,j-\frac{1}{2}}, \tag{2.27d}$$

and the right hand side term $S_{i,j}$ is given by

$$S_{i,j} = (1 - \chi_{i,j})\, f_{i,j} + \frac{(\chi\beta)_{i+\frac{1}{2},j} - (\chi\beta)_{i-\frac{1}{2},j}}{\Delta x} + \frac{(\chi\beta)_{i,j+\frac{1}{2}} - (\chi\beta)_{i,j-\frac{1}{2}}}{\Delta y}$$
$$- \chi_{i,j}\left(\frac{\beta_{i+\frac{1}{2},j} - \beta_{i-\frac{1}{2},j}}{\Delta x} + \frac{\beta_{i,j+\frac{1}{2}} - \beta_{i,j-\frac{1}{2}}}{\Delta y}\right). \tag{2.28}$$

Analogous discretization formulas can be written for the three-dimensional VP Poisson equation. In the discretized Eq. (2.26) written above, the indicator function χ and the diffusion coefficient κ are required at the edge centers; these properties are first defined at the cell centers and then interpolated onto the edge centers using a second-order accurate linear interpolation scheme. The flux-forcing function $\boldsymbol{\beta}(\mathbf{x})$ is also required at the edge centers; methods to construct $\boldsymbol{\beta}$ are discussed next.

2.5.1 Construction of flux-forcing functions

The vector-valued flux-forcing function $\boldsymbol{\beta}(\mathbf{x})$ plays a crucial role in imposing the desired inhomogeneous Neumann and Robin boundary conditions on the interface. In this section, we introduce three approaches to construct $\boldsymbol{\beta}(\mathbf{x})$, namely Approach A, B, and C. The three approaches are in increasing order of generality. While Approach A is specialized for the Neumann problem, Approaches B and C are equally applicable to the Robin problem.

Approach A: Analytical construction of spatially varying g

Consider for a moment that the solution to the non-penalized Poisson equation with inhomogeneous Neumann boundary conditions on $\partial\Omega_\text{s}$ is known. Denote the exact solution by q_exact. If the flux-forcing function is taken to be of the form $\boldsymbol{\beta}(\mathbf{x}) = \kappa\ \nabla q_\text{exact}(\mathbf{x})$, then it satisfies the requirement of $\boldsymbol{\beta} \cdot \mathbf{n} = -g$ on $\partial\Omega_\text{s}$. In practice the solution to the Poisson Eq. (2.1) with boundary condition (2.2) is sought and not known *a priori*. However, if an analytical approximation $\widetilde{q}$ to the exact solution q_{exact} exists, such that $\mathbf{n} \cdot \nabla \widetilde{q} = \mathbf{n} \cdot \nabla q_{\text{exact}}$ on $\partial\Omega_{\text{s}}$, then

$$\boldsymbol{\beta}(\mathbf{x}) = \kappa \, \nabla \widetilde{q}(\mathbf{x}), \tag{2.29}$$

can be prescribed as a flux-forcing function. Away from the interface, the approximation $\widetilde{q}$ can be close to or very different from q_{exact}, depending upon whether a continuous or a discontinuous indicator function χ is used. We denote the analytical construction of $\boldsymbol{\beta}$ as Approach A. In component form, Approach A is written as

$$\beta_{i-\frac{1}{2},j} = \left(\kappa \frac{\partial \widetilde{q}}{\partial x}\right)_{\mathbf{x}_{i-\frac{1}{2},j}}, \tag{2.30a}$$

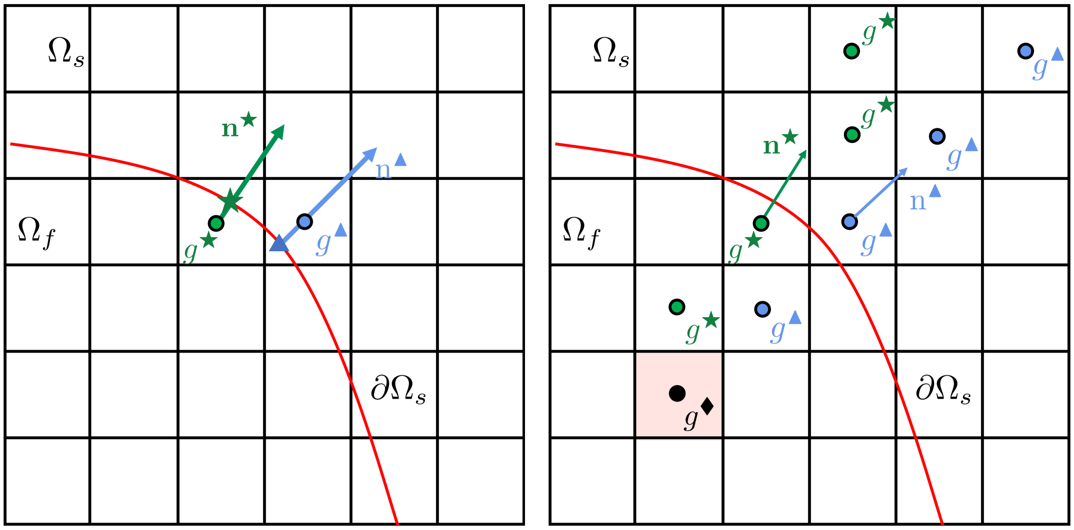
$$\beta_{i,j-\frac{1}{2}} = \left(\kappa \frac{\partial \widetilde{q}}{\partial y}\right)_{\mathbf{x}_{i,j-\frac{1}{2}}}. \tag{2.30b}$$

Approach A was employed in Sakurai et al. [1] and Thirumalaisamy et al. [46] to demonstrate the feasibility of flux-based volume penalization method to solve PDEs with flux boundary conditions in complex domains, but is quite restrictive in practice as discussed next.

Approach B: Numerical construction of spatially constant g

Although Approach A allows for imposing spatially varying g values on the interface, approximating an analytical solution to the exact solution near the interface is a non-trivial task, especially if the interface is geometrically complex. However, if g is spatially constant, then constructing $\boldsymbol{\beta}$ is easy. This is achieved by taking $\boldsymbol{\beta}(\mathbf{x}) = -g \; \mathbf{n}(\mathbf{x})$, as it satisfies the requirement of $\boldsymbol{\beta} \cdot \mathbf{n} = -g$ on $\partial\Omega_{\text{s}}$. Now, recalling from Sec. 2.4.3 that the negative gradient of the signed distance function $\phi(\mathbf{x})$ is the continuous normal vector field $\mathbf{n}(\mathbf{x})$, the flux-forcing function can be constructed numerically for an irregular boundary as

$$\boldsymbol{\beta}(\mathbf{x}) = -g \; \mathbf{n}(\mathbf{x}) = g \; \nabla \phi(\mathbf{x}). \tag{2.31}$$



(A) Cartesian grid and an embedded interface

(B) Propagation of interface cell g values

Figure 2.3. Approach C for constructing the flux-forcing function $\boldsymbol{\beta}$. (A) Schematic of a Cartesian grid (black lines) with an embedded fluid-solid interface $\partial\Omega_{\mathrm{s}}$ (red line). Two interface cells are highlighted in the figure: one whose cell center $\mathbf{x}^{\bigstar}$ lies in the fluid region and the other whose cell center $\mathbf{x}^{\blacktriangle}$ lies in the solid region. For these interface cells the corresponding function values $g^{\bigstar}$ and $g^{\blacktriangle}$ and the outward unit normal vectors $\mathbf{n}^{\bigstar}$ and $\mathbf{n}^{\blacktriangle}$ are shown. (B) Propagation of the g value into the domain following an interface cell normal and a grid cell where multiple normals intersect shown in light pink color. Out of the $g^{\bigstar}$ and $g^{\blacktriangle}$ values arriving at the shaded cell, the one with the larger modulus is chosen.

We denote the numerical construction of spatially constant g value on the interface as Approach B, which in component form is written as

$$\beta_{i-\frac{1}{2},j} = g\left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}\right), \tag{2.32a}$$

$$\beta_{i,j-\frac{1}{2}} = g\left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y}\right). \tag{2.32b}$$

Approach C: Numerical construction of spatially varying g

As a generalization of Approach B, the flux-forcing function can be taken as

$$\boldsymbol{\beta}(\mathbf{x}) = -g(\mathbf{x})\,\mathbf{n}(\mathbf{x}) = g(\mathbf{x})\nabla\phi(\mathbf{x}). \tag{2.33}$$

However, Eq. (2.33) poses a challenge of extending the codimension-1 boundary condition function g defined over the interface to a codimension-0 function $g(\mathbf{x})$ defined in the neighborhood of the interface. Although there are several ways to achieve this function extension (in absence of a constraint), in this work we follow a simple strategy of propagating the interfacial g values to the neighboring grid cells along the interface normal. More specifically, consider a fluid-solid interface $\partial\Omega_\text{s}$ embedded into a Cartesian grid as shown in Fig. 2.3(A). The signed distance function $\phi(\mathbf{x})$ can be used to identify the grid cells through which the interface passes. Denote these grid cells as *interface cells*. Fig. 2.3(A) highlights two such interface cells: one whose cell center $\mathbf{x}^\bigstar$ lies in the fluid region and the other whose cell center $x^\blacktriangle$ lies in the solid region. The normal vector of the interface cells is also known from the signed distance function: $\mathbf{n}^\bigstar = (-\nabla\phi)^\bigstar$ and $\mathbf{n}^\blacktriangle = (-\nabla\phi)^\blacktriangle$. Next, the g value at the cell center of an interface cell is set equal to the closest interfacial g value:

$$g^\bigstar \leftarrow g(\mathbf{x}^\bigstar_{\partial\Omega_\text{s}}), \tag{2.34a}$$

$$g^\blacktriangle \leftarrow g(\mathbf{x}^\blacktriangle_{\partial\Omega_\text{s}}), \tag{2.34b}$$

in which $\mathbf{x}^\bigstar_{\partial\Omega_\text{s}} = \mathbf{x}^\bigstar + \phi^\bigstar\mathbf{n}^\bigstar$ and $\mathbf{x}^\blacktriangle_{\partial\Omega_\text{s}} = \mathbf{x}^\blacktriangle + \phi^\blacktriangle\mathbf{n}^\blacktriangle$ are the closest points on the interface to the cell centers $\mathbf{x}^\bigstar$ and $\mathbf{x}^\blacktriangle$, respectively. Note that the g function on the interface is prescribed and therefore, $g(\mathbf{x}^\bigstar_{\partial\Omega_\text{s}})$ and $g(\mathbf{x}^\blacktriangle_{\partial\Omega_\text{s}})$ are known *a priori*. In the next part of the algorithm, $g^\bigstar$ and $g^\blacktriangle$ values are propagated to the grid cells that are within a distance of $n_\text{prop}\ h$ to the interface cells along $\pm\ \mathbf{n}^\bigstar$ and $\pm\ \mathbf{n}^\blacktriangle$ directions, respectively. This procedure is pictorially described in Fig. 2.3(B). The number of grid cells n_prop to which g values are propagated depends upon the choice of the indicator function χ—we will explore the effect of n_prop on the solution accuracy in Sec. 2.7. Note that propagating g values along the normal directions may lead to a situation of conflict at a grid cell where two or more interface cell normals intersect. This situation is shown for the shaded cell in Fig. 2.3(B) where the two normals $\mathbf{n}^\bigstar$ and $\mathbf{n}^\blacktriangle$ intersect. For such cells, a g

value with the larger modulus is chosen:

$$g^{\blacklozenge} = \mathrm{maxmodulus}(g^{\bigstar}, g^{\blacktriangle}). \tag{2.35}$$

We also considered an average and the minimum modulus of g at the conflicted cells; these choices however reduced the order of accuracy of the solution for the continuous/smoothed indicator function. Note that for the discontinuous indicator function, the g value at the conflicted cells does not matter much for the solution accuracy. This is because such cells are generally located far away from the interface where the discontinuous indicator function is already zero. Nevertheless, we always make use of Eq. (2.35) even for the discontinuous indicator function in this work. With g values defined at the cell centers, the component form of $\boldsymbol{\beta}$ reads as

$$\beta_{i-\frac{1}{2},j} = \left(\frac{g_{i-1,j} + g_{i,j}}{2}\right)\left(\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}\right), \tag{2.36a}$$

$$\beta_{i,j-\frac{1}{2}} = \left(\frac{g_{i,j-1} + g_{i,j}}{2}\right)\left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y}\right). \tag{2.36b}$$

The propagation strategy of Approach C can also be implemented by solving a hyperbolic equation of the form

$$\frac{\partial g(\mathbf{x})}{\partial \tau} + \mathbf{n}(\mathbf{x}) \cdot \nabla g(\mathbf{x}) = 0. \tag{2.37}$$

The equation above can be integrated over a pseudo-time interval $\Delta\tau$ that is directly related to the propagation distance. However, the test examples of Sec. 2.7 show that the method of g propagation described in Approach C is quite effective in imposing the spatially varying flux boundary conditions. Moreover, it does not require solving any additional partial differential equation.

Note that there can be other ways of extending the flux-forcing function in the vicinity of the interface as discussed at the beginning of this section. One straightforward approach is to extend the $\boldsymbol{\beta}$ function defined over the interface Ω_{s} to a flux-forcing function valid near the interface using the top hat or a Gaussian bell-like function, which we refer to as Approach D. However, as demonstrated in Appendix 6.1, this particular function continuation approach does not produce satisfactory results; the numerical and actual solutions differ significantly and the numerical scheme does not converge under grid refinement in any norm. In contrast, Approach C produces the correct solution and a convergent numerical scheme. This also highlights the non-triviality in allowing spatially varying Neumann/Robin boundary conditions in the flux-based VP method.

2.6 Software

The numerical formulations and the problems described in this thesis are implemented within the IBAMR library [60], which is an open-source C++ simulation software focused on immersed boundary and volume penalization methods with adaptive mesh refinement. The code and test cases presented in this thesis are publicly available at https://github.com/IBAMR/IBAMR. IBAMR relies on SAMRAI [61, 62] for Cartesian grid management and the AMR framework. Linear and nonlinear solver support in IBAMR is provided by the PETSc library [63, 64, 65]. All of the example cases in the present work made use of distributed-memory parallelism using the Message Passing Interface (MPI) library.

2.7 Results and discussion

In this section we discretely solve the volume penalized Poisson Eqs. 2.3 and 2.19 satisfying inhomogeneous Neumann and Robin boundary conditions, respectively, to assess the accuracy of the numerical solutions. We use the flexible GMRES (FGMRES) iterative solver with a tight relative residual tolerance of 10^{-12} to solve the system of linear equations. The order of accuracy results presented here are computed only in the fluid domain and are determined based on the L^1 and L^∞ norm of the error (denoted $\mathcal{E}^1$ and $\mathcal{E}^\infty$, respectively) between the numerical and analytical [2] solutions. Since the VP method is expected to produce a non-uniform convergence rate under grid refinement because of the delta function formulation (see Appendices 2.3 and 2.4 for derivation), we curve-fit the error data and report the slope/convergence rate, denoted m and the coefficient of determination, denoted R^2, in each case. Appendix 6.9 tabulates the error data. The spatial convergence rate of the error is shown for both continuous (denoted $\mathcal{E}_{\text{c}}^{\infty}$ and $\mathcal{E}_{\text{c}}^{1}$) and discontinuous (denoted $\mathcal{E}_{\text{d}}^{\infty}$ and $\mathcal{E}_{\text{d}}^{1}$) indicator functions. We consider two- and three-dimensional examples involving constant and spatially varying flux boundary conditions on $\partial\Omega_{\text{s}}$. In the test examples, the fluid region Ω_{f} is embedded into a larger computational domain Ω with Dirichlet boundary conditions imposed on the external boundary $\partial\Omega$ of the domain. The computational domain is discretized into $N \times N$ and $N \times N \times N$ grid cells for the two- and three-dimensional examples, respectively. The penalization parameter η is taken to be 10^{-8} (Appendix 6.2 considers the effect of η on the convergence rate) and the diffusion coefficient κ is taken to be 1 for all of the tests. While imposing the Robin boundary conditions we take ζ to be 1 in the test examples. The numerical solutions are presented for Approach C and where applicable, results obtained from Approach C are compared against Approach A or B. Since Approach A constructs the flux-forcing function from the known solution to the problem, for a given indicator function χ, Approach A is expected to perform better than or at least as well as Approach B and C. This expectation is also confirmed from the tests that follow next.

2.7.1 Analysis of 1D Poisson equation with same inhomogeneous Neumann boundary condition

We first consider the 1D Poisson problem with the same flux boundary condition on the two ends of the fluid domain Ω_{f}, as done in Sec. 2.1 of Sakurai et al. [1]. The fluid domain $\Omega_{\text{f}} \in [0, \pi]$ is embedded into a larger computational domain $\Omega \in [0, 2\pi]$, as shown in Fig. 2.4. Same inhomogeneous Neumann boundary condition value is imposed on the two fluid-solid

---

[2] Analytical solution of the non-penalized equation is used for computing the error.

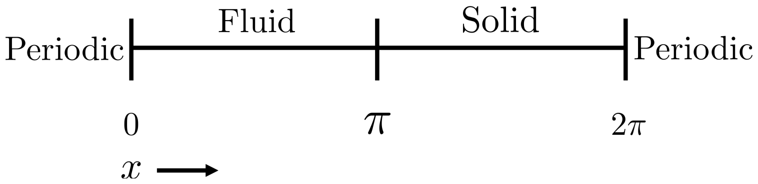


Figure 2.4. Schematic of the computational domain used in Sakurai et al. [1] to solve the 1D Poisson problem in the fluid region. The solid region in the figure represents the fictitious domain. Periodic boundary conditions are imposed on the external boundaries.

interfaces located at $x = 0$ and $x = \pi$, respectively, and is taken to be

$$\left.\frac{\mathrm{d}q}{\mathrm{d}x}\right|_{x=0} = \alpha \quad \text{and} \quad \left.\frac{\mathrm{d}q}{\mathrm{d}x}\right|_{x=\pi} = \alpha. \tag{2.38}$$

We take the flux forcing function to be $\boldsymbol{\beta} = \alpha\,\hat{\mathbf{i}}$ for this test case. Here, $\hat{\mathbf{i}}$ denotes the unit vector in the positive $x-$direction. The forcing function $f(\mathbf{x})$ is taken to be

$$f(x) = m^2 \cos(mx). \tag{2.39}$$

The analytical solution of this problem using a zero-mean condition on q in Ω_f, $\int_{\Omega_\mathrm{f}} q(x)\ \mathrm{d}x = 0$, reads as

$$q_{\text{exact}}(x) = \cos(mx) + \alpha x - \frac{\pi\alpha}{2}. \tag{2.40}$$

We solve the penalized Poisson equation using both continuous and discontinuous indicator functions as defined in Sec. 2.4.3. The parameters α and m are taken to be 1, and periodic boundary conditions are imposed on $\partial\Omega$ (see Fig. 2.4). Since the solution to the Poisson equation on a periodic computational domain is determinable only up to an additive constant, the discrete set of equations for this case results in a singular matrix. To invert the matrix using a direct solver, we replace the first linear equation with $\int_\Omega q(x)\ \mathrm{d}x = 0$ condition, as done in Kolomenskiy et al. [53]. We remark that although the obtained numerical solution depends on the linear equation that is replaced by the zero-mean condition (as also noted in [53]), the order of accuracy of the solution remains the same.

Two sets of N values are selected to assess the order of accuracy of the solution: (i) N_{aligned} = [32, 64, 128, 256, 512,1024] which aligns the fluid-solid interface located at $x = \pi$ with the Cartesian cell face, as done in [1], and (ii) $N_{\text{non-aligned}}$ = [25, 75, 225, 675, 2025] which does not. The other two fluid-solid interfaces at $x = 0$ and $x = 2\pi$ are located on grid cell faces by construction. Fig. 2.5 compares the spatial convergence rate for the two grid setups. As observed in Fig. 2.5(A), when the interface aligns with the Cartesian grid face, $\mathcal{O}(h^2)$ convergence rate is obtained using both continuous and discontinuous indicator functions. Second-order spatial accuracy is also obtained in Sakurai et al. [1] using the discontinuous indicator function using a similar grid setup. However, the order of accuracy degrades to $\mathcal{O}(h^1)$ when the interface is not aligned with the grid, as observed in Fig. 2.5(B). The authors in [1] did not present the order of accuracy results using a non-conforming grid (to the interface) for this problem. Finally, Fig. 2.5(C) shows the numerical solution q, and compares it against the exact solution for $N = 256$ grid. An excellent agreement is obtained.

2.7.2 Analysis of 1D Poisson equation with different inhomogeneous Neumann boundary conditions

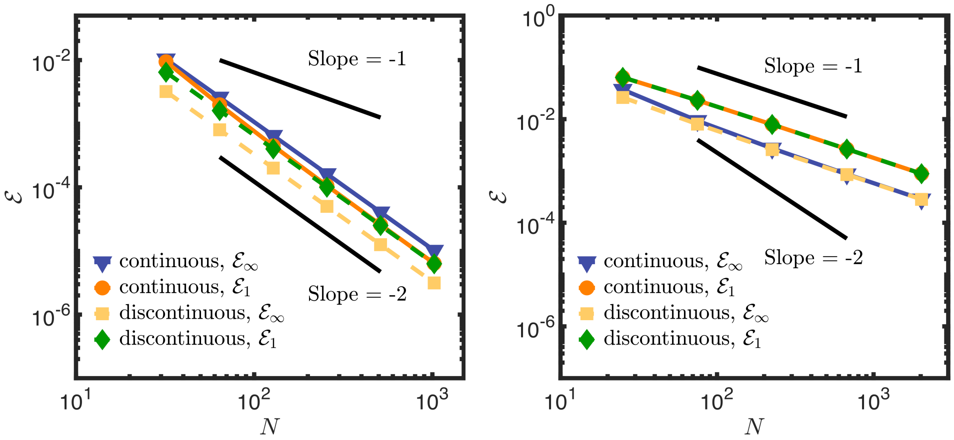
We now consider the 1D Poisson problem with different inhomogeneous Neumann boundary conditions on the two ends of the fluid domain, as done in Sec. 2.3 of Sakurai et al. The forcing function $f(\mathbf{x})$ for this case is

$$f(x) = m^2 \sin(mx), \tag{2.41}$$

and the inhomogeneous Neumann boundary condition values on the two ends are

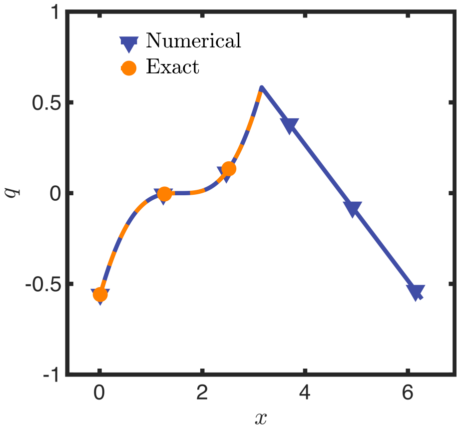
$$\left.\frac{\mathrm{d}q}{\mathrm{d}x}\right|_{x=0} = \alpha + m \quad \text{and} \quad \left.\frac{\mathrm{d}q}{\mathrm{d}x}\right|_{x=\pi} = \alpha - m. \tag{2.42}$$

Here, α and m parameters are taken to be 1. The problem setup remains the same as shown in Fig. 2.4. The analytical solution of this problem (using a zero-mean condition on q in Ω_f) reads



(A) Convergence rate using interface conforming grid

(B) Convergence rate using interface non-conforming grid



(C) Solution

Figure 2.5. 1D Poisson problem with same flux boundary conditions. Error norms $\mathcal{E}_1$ and $\mathcal{E}_\infty$ as a function of grid size N using continuous (solid line with symbols) and discontinuous (dashed line with symbols) indicator functions when (A) the fluid-solid interface at $x = \pi$ is aligned with the Cartesian cell face, and (B) when it is not; (C) numerical solution q obtained using $N = 256, \alpha = 1$, and $m = 1$, along with the exact solution. The penalization parameter η is taken as 10^{-8}.

as

$$q_{\text{exact}}(x) = \sin(mx) + \alpha x - \frac{2}{m\pi} - \frac{\pi\alpha}{2}. \tag{2.43}$$

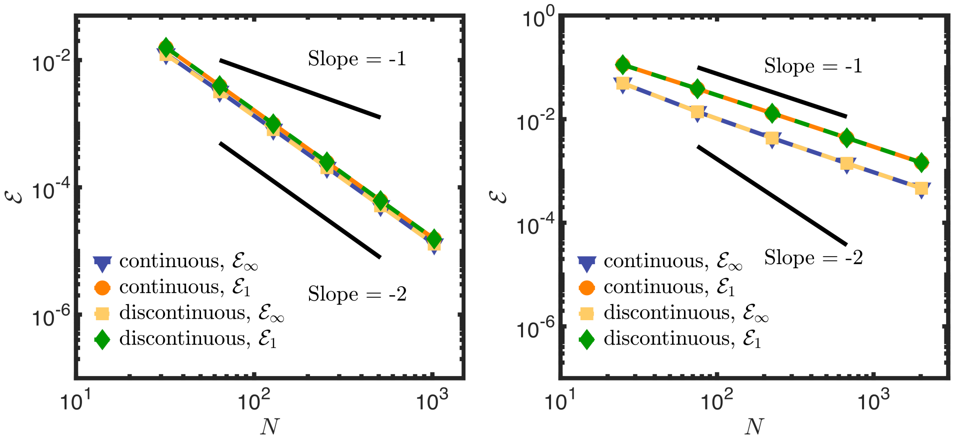
The flux forcing function is taken to be $\boldsymbol{\beta} = \nabla q_{\text{exact}} = (m\cos(mx) + \alpha)\,\hat{\mathbf{i}}$, which also satisfies the boundary conditions written in Eq. (2.42). We again replace the first linear equation with zero mean of q in Ω to obtain the unique solution. The results for this case are presented in Fig. 2.6 for N_{aligned} and $N_{\text{non-aligned}}$ grid size values, as taken in the previous section.

In Fig. 2.6(A) we observe $\mathcal{O}(2)$ convergence for both types of indicator functions when the interface aligns with the Cartesian grid face. This is in contrast to Sakurai et al. [1] where $\mathcal{O}(h^1)$ convergence is reported for this test problem using a similar grid setup; the results reported in [1] are not reproducible despite the use of same discretization method and problem setup. The authors in [1] attribute the reduction in accuracy to different values of flux boundary condition, which is clearly not the case here. Fig. 2.6(B) shows the order of accuracy results when the interface is not aligned with the Cartesian cell face — $\mathcal{O}(h^1)$ convergence rate is exhibited using both continuous and discontinuous indicator functions. Finally, Fig. 2.6(C) shows the numerical solution q, and compares it against the exact solution for $N = 256$ grid. An excellent agreement is obtained.

The results presented in the above two sections may suggest that the spatial accuracy of the flux-based VP method is $\mathcal{O}(h^2)$ when the interface aligns the Cartesian mesh, but degrades to $\mathcal{O}(h^1)$ when it does not. This is also one of the conclusions in Sakurai et al. However, our next examples contradict this conclusion.

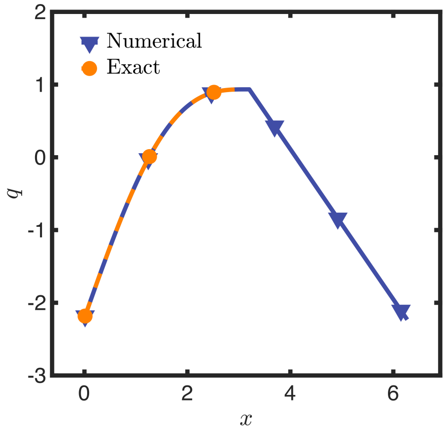
2.7.3 Concentric circular annulus with spatially constant flux on the interface

We consider the concentric circular annulus problem from Sakurai et al. [1] in which different inhomogeneous, but spatially constant, Neumann boundary conditions are specified on the two interfaces defining the annulus. The inner radius of the annulus is $r_i = \pi/4$ and its outer radius is $r_o = 3\pi/4$. The center of the annulus is positioned at (π, π). The circular annulus is



(A) Convergence rate using interface conforming grid

(B) Convergence rate using interface nonconforming grid



(C) Solution

Figure 2.6. 1D Poisson problem with different flux boundary conditions. Error norms $\mathcal{E}_1$ and $\mathcal{E}_\infty$ as a function of grid size N using continuous (solid line with symbols) and discontinuous (dashed line with symbols) indicator functions when (A) the fluid-solid interface at $x = \pi$ is aligned with the Cartesian cell face, and (B) when it is not; (C) numerical solution q obtained using $N = 256$, $\alpha = 1$, and $m = 1$, along with the exact solution. The penalization parameter η is taken as 10^{-8}.

embedded into a larger computational domain of extents $\Omega \in [0, 2\pi]^2$. The source term of the Poisson equation for this case is

$$f(r) = 16\cos(4r) + \frac{4\sin(4r)}{r}, \tag{2.44}$$

in which $r = \sqrt{(x-\pi)^2 + (y-\pi)^2}$ and the Neumann boundary condition values on the two interfaces are taken to be

$$\left.\frac{\mathrm{d}q}{\mathrm{d}r}\right|_{r=\frac{\pi}{4}} = 3m \quad \text{and} \quad \left.\frac{\mathrm{d}q}{\mathrm{d}r}\right|_{r=\frac{3\pi}{4}} = m. \tag{2.45}$$

The exact solution of this problem using the zero-mean condition $\int_{\Omega_\mathrm{f}} rq(r)\ \mathrm{d}r = 0$ reads as

$$q_{\text{exact}}(r) = \cos(4r) + \frac{3}{4}m\pi\log(r) - \frac{3}{32}m\pi\left(9\log\left(\frac{3}{4}\pi\right) - \log\left(\frac{\pi}{4}\right) - 4\right). \tag{2.46}$$

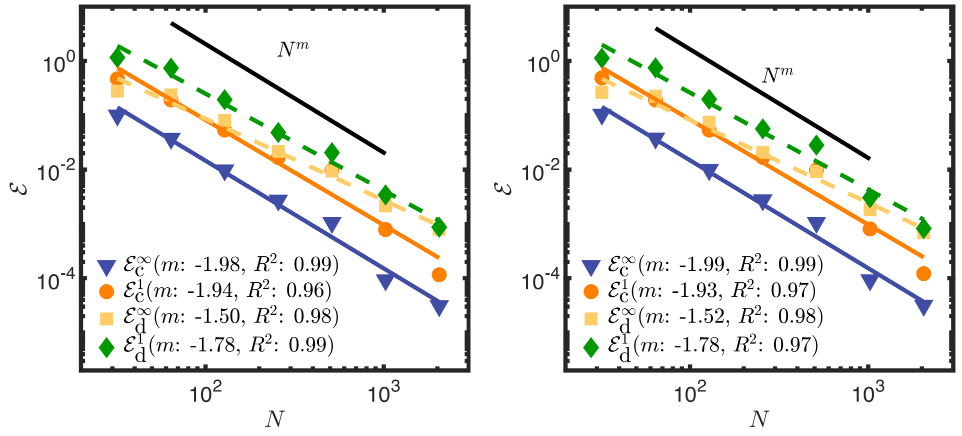
The mean of the numerical solution in the fluid region is subtracted as a post-processing step to impose the zero-mean condition numerically.

Relatively simple geometry and constant flux boundary conditions of this test problem allows for an analytical construction of $\boldsymbol{\beta}$. Indeed, in [1, 46], the flux-forcing function $\boldsymbol{\beta}$ was constructed analytically (Approach A) as $\boldsymbol{\beta} = \kappa\, \nabla\widetilde{q} = \frac{\mathrm{d}\widetilde{q}}{\mathrm{d}r}\mathbf{e}_r$, in which $\mathbf{e}_r = \left(\frac{x-\pi}{r}, \frac{y-\pi}{r}\right)$ and $\frac{\mathrm{d}\widetilde{q}}{\mathrm{d}r}$ is

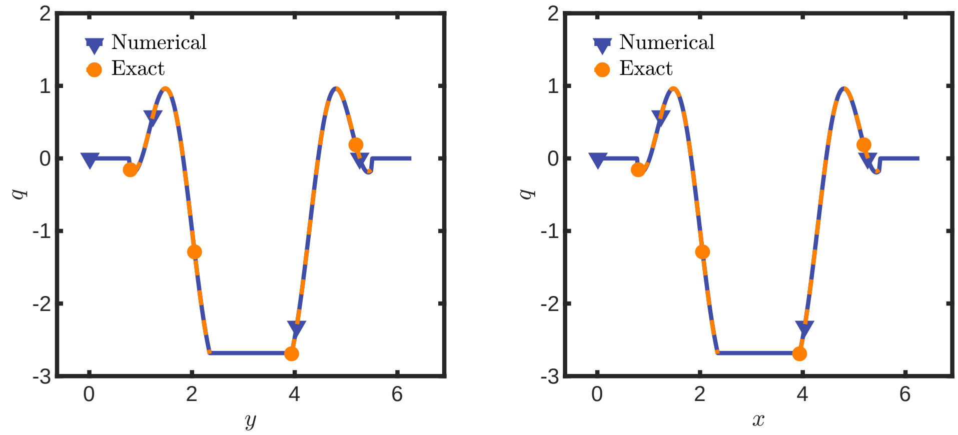
$$\frac{\mathrm{d}\widetilde{q}}{\mathrm{d}r} = \begin{cases} m\left(\frac{4r}{3\pi}\right)^2\left(4\left(1-\frac{r}{\pi}\right)\right)^3, & \text{if} \quad 0 \le r \le \pi, \\ 0, & \text{otherwise.} \end{cases} \tag{2.47}$$

It is to be noted that $\frac{\mathrm{d}\widetilde{q}}{\mathrm{d}r}$ reduces to $3m$ at $r = \pi/4$ and m at $r = 3\pi/4$, respectively.

Since the flux boundary condition value is spatially constant on the interface, Approach B is also applicable for this test problem. The g value required for Approach B is m and $-3m$ on the inner and outer interface, respectively. Figs. 2.7(A) and 2.7(B) show the order of accuracy of the solution as a function of mesh resolution for Approach A and B, respectively. As can

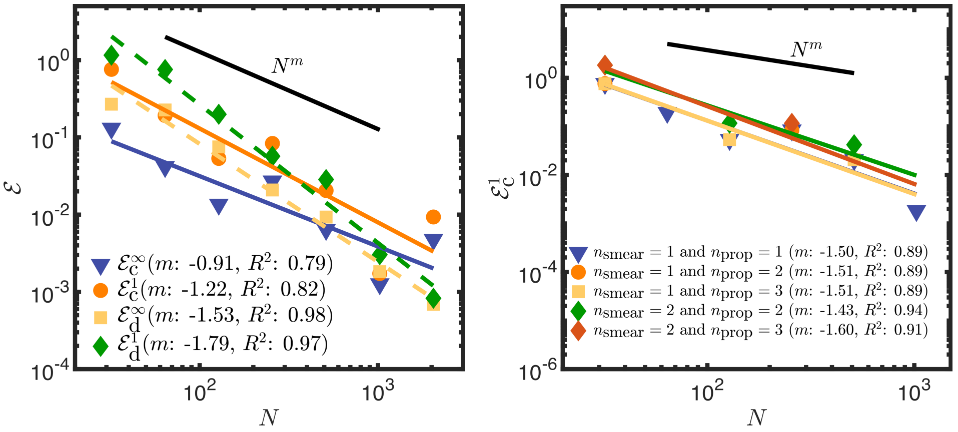


(A) Spatial convergence rate for Approach A (B) Spatial convergence rate for Approach B



(C) Solution variation along $y-$direction (D) Solution variation along $x-$direction

Figure 2.7. Concentric circular annulus with constant flux on the two interfaces using Approach A and B. Error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the continuous (solid lines with symbols) and discontinuous (dashed lines with symbols) indicator functions for (A) Approach A; and (B) Approach B. The penalization parameter η is taken as 10^{-8} and m and κ are taken as 1. (C) Variation of the numerical solution along $y-$direction at a fixed $x = 3.12$ location; and (D) variation of the numerical solution along $x-$direction at a fixed $y = 3.12$ location using $N = 256$ grid.



(A) Spatial convergence rate for Approach C (B) Effect of number of smear width and propagation cells

Figure 2.8. Concentric circular annulus with constant flux on the two interfaces using Approach C. Error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the continuous (solid lines with symbols) and discontinuous (dashed lines with symbols) indicator functions. (A) Convergence rate using $n_{\rm smear} = 1$ and $n_{\rm prop} = 2$. (B) Effect of $n_{\rm smear}$ and $n_{\rm prop}$ on the solution accuracy.

be observed in Fig. 2.7(A), for Approach A, $\mathcal{O}(h^{1.98})$ (respectively, $\mathcal{O}(h^{1.94})$) convergence rate with an R^2 value of 0.99 (respectively, 0.96) in L^∞ (respectively, L^1) norm is achieved using the continuous indicator function. With the discontinuous χ, $\mathcal{O}(h^{1.50})$ (respectively, $\mathcal{O}(h^{1.78})$) convergence rate with an R^2 value of 0.98 (respectively, 0.99) in L^∞ (respectively, L^1) norm is achieved. We note that the convergence rate using the continuous indicator function is better than the discontinuous function for Approach A. Looking at Fig. 2.7(B), it is seen that Approach B exhibits a very similar convergence rate as Approach A, but in contrast to Approach A, Approach B is more versatile as it requires only $\nabla\phi(\mathbf{x})$ information, which can be constructed for any irregular boundary [48]. Figs. 2.7(C) and 2.7(D) compare the analytical and numerical solutions along $x-$ and $y-$direction, respectively. Numerical solutions using Approach B and discontinuous indicator function are presented. As can be observed in the figures, an excellent agreement is obtained between the analytical and numerical solutions.

Next, we solve this problem using Approach C. As can be seen in Fig. 2.8(A), Approach C also exhibits a very similar convergence rate as Approach A and B when the discontinuous indicator function is used, whereas the convergence rate is reduced when the continuous function is employed. Specifically, $\mathcal{O}(h^{0.91})$ (respectively, $\mathcal{O}(h^{1.22})$) convergence rate with an R^2 value of 0.79 (respectively, 0.82) in L^∞ (respectively, L^1) norm is obtained using the continuous χ. However, Approach C is the most general one, since it can be used for imposing spatially varying flux values as demonstrated in later examples.

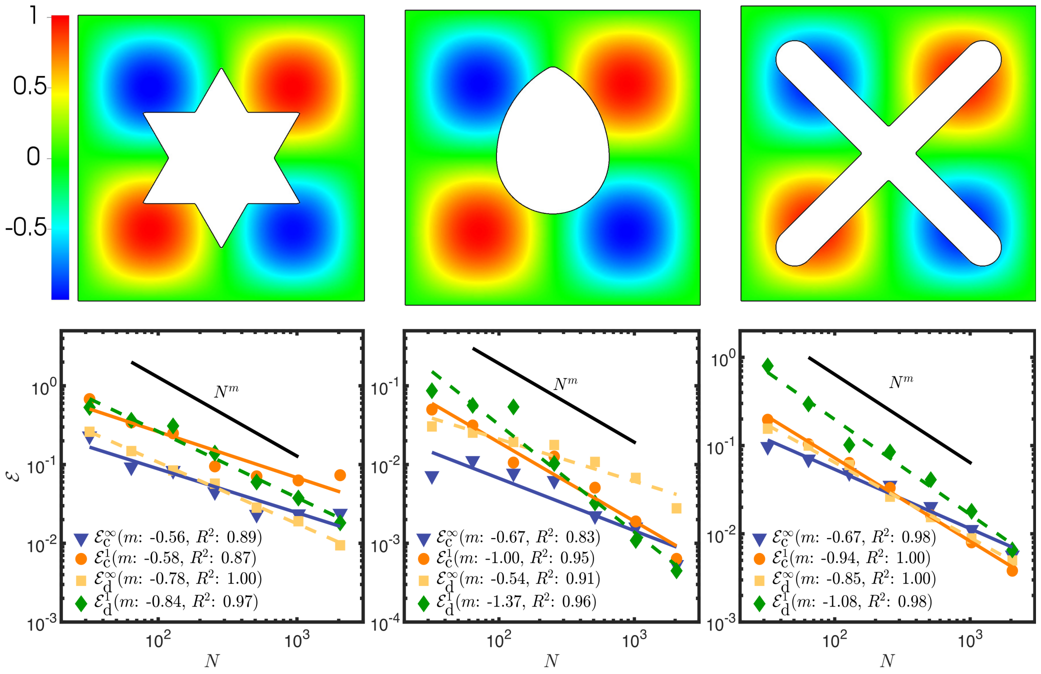
The present example is also used to study the effect of the number of propagation cells n_{prop} on the solution accuracy for the continuous indicator function. The results are shown in Fig. 2.8(B), in which it can be observed that 2 cells on either side of the interface are sufficient for propagating g values for a fixed number of n_{smear} cells; the error norms are mostly affected by the n_{smear} choice. Based on the results of this problem, we choose $n_{\text{smear}} = 1$ and $n_{\text{prop}} = 2$ for the continuous masking function, unless otherwise stated. For the discontinuous indicator function also, we use $n_{\text{prop}} = 2$ for the remainder of the problems (although $n_{\text{prop}} = 1$ is also sufficient).

2.7.4 Spatially varying flux values along complex interfaces

In this section, we assess the accuracy of the numerical solution for spatially varying flux values using a manufactured solution of the form

$$q_{\text{exact}}(\mathbf{x}) = \sin(x)\sin(y). \tag{2.48}$$

Inhomogeneous Neumann boundary conditions $g(\mathbf{x}) = -\kappa \; \mathbf{n} \cdot \nabla q_{\text{exact}}$ are imposed on the fluid-solid interface $\partial\Omega_{\text{s}}$, whereas Dirichlet boundary conditions are imposed on the external boundaries of the computational domain, i.e., $q|_{\partial\Omega(\mathbf{x})} = q_{\text{exact}}(\mathbf{x})$. Note that $g(\mathbf{x})$ varies spatially, and therefore, Approach B is not applicable for this test. Eq. (2.48) is plugged into the non-penalized Poisson Eq. (2.1) to generate the required source term $f(\mathbf{x})$. We consider three geometrically complex solid domains: a hexagram, an egg, and a x-cross; these geometries are embedded in a larger Cartesian domain of extents $\Omega \in [0, 2\pi]^2$ and the numerical solutions are computed in the corresponding fluid domains $\Omega_{\text{f}} = \Omega \setminus \Omega_{\text{s}}$.



(A) (B) (C)

Figure 2.9. Numerical solution q (top row) and the error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N (bottom row) for three complex shapes: (A) Hexagram; (B) Egg; and (C) X-cross using Approach C. The convergence rate results are shown for both continuous (solid line with symbols) and discontinuous (dashed line with symbols) indicator functions. The penalization parameter η is taken as 10^{-8} and κ is taken to be 1.

Using Approach A for constructing the flux-forcing function for this problem, wherein q_{exact} was used to define $\boldsymbol{\beta} = \kappa \, \nabla q_{\text{exact}}$ results in second-order convergence rate of the numerical solution for this problem; refer [46]. We solve the same problem using Approach C. Fig. 2.9 presents the numerical solution and its convergence rate as a function of grid resolution. For the hexagram case, at least $\mathcal{O}(h^{0.56})$ accuracy is achieved using the continuous indicator function whereas at least $\mathcal{O}(h^{0.78})$ accuracy is achieved using the discontinuous indicator function. Similarly, for the egg case, at least $\mathcal{O}(h^{0.67})$ accuracy is achieved using the continuous χ and at least $\mathcal{O}(h^{0.54})$ accuracy is obtained using the discontinuous indicator function. Lastly, for the x-cross geometry, Approach C exhibits at least $\mathcal{O}(h^{0.67})$ accuracy using the continuous indicator function and at least $\mathcal{O}(h^{0.85})$ accuracy with the discontinuous one. As noted in the previous section also, Approach C with the discontinuous indicator function is able to achieve a better convergence rate than with the discontinuous one. We remark that for Approach C, the reduction in accuracy (when compared to Approach A) is attributed to the codimension-0 extension of the spatially varying g function in the neighborhood of the interface. Nevertheless, Approach C is able to impose spatially varying flux values on a complex interface (sharp corners, etc.) and the solution accuracy is also reasonable. Later in Sec. 2.7.8 we demonstrate that smoothing of geometric features like sharp corners improves the accuracy of Approach C further.

2.7.5 Constant and spatially varying flux on three-dimensional interfaces

In this section, we consider two complex geometries in three spatial dimensions: a sphere and a torus. These geometries are embedded into a larger computational domain of extents $\Omega \in [0, 2\pi]^3$.

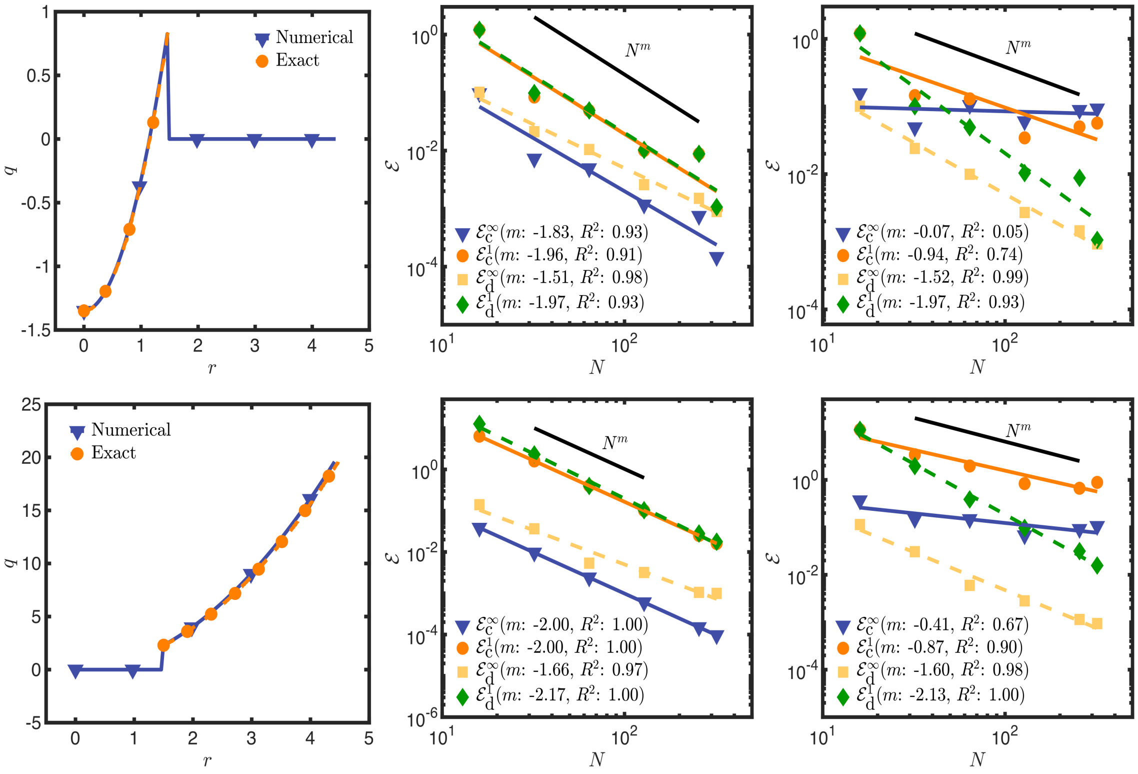
For the spherical geometry, the manufactured solution is taken to be

$$q_{\text{exact}}(r) = r^2 + c, \tag{2.49}$$

in which $r = \sqrt{(x-\pi)^2 + (y-\pi)^2 + (z-\pi)^2}$ and c is a constant. Eq. (2.49) when plugged into the non-penalized Poisson Eq. (2.1) yields a constant source term $f(\mathbf{x}) = -6$. The radius of the sphere is taken to be $R = 3/2$. Two cases are considered for the spherical geometry: fluid inside the sphere and fluid outside it. For the first case, a constant flux boundary condition $g = -\kappa\ \partial q_{\text{exact}}/\partial n = -2R$ is imposed on the spherical surface and a homogeneous Dirichlet boundary condition is imposed on the external domain boundary $\partial\Omega$. Since the solution of this Poisson problem is defined up to an additive constant c, we use the zero-mean condition $\int_0^R 4\pi r^2\ q_{\text{exact}}\ \mathrm{d}r = 0$ to determine the constant $c = \frac{-3R^2}{5}$. For the second case, in which the fluid is considered between the spherical interface and the computational domain boundary, constant flux boundary condition $g = -\kappa \partial q_{\text{exact}}/\partial n = -2R$ is imposed on the spherical interface and inhomogeneous Dirichlet boundary conditions $q|_{\partial\Omega(\mathbf{x})} = q_{\text{exact}}(\mathbf{x})$ are imposed on the external boundary. The constant c is taken to be zero for this case.

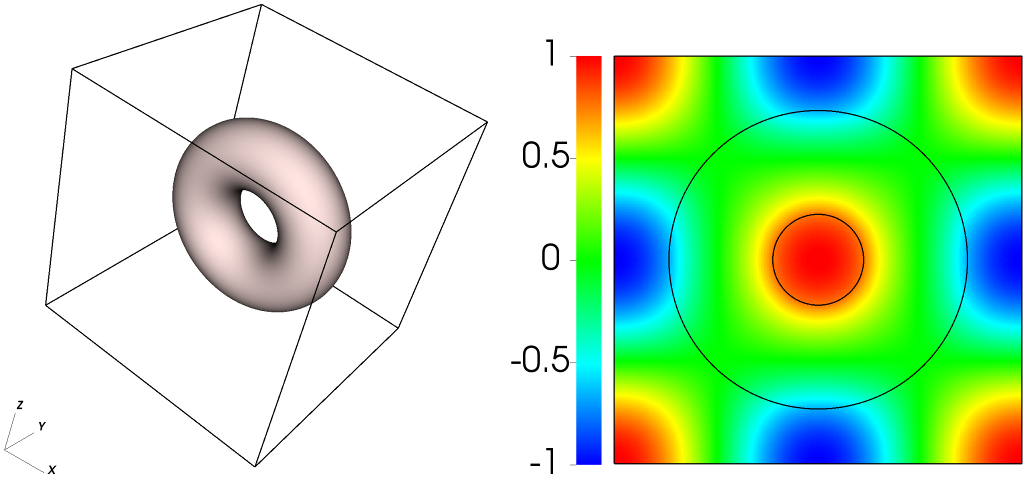
The comparison between the numerical and analytical solutions, as well as the spatial convergence rate of $\mathcal{E}^1$ and $\mathcal{E}^\infty$ error norms using Approach B and C are shown in Fig. 2.10. As can be observed in the figure, the numerical solution is in excellent agreement with the exact solution. Largely $\mathcal{O}(h^2)$ convergence rate is obtained for this example using Approach B with both continuous and discontinuous indicator functions. Approach C also yields the same order of accuracy with the discontinuous indicator function as Approach B. However, the convergence rate using the continuous indicator function is between 0 and 1 for Approach C. Clearly, the discontinuous indicator function performs better than the continuous one for Approach C. Second-order convergence rate is also obtained with Approach A, when $\boldsymbol{\beta} = \kappa\ \nabla q_{\text{exact}}$ is used for the spherical geometry (data not shown for brevity). Better performance of Approach A compared to Approach B and C is expected, as mentioned in the beginning of Sec. 2.7.

For the next three-dimensional test example, a solid torus is embedded in a computational domain of extents $\Omega \in [0, 2\pi]^3$ as shown in Fig. 2.11(A) and the fluid region is taken outside of

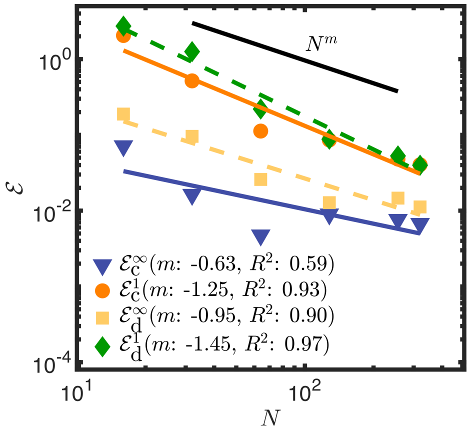


(A) (B) (C)

Figure 2.10. Spherical interface with constant flux boundary condition using Approach B and C. Top row corresponds to fluid inside the sphere case and the bottom row corresponds to fluid outside the sphere case. Also shown are the error norms, $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using continuous (solid lines with symbols) and discontinuous (dashed lines with symbols) indicator functions. (A) Comparison of numerical solution at $z = \pi$ using $N = 256$ grid. Error norms using (B) Approach B; and (C) Approach C. The penalization parameter η is taken as 10^{-8}, κ as 1 and the radius of the sphere is 3/2.



(A) Toroidal interface (B) Numerical solution at $y = \pi$



(C) Spatial convergence rate for Approach C

Figure 2.11. Toroidal interface with spatially varying flux boundary conditions using Approach C. (A) Zero-contour of the solid torus; (B) numerical solution at $y = \pi$ using $N = 256$ grid; and (C) the error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the continuous (solid lines with symbols) and discontinuous (dashed lines with symbols) indicator functions. The penalization parameter η is taken as 10^{-8} and κ is taken to be 1.

the torus. We consider a manufactured solution of the form

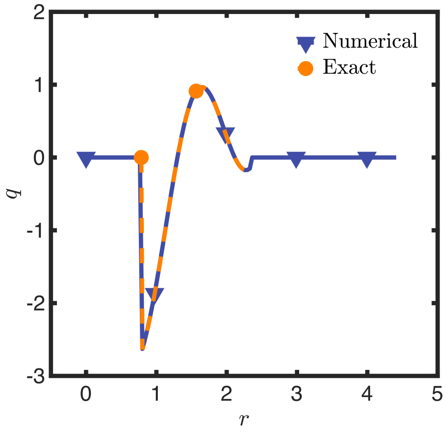
$$q_{\text{exact}}(\mathbf{x}) = -\cos(x)\cos(y)\cos(z). \tag{2.50}$$

Eq. (2.50) is plugged into the non-penalized Poisson Eq. (2.1) to generate the required source term $f(\mathbf{x})$. On the toroidal interface, spatially varying inhomogeneous Neumann boundary conditions are imposed, whereas on the external domain boundary inhomogeneous Dirichlet boundary conditions using the exact solution are imposed. We solve this test problem using Approach C and the results are shown in Fig. 2.11. As can be observed in Fig. 2.11, at least $\mathcal{O}(h^{1.25})$ is achieved with the discontinuous indicator function and at least $\mathcal{O}(h^{0.63})$ is achieved with the continuous indicator function.

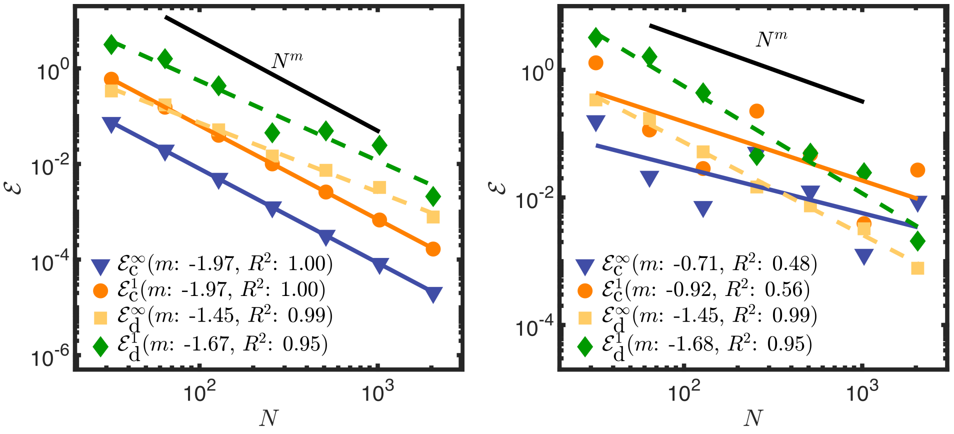
2.7.6 Spatially constant Robin boundary condition on two-dimensional interfaces

We consider the concentric circular annulus problem of Sec. 2.7.3 with the same exact solution $q_{\text{exact}}(\mathbf{r})$ and source term $f(r)$, as written in Eqs. (2.46) and (2.44), respectively. Plugging the exact solution into the Robin boundary condition Eq. 2.18 yields a spatially constant g value for the inner and outer interface, respectively.

The VP Poisson Eq. 2.19 is solved using Approach B and C for this problem. The numerical solution compared against Eq. 2.46 using Approach C is shown in Fig. 2.12(A); an excellent agreement is obtained. We also present the convergence rate for Approach B and C in Fig. 2.12. As can be seen in Fig. 2.12, the convergence rates obtained by using the discontinuous indicator function for Approach B and C are quite close to what we had obtained in Sec. 2.7.3. For the continuous indicator function, we obtain approximately second-order accuracy with Approach B and at least $\mathcal{O}(h^{0.71})$ accuracy with Approach C. As also observed in Sec. 2.7.3, the discontinuous indicator function performs better than the continuous one with Approach C; the reverse is true for Approach B.

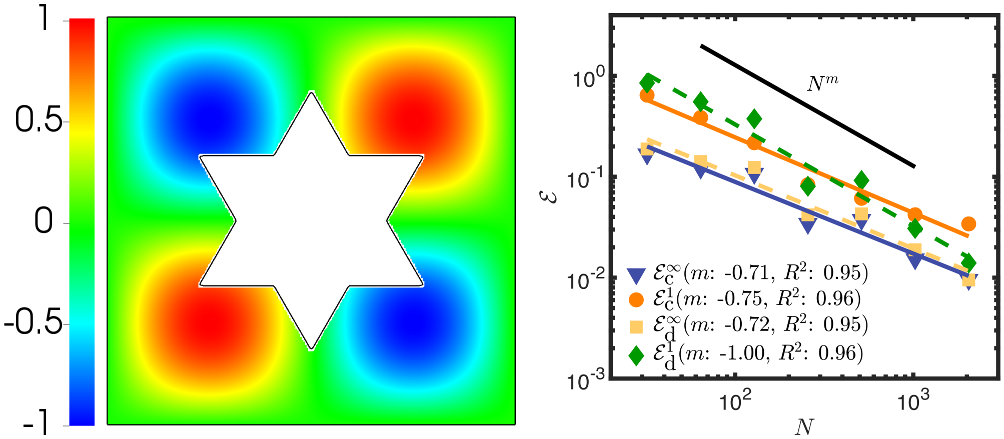


(A) Numerical solution



(B) Order of convergence using Approach B (C) Order of convergence using Approach C

Figure 2.12. Concentric annulus with spatially constant Robin boundary conditions using Approach B and C. (A) Numerical solution using $N = 256$ grid. Error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the continuous (solid lines with symbols) and discontinuous (dashed lines with symbols) indicator functions for (B) Approach B; and (C) Approach C. The penalization parameter η is taken as 10^{-8}. The values of κ and ζ are taken to be 1.



(A) Numerical solution

(B) Order of convergence

Figure 2.13. Hexagram domain with spatially varying Robin boundary conditions using Approach C. (A) Numerical solution using $N = 256$ grid; and (B) the error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the continuous (solid lines with symbols) and discontinuous (dashed lines with symbols) indicator functions. The penalization parameter η is taken as 10^{-8}. The values of κ and ζ are taken to be 1.

2.7.7 Spatially varying Robin boundary condition on a complex two-dimensional interface

In this section, we assess the accuracy of Approach C for spatially varying Robin boundary conditions on a complex two-dimensional interface. A hexagram geometry is embedded into a computational domain of extents $\Omega \in [0, 2\pi]^2$, as considered in Sec. 2.7.4. The fluid is occupied between the computational domain boundary $\partial\Omega$ and the fluid-solid interface $\partial\Omega_s$. The same manufactured solution as written in Eq. 2.48 is considered here; this solution yields spatially varying g values when plugged into the Robin boundary condition Eq. (2.18). We solve the VP Poisson Eq. 2.19 using Approach C. The numerical solution and the spatial convergence rate of the error norms are presented in Fig. 2.13. As observed in the figure, at least $\mathcal{O}(h^{0.71})$ accuracy is achieved with the continuous indicator function and at least $\mathcal{O}(h^{0.72})$ accuracy is achieved with the discontinuous indicator function.

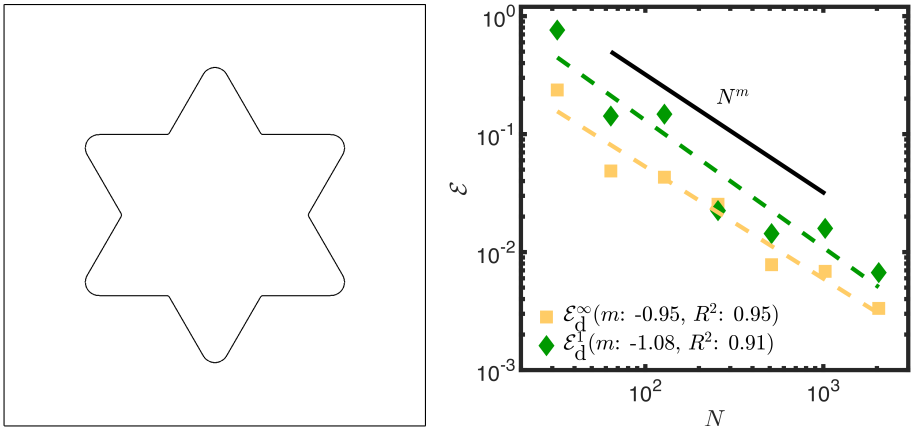
2.7.8 Effect of smoothing geometric features on the convergence rate of Approach C

In this section, we study the effect of sharp geometric features, such as corners on the convergence rate of Approach C. We consider a slightly modified version of the hexagram interface, which was considered earlier in Secs. 2.7.4 and 2.7.7. Instead of the sharp corners, in this case, we embed a hexagram having smooth exterior corners (see Fig. 2.14(A)) in a larger Cartesian domain of extents $\Omega \in [0, 2\pi]^2$ and solve the Neumann/Robin problem using Approach C. The order of accuracy results for spatially varying Neumann and Robin boundary conditions using the discontinuous indicator function are presented in Fig. 2.14. For the sharp corner geometry case with spatially varying Neumann boundary conditions (as shown in Fig 2.9(A)), the convergence rates were $\mathcal{O}(h^{0.78})$ and $\mathcal{O}(h^{0.84})$ in L^∞ and L^1 norm, respectively. In contrast, with smooth corners, the convergence rates are $\mathcal{O}(h^{0.95})$ and $\mathcal{O}(h^{1.08})$ in L^∞ and L^1 norm, respectively. A similar trend is obtained when spatially varying Robin boundary conditions are considered: For the sharp geometry case (as shown in Fig. 2.13), the convergence rates were $\mathcal{O}(h^{0.71})$ and $\mathcal{O}(h^{1.00})$ in L^∞ and L^1 norm, respectively. With smooth corners, the convergence rates are $\mathcal{O}(h^{1.00})$ and $\mathcal{O}(h^{1.26})$ in L^∞ and L^1 norm, respectively. This test demonstrates that the convergence rate of Approach C also depends upon local geometric features.

2.7.9 Application to free convection problem

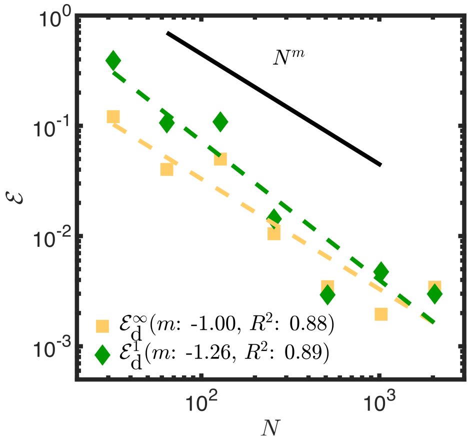
Finally, we consider steady natural convection in a concentric annulus. A constant heat flux Q is imposed on the inner cylinder of radius r_i and a fixed temperature T_o is maintained on the outer cylinder of radius r_o. The concentric annulus is embedded into a larger computational domain of extents $\Omega \in [-2.56, 2.56]^2$, as shown in Fig. 2.15. This example was studied in Yoo [66] using a body-fitted grid approach and more recently, it has been used to validate the IB/FCT relying on time-splitting approach to handle the flux boundary conditions on embedded interfaces [55, 56, 57]; see Introduction Sec. 2.1 for a brief discussion on IB/FCT.

We solve the volume penalized advection-diffusion equation for the temperature field



(A) Smoothed hexagram geometry

(B) Order of convergence for spatially varying Neumann boundary condition



(C) Order of convergence for spatially varying Robin boundary condition

Figure 2.14. Hexagram domain with smooth exterior corners. (A) Zero-contour of the smoothed hexagram interface. Error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the discontinuous indicator function with Approach C. (B) Spatially varying Neumann boundary conditions; and (C) spatially varying Robin boundary conditions. The penalization parameter η is taken as 10^{-8}. The values of κ and ζ are taken to be 1.

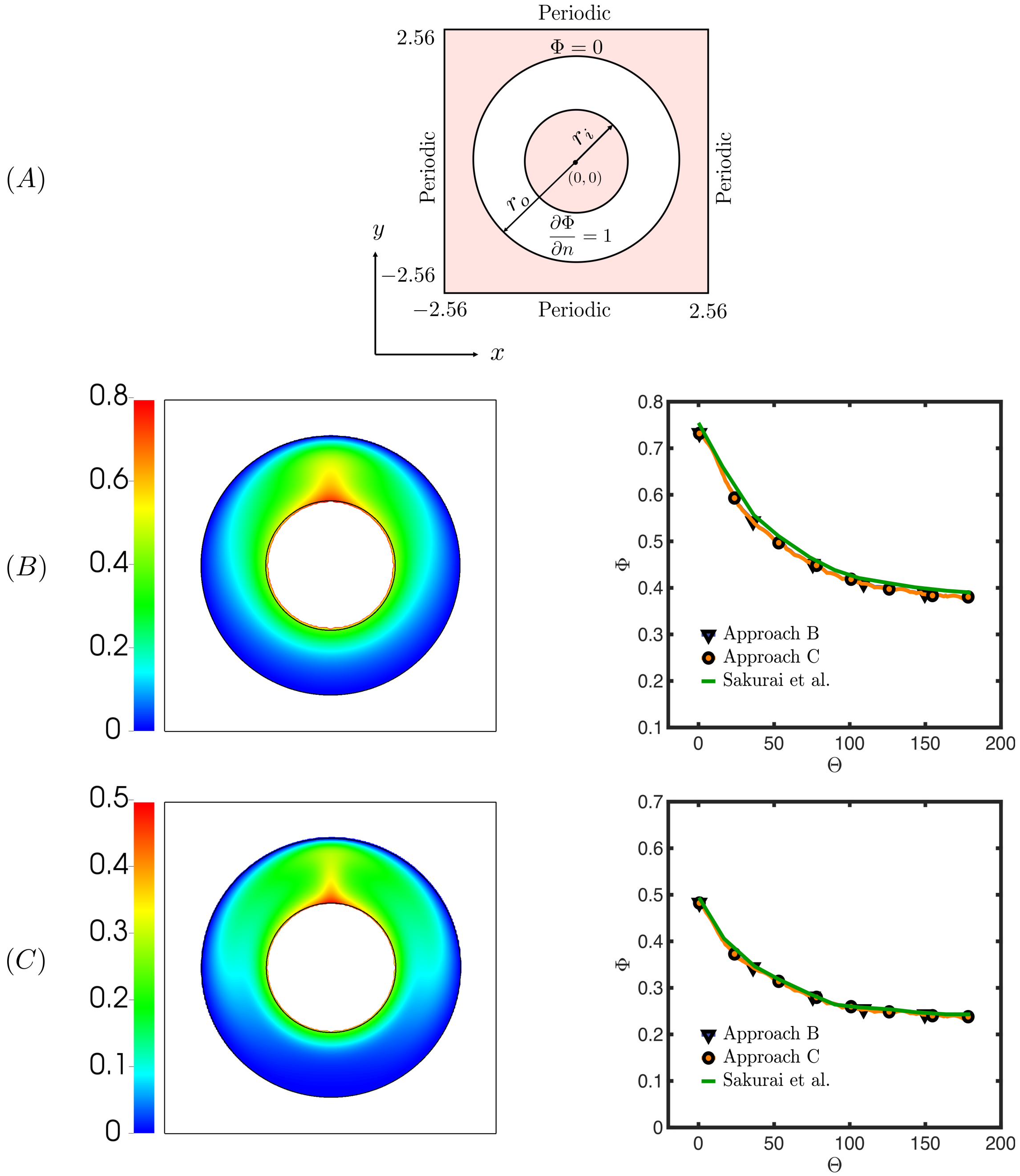


Figure 2.15. 2D free convection problem at two Rayleigh numbers: Ra = 5700 and Ra = 5×10^4.(A) Schematic of the problem. Steady state temperature field and temperature distribution on the surface of the left half of the inner cylinder at (B) Ra $= 5700$ and (C) Ra = 5×10^4.

coupled to the volume penalized incompressible Navier-Stokes equations (Eqs. (5.6)-(2.23)) in a non-dimensional form. The non-dimensional quantities are defined as: dimensionless temperature $\Phi^* = k(T - T_0)/(QL)$, velocity $\mathbf{u}^* = \mathbf{u}L/\alpha$, time $t^* = t\alpha/L^2$, and position $\mathbf{x}^* = \mathbf{x}/L$. Here, k is the thermal conductivity, α is the thermal diffusivity $\alpha = k/(\rho c_p)$, ρ is the density, c_p is the heat capacity at constant pressure, and $L = r_o - r_i$ is the annulus thickness. For this case we consider $r_i^* = r_i/L = 1$ and $r_o^* = r_o/L = 2$. Dropping the * superscript from the non-dimensional quantities, the system of non-dimensional equations reads as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \text{Pr}\nabla^2 \mathbf{u} + [1 - (\chi^{\text{d}} + \chi^{\text{n}})]\text{RaPr}\Phi \mathbf{e}_y - \frac{\chi^{\text{d}} + \chi^{\text{n}}}{\eta_{\text{d}}}\mathbf{u}, \quad (2.51)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.52)$$

$$\frac{\partial \Phi}{\partial t} + (1 - \chi^{\text{n}})(\mathbf{u} \cdot \nabla \Phi) = \nabla \cdot \left[\{(1 - \chi^{\text{n}}) + \eta_{\text{n}} \chi^{\text{n}}\} \nabla \Phi\right] + \nabla \cdot (\chi^{\text{n}} \boldsymbol{\beta}) - \chi^{\text{n}} \nabla \cdot \boldsymbol{\beta} - \frac{\chi^{\text{d}}}{\eta_{\text{d}}}\Phi. \quad (2.53)$$

Here, $\mathbf{e}_y = (0, 1)$ is a unit vector in the $y-$direction and the continuous indicator functions χ^{d} and χ^{n} are defined to be

$$\chi^{\text{d}} = \begin{cases} 1, & \phi^{\text{d}}(\mathbf{x}) < -n_{\text{smear}}\, h, \\ 1 - \frac{1}{2}\left(1 + \frac{1}{n_{\text{smear}} h}\phi^{\text{d}}(\mathbf{x}) + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{smear}} h}\phi^{\text{d}}(\mathbf{x})\right)\right), & |\phi^{\text{d}}(\mathbf{x})| \le n_{\text{smear}}\, h, \\ 0, & \text{otherwise}, \end{cases} \quad (2.54)$$

$$\chi^{\text{n}} = \begin{cases} 1, & \phi^{\text{n}}(\mathbf{x}) < -n_{\text{smear}}\, h, \\ 1 - \frac{1}{2}\left(1 + \frac{1}{n_{\text{smear}} h}\phi^{\text{n}}(\mathbf{x}) + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{smear}} h}\phi^{\text{n}}(\mathbf{x})\right)\right), & |\phi^{\text{n}}(\mathbf{x})| \le n_{\text{smear}}\, h, \\ 0, & \text{otherwise}. \end{cases} \quad (2.55)$$

In the above, $\phi^{\text{d}}(\mathbf{x})$ and $\phi^{\text{n}}(\mathbf{x})$ are the signed distance functions for the Dirichlet (outer cylinder) and Neumann (inner cylinder) boundary, respectively. The Rayleigh number Ra = $G\gamma Q L^4/(k\alpha\nu)$ and the Prandtl number Pr = ν/α are the two main non-dimensional parameters that characterize buoyancy-driven flows; these parameters are seen in the right-hand side of the non-dimensional momentum Eq. (2.51). Here, γ is the coefficient of thermal expansion, G is the gravitational constant, and ν is the kinematic viscosity. The flux-forcing function $\boldsymbol{\beta}$ imposing the constant flux boundary condition on the surface of the inner cylinder, $\mathbf{n}_i \cdot \nabla \Phi = 1$, is constructed using Approach B and C for this problem. Here, $\mathbf{n}_i$ is the unit outward normal vector of the inner cylinder. On the outer cylinder homogeneous Dirichlet boundary condition, $\Phi = 0$, is imposed through the last term of Eq. (2.53). Periodic boundary conditions are used on the external domain boundaries.

Two Rayleigh numbers Ra = 5700 and Ra = 5×10^4 are considered for this problem. The same Prandtl number Pr = 0.71 is used for the two cases. The computational domain Ω is discretized by a uniform Cartesian grid of size 256×256. The penalization parameters η_{d} and η_{n} are taken to be 10^{-8}. We treat the convective and the advective terms of Eqs. (2.51) and (2.53) explicitly, whereas the rest of the terms are treated implicitly. The implicit treatment of volume penalization terms in Eqs. (2.51) and (2.53) allows us to use a relatively large time step sizes of $\Delta t = 10^{-4}$ and $\Delta t = 5 \times 10^{-5}$ for Ra $= 5700$ and Ra $= 5 \times 10^4$ cases, respectively. In contrast, Sakurai et al. [1] used a time step size of $\Delta t = 10^{-6}$ for these two cases as they employed an explicit Euler time marching scheme. More details on the second-order accurate spatial discretization and time-stepping scheme employed in the fluid solver can be found in our prior works [59, 4].

To compare our results with those reported in [1] we plot the steady-state temperature distribution on the left half of the inner cylinder for both Ra cases in Fig. 2.15. In the figure, the polar angle $\Theta = 0°$ starts from the top position $(x, y) = (0, 1)$ of the inner cylinder and ends at its bottom position $(x, y) = (0, -1)$, where $\Theta = 180°$. As observed in the figure, the numerical results obtained using both Approach B and C are in excellent agreement with those reported in [1] who used Approach A for constructing $\boldsymbol{\beta}$. Sakurai et al. compared their numerical results with Yoo [66] and Ren et al. [55]; comparison with Yoo and Ren et al. is therefore omitted in Fig. 2.15 in the interest of clarity. We also present the steady-state temperature field in the whole annular domain at the two Rayleigh numbers in Fig. 2.15.

2.8 Conclusions

In this work, we proposed a numerical technique for constructing flux-forcing functions for the flux-based VP method introduced by Sakurai et al. We also extended the flux-based VP approach to include Robin boundary conditions. Our method of flux-forcing functions is more general than the analytical approach (denoted Approach A in this work) of Sakurai et al. and requires only a signed distance function to construct the flux-forcing function. Two numerical-based approaches were presented for constructing flux-forcing functions: Approach B for imposing spatially constant and Approach C for imposing spatially varying (as well spatially constant) Neumann/Robin boundary conditions. Within Approach C we extended the (spatially varying) codimension-1 g function to the neighborhood of the interface using a simple propagation strategy. We considered several two- and three-dimensional Poisson problems in complex domains to assess the accuracy of the numerical solutions. Results were presented for both continuous and discontinuous indicator functions. For Approach B, largely $\mathcal{O}(h^2)$ accuracy is observed using the continuous indicator function. Between $\mathcal{O}(h^1)$ and $\mathcal{O}(h^2)$ convergence rate is observed for Approach B with the discontinuous indicator function and a similar convergence rate is observed for Approach C with the discontinuous indicator function when it is used for solving the constant Neumann/Robin boundary condition problem. For spatially varying boundary conditions, Approach C using the discontinuous indicator function exhibits close to $\mathcal{O}(h^1)$ convergence rate; the accuracy of the method is further improved by smoothing the sharp geometric features. However, Approach C using the continuous indicator function exhibits a convergence rate between $\mathcal{O}(h^0)$ and $\mathcal{O}(h^1)$. Based on our results, we recommend using the discontinuous indicator function with Approach C to impose spatially varying Neumann/Robin boundary conditions and using the continuous indicator function with Approach B to impose spatially constant Neumann/Robin boundary conditions. Finally, we used Approach B and

C to study the flux-driven thermal convection problem in a concentric annulus and compared our results against the literature. An excellent agreement was obtained. We also provided formal derivation of the flux-based volume penalized Poisson equations in strong form for both Neumann and Robin problems. The formulation shows that an explicit construction of the delta function is not necessary for the flux-based VP method, which makes it different from other diffuse domain equations presented in the literature.

Acknowledgements

Chapter 2, in part, is a reprint of the material as it appears in Critique on "Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry". Thirumalaisamy, Ramakrishnan; Nangia, Nishant; Bhalla, Amneet Pal Singh, Journal of Computational Physics, vol. 433, 110163, 2021. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 2, in part, is a reprint of the material as it appears in Handling Neumann and Robin boundary conditions in a fictitious domain volume penalization framework. Thirumalaisamy, Ramakrishnan; Patankar, Neelesh A.; Bhalla, Amneet Pal Singh, Journal of Computational Physics, vol. 448, 110726, 2022. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3

An effective preconditioning strategy for volume penalized incompressible/low Mach multiphase flow solvers

In this chapter, we propose a projection method-based preconditioning strategy for solving volume penalized (VP) incompressible and low-Mach Navier-Stokes equations. The projection preconditioner enables the monolithic solution of the coupled velocity- pressure system in both single phase (uniform density and viscosity) and multiphase (variable density and viscosity) flow settings. In this approach, the penalty force is treated implicitly, which is allowed to take arbitrary large values without affecting the solver's convergence rate or causing numerical stiffness. It is made possible by including the penalty term in the pressure Poisson equation (PPE), which was not included in previous works that solved VP incompressible Navier-Stokes equations using the projection method. We show how and where the Brinkman penalty term enters the PPE by re-deriving the projection algorithm for the VP method. Solver scalability under grid refinement is demonstrated, i.e., convergence is achieved with the same number of iterations regardless of the problem size. A manufactured solution in a single phase setting is used to determine the spatial accuracy of the penalized solution. Various values of body's permeability are considered. Second-order pointwise accuracy is achieved for both velocity and pressure solutions for reasonably small values of permeability. Error saturation occurs when permeability is extremely small, but the convergence rate of the solver does not degrade. The solver converges faster as permeability decreases, contrary to prior experience. Two multiphase fluid-structure interaction (FSI) problems from the ocean engineering literature are also simulated to evaluate the solver's robustness and performance (in terms of its number of iterations). The proposed solver also allows us to investigate the effect of permeability on the motion of the contact line over the surface of the immersed body. It also allows us to investigate the dynamics of the free surface of a solidifying metal.

3.1 Introduction

The VP method was originally developed by Arquis and Caltagirone [67] to simulate isothermal obstacles in incompressible single-phase flows. In 1999, Angot et al. [7] provided convergence proofs and error estimates of the penalized solution in terms of the *penalty* parameter. Inspired by Brinkman's work [68], the VP technique treats solids embedded in a fluid as porous media with extremely low permeabilities $\kappa \ll 1$. The velocity boundary condition on the fluid-solid interface, which is of Dirichlet type, is imposed through a volumetric feedback force that is inversely proportional to the body's permeability κ. The feedback force is commonly referred to as the Brinkman penalty force. A number of extensions and improvements have been made to the VP method over the years due to its simplicity, robustness, ease of implementation, and ability to handle complex geometries. We have extended the VP method to allow spatially-varying Neumann and Robin boundary conditions over complex interfaces for advection-diffusion PDEs (see Chapter 2 and [46, 69]). In our VP method [46, 69], the parameter κ has no physical relationship to the solid's permeability, but rather is a numerical diffusion parameter that ensures flux continuity across the interface. To achieve high-order (up to fourth-order) spatial accuracy of the penalized solution, Kou et al. [70] used a combination of VP and high-order flux reconstruction techniques in a discontinuous Galerkin framework. VP is also used to model phase change problems, but under a different name, the Carman-Kozeny drag model [17, 18]. In melting/solidification problems, a volumetric penalty/drag force is applied to retard solid phase motion [18, 19]. In literature, volume penalization is most commonly used to model fluid-structure interaction (FSI). The technique has been extensively used to model FSI in incompressible single phase flows (uniform density and viscosity) [10, 11, 12]. The VP technique has also recently gained popularity in modeling multiphase FSI, such as solid motion in incompressible gas-liquid flows. In Bhalla et al. [41], water entry/exit problems were simulated and hydrodynamic loads were calculated for solid bodies slamming into air-water interfaces. Khedkar et al. [13, 14] modeled FSI and optimal control of wave energy converters. Sharaborin et al. [15] combined the VP method with a volume of fluid approach to model the prescribed motion of rigid bodies in gas-liquid flows. Bergmann used the VP method to investigate the hydrodynamics of a dolphin jumping out of water [16]. In addition, there are some alternative methods for modeling multiphase FSI that are worth mentioning. In recent years, sharp interface cut cell methods have been proposed to simulate solid-liquid-gas flows [71, 72]. They have the advantage of conserving mass and resolving three-phase triple points over VP methods. These methods also employ discontinuous interface tracking methods, such as the volume of fluid method to truncate the air-water interface at the solid surface. This is different from a fictitious domain/VP method that employs a continuous representation of the air-water interface, whose implications are discussed in Sec. 3.5.1 of this article. The implementation of cut cell methods requires more work, especially when considering three spatial dimensions, and special care must be taken when addressing non-prescribed motions of solid bodies and the intersection of multiple phases [72].

In spite of the fact that the VP method has been extensively studied to understand the accuracy of the penalized solution, there is no study (to our knowledge) that proposes efficient solvers for the volume penalized Navier-Stokes equations, particularly when the system becomes stiff as $\kappa \to 0$. Several studies have treated the Brinkman penalty term explicitly [73, 1] (i.e., using the prior time value of the penalty force) or via an operator-splitting approach [74, 10, 75, 76] (i.e., accounting the penalty force separately in a substep), but this limits the time step size Δt and/or κ values in order to avoid numerical stiffness/instability. An interesting linearizationbased technique was recently proposed by Kou et al. [70] to treat the stiff penalty term in an implicit-explicit manner in the context of volume penalized compressible flows, which allowed the authors to take κ one to two orders of magnitude lower than needed in an explicit treatment. As we aim to develop solver technology for arbitrary small κ values (e.g., four orders lower compared to that required in an explicit treatment) for incompressible flows in this thesis, we do not discuss the explicit and operator-splitting approaches further. This is needed in situations like modeling phase change problems, where it is necessary to experiment with increasingly large values of κ^{-1} in order to find out what is large enough to obtain physically correct phase change dynamic. Similarly, for multiphase FSI problems κ controls the contact line motion over the immersed surface, as demonstrated in this work. To alleviate the aforementioned issues related to numerical stiffness and instability, the stiff penalty term can be treated implicitly, and this is commonly accomplished by solving the penalized system using projection solvers in literature. Bergmann and Iolla [11] describe the projection method to solve the volume penalized single-phase incompressible Navier-Stokes (INS) system, and more recently, Sharaborin et al. [15] have extended and described the projection approach for volume penalized multiphase flows. Prior works [11, 15] have employed an incorrect projection algorithm to solve the volume penalized INS equations, which is corrected in this study.

An alternative to the projection solver is the monolithic velocity-pressure solver, which does not split velocity and pressure degrees of freedom. Monolithic flow solvers are believed to be computationally inefficient compared to velocity-pressure split solvers such as the projection solver. Griffith [77] and Cai et al. [78] point out that this is a misconception, and that the coupled velocity-pressure system can be solved as efficiently as the projection method. Moreover, the monolithic approach does not suffer from order of accuracy reductions caused by operator-splitting or artificial boundary conditions that are required in the projection algorithm. In 2009, Griffith proposed to use an *inexact* projection solver as a preconditioner for the GMRES/FGMRES solver to solve the single phase INS equations monolithically and efficiently. Cai et al. [78] extended the projection preconditioner approach to the variable density and viscosity

INS system in 2014. The authors considered periodic, free-slip, and velocity boundary conditions for the multiphase solver in their work [78]. Later in 2019, Nangia et al. [59] demonstrated that the multiphase projection preconditioner can also handle spatially and temporally varying traction boundary conditions. Based on the success of using projection method as a preconditioner for the coupled velocity-pressure system, this work extends the technique to volume penalized single and multiphase INS systems. The same preconditioner can also be used to solve volume penalized low Mach Navier-Stokes equations. Low Mach volume penalized systems, for instance, are used to model the melting/solidification of phase change materials undergoing volume changes (i.e., velocity is not divergence-free) during the phase change process. An example of this can be found in the motivating Sec. 3.2. It is interesting to note that some phase change problems can also be modeled under isothermal conditions; see for example Ahlkrona and Elfverson [79] and Löfgren [80], who simulate glacier melting due to the shear thinning of ice. Unlike Sec. 3.2, these works assume divergence-free velocity.

An obvious choice for the projection preconditioner (to solve the volume penalized INS system) is *inexact* versions of the projection solvers of Bergmann and Iolla [11] and Sharaborin et al. [15]. An inexact solver solves the system of equations only approximately, such as by using few iterations or setting a loose convergence tolerance. Empirical testing shows, however, that the projection algorithms of [11, 15] do not lead to robust convergence of the monolithic solver, particularly when κ is small. This is because prior projection algorithms have not taken into account the Brinkman penalty term in the pressure Poisson equation (PPE). By re-deriving the projection algorithm for the VP method, we show how and where the Brinkman penalty term enters the PPE. Using the correct projection algorithm, we are able to achieve robust convergence of the monolithic multiphase flow solver, even when the value of κ is very small. In addition, the solver is demonstrated to remain scalable under grid refinement, i.e., the number of iterations required to converge remains essentially the same regardless of the problem size. To test the spatial accuracy of the penalized solution, we consider a manufactured solution for a uniform density and viscosity flow. A wide range of κ values is considered in the test. For both velocity

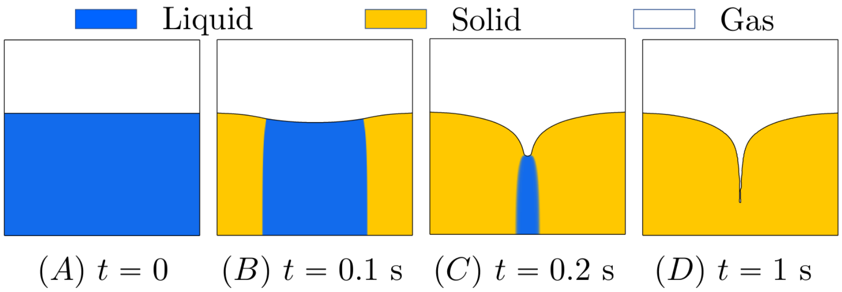


Figure 3.1. Solidification of liquid aluminum in presence of gas phase. As the liquid aluminum solidifies a pipe defect appears due to the volume shrinkage effect. The effect of the permeability parameter $\kappa = \epsilon \Delta t/(\rho^{\mathrm{S}})$ on the solidification dynamics is shown. Based on the results of this simulation $\kappa = \Delta t/(100\, \rho^{\mathrm{S}})$ or lower is suggested for this class of problems.

and pressure solutions, second-order pointwise accuracy is achieved for reasonably small values of κ. In the case of extremely small κ, error saturation occurs, but the convergence rate of the solver does not deteriorate. In fact, as κ values decrease, the proposed solver converges faster, contrary to prior experience [73, 1, 76, 74, 10], where the system becomes stiff (difficult to solve) at small values of κ. An additional test evaluates the solver's performance (in terms of iteration count) for a multiphase FSI problem with two- to three-order differences in density and viscosity. The convergence rates remain robust in the multiphase case as well. Finally, the proposed preconditioner also allows us to study the effect of κ on the contact line motion over the immersed surface.

3.2 Motivation behind the proposed solver and preconditioner

The motivation behind the development of the preconditioner proposed in this work comes from our efforts to model phase change problems involving melting/solidification of phase change materials (PCMs). Consider, for example, liquid aluminum/PCM (blue color) solidifying in a cast in the presence of air (white color) as illustrated in Fig. 3.1. The computational domain is taken to be a square of extents $\Omega \in [0, 8 \times 10^{-3}]^2$, which is discretized into $N \times N = 256 \times 256$ uniform cells.Temperature T is fixed on all boundaries at $0.5 T_m$ ($T_m = 933.6$ K is aluminum's solidification temperature), except at the bottom wall, where zero-heat flux is imposed. Initially, liquid aluminum has a temperature of $2T_m$, whereas gas has a temperature of $0.5T_m$. The imposed boundary conditions cause solidification to begin on the right and left sides of the domain. Solidification is not affected by the top open boundary (which is exposed to the atmosphere) since air has a low thermal conductivity k (gas, liquid, and solid conductivities are taken to be $k^{\mathrm{G}} = 6.1 \times 10^{-2}$, $k^{\mathrm{L}} = 91$, and $k^{\mathrm{S}} = 211$ W/m·K, respectively). Solid and liquid aluminum and gas have densities of $\rho^{\mathrm{S}} = 2700$, $\rho^{\mathrm{L}} = 2475$, and $\rho^{\mathrm{G}} = 0.4$ kg/m^3, respectively, specific heats of $C^{\mathrm{S}} = 910$, $C^{\mathrm{L}} = 1042.4$, and $C^{\mathrm{G}} = 1100$ J/kg·K, respectively, and viscosities of $\mu_{\mathrm{l}} = \mu_{\mathrm{s}} = 1.4 \times 10^{-3}$, and $\mu_{\mathrm{g}} = 4.5 \times 10^{-5}$ Pa·s, respectively. Viscosity in the solid phase is fictitious and does not affect numerical results. The surface tension coefficient between liquid aluminum and gas is taken to be $\gamma = 0.87$ N/m. The latent heat of fusion/melting of aluminum is $L = 383840$ J/kg.

The time evolution of the non-isothermal phase changing gas-liquid-solid system is governed by the equation of state, and conservation of mass, momentum and energy equations that read as

$$\text{Equation of state (EOS):} \quad \rho = \rho^{\mathrm{G}} + (\rho^{\mathrm{S}} - \rho^{\mathrm{G}})H + (\rho^{\mathrm{L}} - \rho^{\mathrm{S}})H\varphi, \tag{3.1a}$$

$$\text{Indicator advection:} \quad \frac{DH}{Dt} = \frac{dH}{d\phi}\left(\frac{D\phi}{Dt} + \mathbf{u}\cdot\nabla\phi\right) = H'\frac{D\phi}{Dt} = 0, \tag{3.1b}$$

$$\text{Mass/Low-Mach:} \quad \nabla\cdot\mathbf{u} = -\frac{1}{\rho}\frac{D\rho}{Dt} = \frac{(\rho^{\mathrm{S}} - \rho^{\mathrm{L}})}{\rho}H\frac{D\varphi}{Dt}, \tag{3.1c}$$

$$\begin{aligned}\text{Momentum:} \quad & \frac{D(\rho\mathbf{u})}{Dt} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = -\nabla p + \nabla\cdot\left[\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\intercal})\right] \\ & + \rho\mathbf{g} - A_d\mathbf{u},\end{aligned} \tag{3.1d}$$

$$\text{Energy/Enthalpy:} \quad \frac{D(\rho\hbar)}{Dt} + \nabla\cdot(\rho\mathbf{u}\hbar) = \nabla\cdot(k\nabla T) + Q_{\mathrm{src}}. \tag{3.1e}$$

The EOS (Eq. (3.1a)) defines density in terms of: (i) an indicator function H that is defined to be 1 in the solid-liquid PCM region and 0 in the gas region; and (ii) a liquid fraction variable φ that is defined to be 1 in the liquid, 0 in the solid, and between 0 and 1 in the "mushy" zone.

H is transported with the local velocity $\mathbf{u}$ (Eq. (3.1b)), whereas φ is defined to be an explicit function of enthalpy and evolves with it. The indicator function H is defined in terms of a signed distance/level set function ϕ that satisfies the same linear advection equation as H; see Eq. (3.1b). More details on the novel low Mach formulation of the enthalpy method, its numerical implementation, and its validation with an analytical solution to a two-phase Stefan problem involving jumps in density, kinetic energy, and specific heat is provided in the next chapter 4. We note that the enthalpy $\hbar$ and temperature T of the system evolves due to the boundary conditions and/or heat source/sink term Q_{src}; see Eq. (3.1e). Q_{src} is taken to be zero for this case.

The momentum Eq. (3.1d) contains a volume penalization term $A_d\mathbf{u}$ that retards any fluid motion in the solid region. Here, $A_d = C_d \dfrac{{\varphi_{\text{S}}}^2}{(1-\varphi_{\text{S}})^3 + \epsilon}$ is the Carman-Kozeny drag coefficient, $\varphi_{\text{S}} = H(1-\varphi)$ is solid fraction of the grid cell, and ϵ is a tunable parameter that controls the strength of the permeability parameter $\kappa = \epsilon/C_d = \epsilon\Delta t/(\rho^{\text{S}})$ in the solid region; small values of ϵ (or κ) increase the drag force and retard the motion of solid. This particular choice of $C_d = \rho^{\text{S}}/\Delta t$ is based on an inertial scale and will be explained later. Here, Δt is the time step size. κ is usually chosen through numerical experiments where one can start with $\epsilon \sim \mathcal{O}(10)$ or $\mathcal{O}(1)$ and gradually reduce its value until no further changes are discernible in the solution or phase change dynamics. As an example, Fig. 3.1 shows the solidification dynamics of liquid aluminum using $\epsilon = 10, 1, 10^{-2}$ in the permeability parameter κ. Additionally, we also simulated the solidification dynamics with $\epsilon = 10^{-3}$; the solidification dynamics remained qualitatively the same as in the $\epsilon = 10^{-2}$ case. In the case of $\epsilon = 10$ and 1, the solidification dynamics are incorrect—a lower value of drag force is not sufficient to prevent the gas-solid interface from moving upon complete solidification, which occurs around $t = 0.25$ s in the simulation. However, lowering ϵ to 10^{-2} or below gives the correct dynamics, which has the solidified metal remaining stationary for $t > 0.25$ s. In addition, the low Mach Eq. (3.1c) captures the volume shrinkage/pipe defect due to the density difference between solid and liquid metal ($\rho^{\text{S}} > \rho^{\text{L}}$). We also compute the percentage change in aluminum's mass as it solidifies over time for different grid sizes. It is computed as $\mathcal{E}_{\text{m}} = \frac{|m(t)-m_0|}{m_0} \times 100$, in which $m(t) = \int_\Omega H[\rho^{\text{S}}(1-\varphi) + \rho^{\text{L}}\varphi]\, \mathrm{d}\Omega$

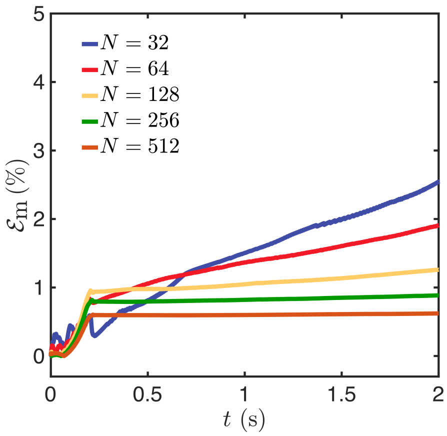


Figure 3.2. Percentage change $\mathcal{E}_\mathrm{m}$ in aluminum's mass as it solidifies over time for different grid sizes. A uniform time step size Δt is used in the simulations. For the coarsest grid $N^2 = 32^2$, a uniform time step size of $\Delta t = 8 \times 10^{-5}$ is employed, and for each successively refined grid Δt is halved.

is aluminum's mass at time t and $m_0 = m(t = 0)$ is its initial mass when it is all liquid. As can be observed from Fig. 3.2, $\mathcal{E}_\mathrm{m}$ decreases significantly under grid refinement. For the finest grid $N^2 = 512^2$, the percentage mass change is approximately 0.62%.

At first, we were unable to go below $\epsilon = 1$ without breaking the monolithic velocity-pressure solver; the linear solver would take a large number of iterations (and long time) to converge, particularly at high mesh resolutions. This was because our initial implementation of the preconditioner employed the projection algorithm suggested by Bergmann and Iolla [11] for solving volume penalized Navier-Stokes equations. Later in this work, we demonstrate that the prior projection algorithm does not consider the volume penalized term in the pressure Poisson equation (PPE), which leads to poor performance of the solver[1]. For incompressible or low Mach systems, one can also use segregated velocity-pressure solvers instead of monolithic ones (where issues related to incorrect/inaccurate PPE would remain "hidden"), but we advocate using monolithic solvers since they have several advantages. This is discussed in Sec. 3.4.3. Monolithic solvers require robust preconditioners, and the proposed preconditioner is an effective strategy.

In what follows, we explain the solution strategy for solving volume penalized equations

---

[1] This foresight came after much struggle and time.

by considering an isothermal (no phase change) and incompressible multiphase system. It avoids the additional complexity associated with heat transfer and phase change. There is no change to the coupled velocity and pressure linear system for low Mach and incompressible Navier-Stokes equations, except for the non-zero right hand side of Eq. (3.1c).

3.3 Equations of motion

3.3.1 The continuous isothermal multiphase equations

Let $\Omega \subset \mathbb{R}^d$ represent a fixed region of space in spatial dimensions $d = 2$ or 3. The volume penalized incompressible Navier-Stokes (INS) equations governing the dynamics of the coupled multiphase fluid-structure system are:

$$\begin{aligned}\frac{\partial \rho \mathbf{u}(\mathbf{x},t)}{\partial t} + \nabla \cdot (\rho\, \mathbf{u}(\mathbf{x},t) \otimes \mathbf{u}(\mathbf{x},t)) = -\nabla p(\mathbf{x},t) + \nabla \cdot\; [\mu\, (\nabla \mathbf{u}(\mathbf{x},t) + \nabla \mathbf{u}(\mathbf{x},t)^{\intercal})] + \mathbf{f} \\ + \frac{\chi(\mathbf{x},t)}{\kappa}\, (\mathbf{u}_b(\mathbf{x},t) - \mathbf{u}(\mathbf{x},t))\,, \qquad (3.2)\end{aligned}$$

$$\nabla \cdot \mathbf{u}(\mathbf{x},t) = 0, \qquad (3.3)$$

which describe the momentum and incompressibility of a fluid with velocity $\mathbf{u}(\mathbf{x},t)$ and pressure $p(\mathbf{x},t)$ in an Eulerian coordinate system $\mathbf{x} \in \mathbb{R}^d$. Eqs. (5.6) and (5.7) are written for the entire computational domain Ω. The domain Ω is further decomposed into two non-overlapping regions, one occupied by the fluid—liquid and gas—$\Omega_f(t) = \Omega_l(t) \cup \Omega_g(t) \subset \Omega$ and the other by an immersed body $\Omega_b(t) \subset \Omega$, so that $\Omega = \Omega_f(t) \cup \Omega_b(t)$. Fig. 3.3 shows the schematic representation of the domain occupied by the three (air, water, solid) phases.

The right-hand side of Eq. (5.6) involves the Brinkman penalty force

$$\boldsymbol{\lambda}(\mathbf{x},t) = \frac{\chi(\mathbf{x},t)}{\kappa}\, (\mathbf{u}_b(\mathbf{x},t) - \mathbf{u}(\mathbf{x},t)) \qquad (3.4)$$

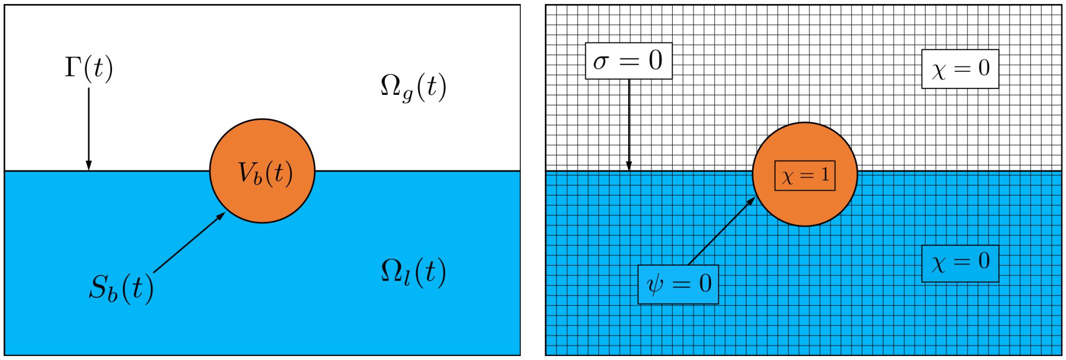
that imposes the structural velocity $\mathbf{u}_b(\mathbf{x},t)$ onto the fictitious fluid contained within $\Omega_b(t)$. In this work, we consider $\mathbf{u}_b$ to be a rigid body velocity. The immersed body is treated as a porous region with vanishing permeability $\kappa \ll 1$, and is tracked using an indicator function $\chi(\mathbf{x}, t)$ that is defined to be one inside $\Omega_b(t)$ and zero outside. In the limit $\kappa \to 0$, the Brinkman penalty coefficient $\chi/\kappa \to \infty$, and $\boldsymbol{\lambda}$ becomes an unknown Lagrange multiplier that needs to be solved for. The Lagrange multiplier formulation is not considered here and we refer the readers to Kallemov et al. [81] and Usabiaga et al. [82] for a solution strategy to this problem. Here, we examine the case of finite, but small values of κ that can make the system of Eqs. (5.6)-(5.7) stiff if treated explicitly or via operator-splitting. The density and viscosity fields vary spatiotemporally and are denoted $\rho(\mathbf{x}, t)$ and $\mu(\mathbf{x}, t)$, respectively. In Eq. (5.6), $\mathbf{f}$ represents an additional body force term, such as gravity. The rigid body velocity $\mathbf{u}_b(\mathbf{x}, t)$ in the solid region $\Omega_b(t)$ can either be prescribed or determined by the combined actions of the hydrodynamic and external forces (e.g., gravity).

When describing multiphase flows, it is useful to introduce additional scalar fields, such as the level set/signed distance function (SDF) whose zero-contour defines the two-phase interface implicitly [83, 84]. To describe three phase solid-liquid-gas flows, two level set functions are required: $\phi(\mathbf{x}, t)$ and $\psi(\mathbf{x}, t)$. The level set function $\phi(\mathbf{x}, t)$ is used to demarcate the liquid (e.g., water) and gas (e.g., air) regions, $\Omega_l \subset \Omega$ and $\Omega_g \subset \Omega$, respectively, in the computational domain. The zero-contour of ϕ defines the two fluid interface $\Gamma(t) = \Omega_l \cap \Omega_{\text{g}}$. Similarly, the surface of the immersed body $S_{\text{b}}(t) = \partial V_b(t)$ is tracked using the zero-contour of the level set function $\psi(\mathbf{x}, t)$; see Fig. 3.3(B). The indicator function $\chi(\mathbf{x}, t)$ for the solid domain is computed based on the level set function ψ. The two SDFs are advected using the local fluid velocity:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{3.5}$$

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0. \tag{3.6}$$

The density and viscosity in the entire computational domain is expressed as a function of $\phi(\mathbf{x}, t)$



(A) Continuous domain (B) Discretized domain

Figure 3.3. (A) Two-dimensional computational domain Ω showing a body immersed in a fluid and interacting with two fluids. (B) Discretization of the domain Ω on a Cartesian mesh and values of the indicator function $\chi(\mathbf{x}, t)$ used to differentiate the fluid and solid regions in the fictitious domain volume penalization method. Here, $\chi(\mathbf{x}, t) = 1$ inside the solid domain and $\chi(\mathbf{x}, t) = 0$ in liquid and gas domains. The liquid-gas interface $\Gamma(t)$ is tracked by the zero-contour of $\phi(\mathbf{x}, t)$, while the zero-contour of $\psi(\mathbf{x}, t)$ tracks the solid-fluid interface $S_{\text{b}}(t)$.

and $\psi(\mathbf{x}, t)$ using the signed distance property:

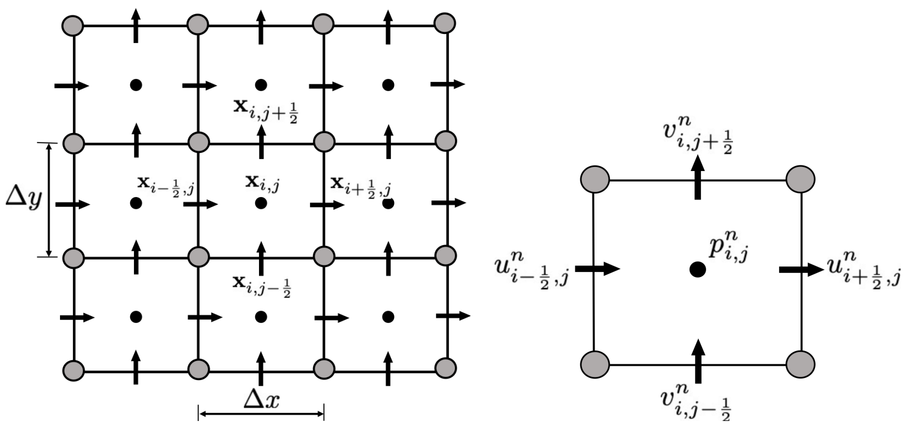
$$\rho(\mathbf{x}, t) = \rho(\phi(\mathbf{x}, t), \psi(\mathbf{x}, t)), \tag{3.7}$$

$$\mu(\mathbf{x}, t) = \mu(\phi(\mathbf{x}, t), \psi(\mathbf{x}, t)). \tag{3.8}$$

To maintain their signed distance property, both level set functions need to be reinitialized after each time step. To reinitialize ϕ, the relaxation approach of Sussman et al. [84] is used to compute the steady state solution to the Hamilton-Jacobi equation. This is explained in Sec. 3.3.4. For simple solid geometries (e.g., cylinder, sphere, wedge) ψ can be reinitialized analytically by using constructive solid geometry operators (i.e., the min/max operator) on primitive shapes [85].

3.3.2 Spatial discretization

The continuous equations of motion given by Eqs. (5.6) and (5.7) are discretized on a staggered Cartesian grid. Without loss of generality, we explain the spatial discretization in $d = 2$ spatial dimensions. Extension to three spatial dimensions is analogous. A discrete $N_x \times N_y$



(A) 2D staggered Cartesian grid (B) Single grid cell

Figure 3.4. Schematic representation of a 2D staggered Cartesian grid. (A) shows the coordinate system for the staggered grid. (B) shows a single grid cell with velocity components u and v approximated at the cell faces ($\rightarrow$) and scalar variable pressure p approximated at the cell center ($\bullet$) at n^{th} time step.

Cartesian grid covers the physical domain Ω with mesh spacing Δx and Δy in each direction. The bottom left corner of the domain is situated at the origin $(0, 0)$. The position of each grid cell center is then given by $\mathbf{x}_{i,j} = \big((i + \frac{1}{2})\Delta x, (j + \frac{1}{2})\Delta y\big)$, where $i = 0, \dots, N_x - 1$ and $j = 0, \dots, N_y - 1$. For a given cell center, $\mathbf{x}_{i-\frac{1}{2},j} = \big(i\Delta x, (j + \frac{1}{2})\Delta y\big)$ where $i = 0, \dots, N_x$ and $j = 0, \dots, N_y - 1$ denotes the physical location of the cell face that is half a grid space away from $\mathbf{x}_{i,j}$ in the negative x-direction. Similarly $\mathbf{x}_{i,j-\frac{1}{2}} = \big((i + \frac{1}{2})\Delta x, j\Delta y\big)$, where $i = 0, \dots, N_x - 1$ and $j = 0, \dots, N_y$ denotes the physical location of the cell face that is half a grid cell away from $\mathbf{x}_{i,j}$ in the negative y-direction. The pressure is defined at cell centers of the staggered grid and are denoted by $p^n_{i,j} = p\left(\mathbf{x}_{i,j}, t^n\right)$, where t^n is the time at time step n. Velocity components are defined at cell faces: $u^n_{i-\frac{1}{2},j} = u\left(\mathbf{x}_{i-\frac{1}{2},j}, t^n\right)$ and $v^n_{i,j-\frac{1}{2}} = v\left(\mathbf{x}_{i,j-\frac{1}{2}}, t^n\right)$. The components of the body force $\mathbf{f} = (f_1, f_2)$ are also defined at x- and y-faces of the staggered grid cells, respectively. The density and viscosity are defined at cell centers of the staggered grid and are denoted by $\rho^n_{i,j} = \rho\left(\mathbf{x}_{i,j}, t^n\right)$ and $\mu^n_{i,j} = \mu\left(\mathbf{x}_{i,j}, t^n\right)$, and are interpolated onto the required degrees of freedom as needed. Similarly, the phase interface is tracked via the level set functions, which are also defined at cell centers and denoted by $\phi^n_{i,j} = \phi\left(\mathbf{x}_{ij}, t^n\right)$ and $\psi^n_{i,j} = \psi\left(\mathbf{x}_{ij}, t^n\right)$.

Fig. 4.2 shows the staggered-grid discretization of Ω.

Standard second-order finite differences are used to approximate the spatial differential operators [77, 78, 86, 87]. These are briefly described here to facilitate the discussion.

The divergence of the velocity field $\mathbf{u} = (u, v)$ is approximated at cell centers by

$$\mathbf{D} \cdot \mathbf{u} = D^x u + D^y v, \tag{3.9}$$

$$(D^x u)_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}, \tag{3.10}$$

$$(D^y v)_{i,j} = \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}. \tag{3.11}$$

The gradient of cell-centered quantities (i.e., p) is approximated at cell faces by

$$\mathbf{G}p = (G^x p, G^y p), \tag{3.12}$$

$$(G^x p)_{i-\frac{1}{2},j} = \frac{p_{i,j} - p_{i-1,j}}{\Delta x}, \tag{3.13}$$

$$(G^y v)_{i,j-\frac{1}{2}} = \frac{p_{i,j} - p_{i,j-1}}{\Delta y}. \tag{3.14}$$

The continuous strain rate tensor form of the viscous term is

$$\nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^\intercal\right)\right] = \left[\begin{array}{c} 2\frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y} + \mu \frac{\partial v}{\partial x}\right) \\ 2\frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x} + \mu \frac{\partial u}{\partial y}\right) \end{array} \right], \tag{3.15}$$

which couples the velocity components through spatially variable viscosity

$$\mathbf{L}_{\boldsymbol{\mu}} \mathbf{u} = \left[\begin{array}{c} (\mathbf{L}_{\boldsymbol{\mu}} \mathbf{u})^x_{i-\frac{1}{2},j} \\ (\mathbf{L}_{\boldsymbol{\mu}} \mathbf{u})^y_{i,j-\frac{1}{2}} \end{array} \right]. \tag{3.16}$$

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

$$\begin{aligned}
(\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u})^x_{i-\frac{1}{2},j} &= \frac{2}{\Delta x}\left[\mu_{i,j}\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x} - \mu_{i-1,j}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{3}{2},j}}{\Delta x}\right] \\
&+ \frac{1}{\Delta y}\left[\mu_{i-\frac{1}{2},j+\frac{1}{2}}\frac{u_{i-\frac{1}{2},j+1}-u_{i-\frac{1}{2},j}}{\Delta y} - \mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{1}{2},j-1}}{\Delta y}\right] \\
&+ \frac{1}{\Delta y}\left[\mu_{i-\frac{1}{2},j+\frac{1}{2}}\frac{v_{i,j+\frac{1}{2}}-v_{i-1,j+\frac{1}{2}}}{\Delta x} - \mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{v_{i,j-\frac{1}{2}}-v_{i-1,j-\frac{1}{2}}}{\Delta x}\right]
\end{aligned} \tag{3.17}$$

$$\begin{aligned}
(\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u})^y_{i,j-\frac{1}{2}} &= \frac{2}{\Delta y}\left[\mu_{i,j}\frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta y} - \mu_{i,j-1}\frac{v_{i,j-\frac{1}{2}}-v_{i,j-\frac{3}{2}}}{\Delta y}\right] \\
&+ \frac{1}{\Delta x}\left[\mu_{i+\frac{1}{2},j-\frac{1}{2}}\frac{v_{i+1,j-\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta x} - \mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{v_{i,j-\frac{1}{2}}-v_{i-1,j-\frac{1}{2}}}{\Delta x}\right] \\
&+ \frac{1}{\Delta x}\left[\mu_{i+\frac{1}{2},j-\frac{1}{2}}\frac{u_{i+\frac{1}{2},j}-u_{i+\frac{1}{2},j-1}}{\Delta y} - \mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{1}{2},j-1}}{\Delta y}\right],
\end{aligned} \tag{3.18}$$

in which viscosity is required on both cell centers and *nodes* of the staggered grid (i.e., $\mu_{i\pm\frac{1}{2},j\pm\frac{1}{2}}$). Node centered quantities are obtained via interpolation by either arithmetically or harmonically averaging the neighboring cell centered quantities. In three dimensions the viscosity is required on both cell centers and *edges* of the staggered grid. Arithmetic averaging is more accurate (second-order accurate interpolation), whereas harmonic averaging provides a better convergence rate at the expense of solution accuracy for solvers dealing with large contrasting material properties, such as density, viscosity, and thermal conductivity. A discussion of this topic can be found in the classic CFD textbook of Patankar [88]. We have also observed this in our previous work related to two phase flows [59]. In this work we use arithmetic averaging.

The linear operators described above are needed to fully discretize the continuous equations of motion. An additional approximation to a variable-coefficient Laplacian is required for the projection preconditioner described in Sec. 3.4.3

$$\begin{aligned}
(\mathbf{L}_{\boldsymbol{\rho_\chi}}\, p)_{i,j} &= \frac{1}{\Delta x}\left[\frac{1}{\rho_{\chi\ i+\frac{1}{2},j}}\frac{p_{i+1,j}-p_{i,j}}{\Delta x} - \frac{1}{\rho_{\chi\ i-\frac{1}{2},j}}\frac{p_{i,j}-p_{i-1,j}}{\Delta x}\right] \\
&+ \frac{1}{\Delta y}\left[\frac{1}{\rho_{\chi\ i,j+\frac{1}{2}}}\frac{p_{i,j+1}-p_{i,j}}{\Delta y} - \frac{1}{\rho_{\chi\ i,j-\frac{1}{2}}}\frac{p_{i,j}-p_{i,j-1}}{\Delta y}\right],
\end{aligned} \tag{3.19}$$

which requires sum of density and Brinkman penalty coefficient, denoted ρ_χ, on faces of the staggered grid ($\rho_{\chi\ i\pm\frac{1}{2},j}$ and $\rho_{\chi\ i,j\pm\frac{1}{2}}$ in Eq. (3.19)). These can also be computed using either the arithmetic or harmonic averages of density from the two adjacent cell centers. It is important to note that our formulation utilizes the interpolated face-centered density within the preconditioner, which does not affect solution accuracy or stability as long as the linear system of equations converges. This means that any suitable interpolation scheme for density works equally well for the preconditioner.

Evaluation of finite difference operators near boundaries of the computational domain requires specification of abutting "ghost" values. We also use an adaptive mesh refinement (AMR) framework to reduce computational costs of 3D simulations. For further details on the spatial discretization and boundary conditions on uniform and spatially adaptive grids, see our prior work [59].

3.3.3 Density and viscosity specification

Smoothed Heaviside functions are used to transition between liquid-gas and fluid-solid interfaces $\Gamma(t)$ and $S_b(t)$, respectively[2] The material properties in the transition area are smoothly varied by using n_{smear} grid cells on either side of the interface. As an example, to calculate a given material property $\Im$, such as density or viscosity, the *flowing* phase property (i.e., gas and liquid) is calculated first

$$\Im_{i,j}^{\text{flow}} = \Im^{\text{L}} + (\Im^{\text{G}} - \Im^{\text{L}})\widetilde{H}_{i,j}^{\text{flow}}, \tag{3.20}$$

and later correcting $\Im^{\text{flow}}$ to account for the solid body by

$$\Im_{i,j}^{\text{full}} = \Im^{\text{S}} + (\Im_{i,j}^{\text{flow}} - \Im^{\text{S}})\widetilde{H}_{i,j}^{\text{body}}. \tag{3.21}$$

---

[2] Due to the scale of problems we are interested in, such as simulating wave energy converters and objects slamming into air-water interfaces, we do not impose any contact angle condition at the three material points, since this does not affect the rigid dynamics and the hydrodynamic forces acting on them. Furthermore, this detail also does not affect the spectrum of linear operators, which is the main area of concern for the preconditioner.

Here, $\Im^{\text{full}}$ is the final scalar material property field throughout Ω. For the transition specified by Eqs. 3.20 and 3.21, the usual numerical Heaviside functions are used:

$$\widetilde{H}_{i,j}^{\text{flow}} = \begin{cases} 0, & \phi_{i,j} < -n_{\text{smear}}\, h, \\ \frac{1}{2}\left(1 + \frac{1}{n_{\text{smear}}\, h}\phi_{i,j} + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{smear}}\, h}\phi_{i,j}\right)\right), & |\phi_{i,j}| \le n_{\text{smear}}\, h, \\ 1, & \text{otherwise.} \end{cases} \tag{3.22}$$

$$\widetilde{H}_{i,j}^{\text{body}} = \begin{cases} 0, & \psi_{i,j} < -n_{\text{smear}}\, h, \\ \frac{1}{2}\left(1 + \frac{1}{n_{\text{smear}}\, h}\psi_{i,j} + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{smear}}\, h}\psi_{i,j}\right)\right), & |\psi_{i,j}| \le n_{\text{smear}}\, h, \\ 1, & \text{otherwise.} \end{cases} \tag{3.23}$$

in which h is a suitable measure of the cell size (e.g., Δx). By convention, we define ϕ and ψ to be negative (positive) in the liquid (gas) and solid (fluid) regions. In all simulations performed in this study, the number of transition cells $n_{\text{smear}} = 1$ for both air-water and fluid-solid interfaces, unless mentioned otherwise.

The VP method is a diffuse interface method in which all quantities (pressure, velocity, etc.) vary smoothly over the transition region (which is $2n_{\text{smear}}$ wide in our model), and therefore, do not jump across the interface. In addition, the transition region remains incompressible, since the same kinematic constraint is imposed on the velocity (Eq. (5.7)) in this region as well. Diffuse interface methods differ from their sharp interface counterparts, such as the immersed interface method [89, 90] and the ghost fluid method [91], which assume the interface thickness is zero and explicitly impose jump conditions for the quantities of interest.

3.3.4 Temporal discretization

A fixed-point iteration scheme with n_{cycles} cycles per time step is used to evolve quantities from time level t^n to time level $t^{n+1} = t^n + \Delta t$. The cycle number of the fixed-point iteration scheme is denoted with a k superscript. At the beginning of each time step, the solutions from the previous time step are used to initialize cycle $k = 0$: $\mathbf{u}^{n+1,0} = \mathbf{u}^n$, $p^{n+\frac{1}{2},0} = p^{n-\frac{1}{2}}$, $\phi^{n+1,0} = \phi^n$, and $\psi^{n+1,0} = \psi^n$. The physical quantities at the initial time $n = 0$ are prescribed via initial conditions. Unless mentioned otherwise, we use $n_{\text{cycles}} = 2$ in all of the test cases.

Level set advection

The two level set/signed distance functions ϕ and ψ are advected using an explicit advection scheme as follows

$$\frac{\phi^{n+1,k+1} - \phi^n}{\Delta t} + Q\left(\mathbf{u}^{n+\frac{1}{2},k}, \phi^{n+\frac{1}{2},k}\right) = 0, \tag{3.24}$$

$$\frac{\psi^{n+1,k+1} - \psi^n}{\Delta t} + Q\left(\mathbf{u}^{n+\frac{1}{2},k}, \psi^{n+\frac{1}{2},k}\right) = 0, \tag{3.25}$$

in which $Q(\cdot,\cdot)$ represents an explicit piecewise parabolic method (xsPPM7-limited) approximation to the linear advection terms on cell centers [77, 92].

Mitigating mass/volume loss with the level set method

With the geometries considered in this study, we are able to reset the solid level set function $\psi(\mathbf{x}, t)$ analytically. The analytical resonstruction preserves the mass/volume of the body while not distorting ψ's signed distance property following the linear advection Eq. (3.25). In contrast, ϕ cannot be reinitialized analytically after its signed distance property is disrupted by the linear advection Eq. (3.24). To restore its signed distance property, a reinitialization strategy is required. Let $\widetilde{\phi}^{n+1}$ denote the level set function following an advection procedure after time stepping through the interval $[t^n, t^{n+1}]$. We aim to reinitialize it to obtain a signed distance function ϕ^{n+1}. As proposed by Sussman et al. [84], this can be achieved by computing a steady-state solution to the Hamilton-Jacobi equation

$$\frac{\partial \phi}{\partial \tau} + \text{sgn}\left(\widetilde{\phi}\right)\left(\|\nabla \phi\| - 1\right) = 0, \tag{3.26}$$

$$\phi(\mathbf{x}, \tau = 0) = \widetilde{\phi}(\mathbf{x}), \tag{3.27}$$

in which we have dropped the $n+1$ superscript because this process is agnostic to the particular time step under consideration. At the end of a physical time step, Eq. (3.26) is evolved in pseudo-time τ, which, at steady state, produces a signed distance function satisfying the Eikonal equation $\|\nabla\phi\| = 1$. Here, sgn denotes the sign of $\widetilde{\phi}$, which is either 1, -1, or 0. The discretization of Eq. (3.26) from the pseudo-time interval $[\tau^m, \tau^{m+1}]$ yields

$$\frac{\phi^{m+1} - \phi^m}{\Delta\tau} + \text{sgn}\left(\widetilde{\phi}_{i,j}\right)\left[H_\text{G}\left(D_x^+\phi_{i,j}, D_x^-\phi_{i,j}, D_y^+\phi_{i,j}, D_y^-\phi_{i,j}\right) - 1\right] = 0, \tag{3.28}$$

in which H_G denotes a discretization of $\|\nabla\phi\|$ using the Godunov-Hamiltonian, and $D_x^\pm$ and $D_y^\pm$ denote one-sided discretizations of $\frac{\partial\phi}{\partial x}$ and $\frac{\partial\phi}{\partial y}$, respectively. These are typically discretized using high-order essentially non-oscillatory (ENO) or weighted ENO (WENO) schemes [93].

It is well known that continually applying Eq. (3.28) will cause the interface to shift as a function of τ [94], which will eventually shrink closed interfaces and lead to substantial spurious changes in the volume of each phase. To mitigate the spurious volume loss associated with Eq. (3.26), we employ second-order ENO finite differences combined with a subcell-fix method described by Min [95]. Briefly, the subcell-fix method uses $\widetilde{\phi}$ to estimate the interface location (i.e., where $\widetilde{\phi} = 0$) by fitting a high-order polynomial and computing an improved estimate of the one-sided derivatives $D_x^\pm$ and $D_y^\pm$ from the polynomial fit. A dimension-by-dimension approach is followed to fit the high-order polynomial. After iterating Eq. 3.28 (using an appropriate time-stepping scheme, e.g., TVD RK2) to some desired convergence criteria, the level set function ϕ^{n+1} is updated, and the next physical time step is carried out. In the present work, we always reinitialize the level set every time step and declare convergence when the L^2 norm between subsequent pseudo-time iterations is smaller than some tolerance (taken to be 10^{-6} in the present work) or when a maximum number of iterations (taken to be 50) have been carried out — whichever happens first.

All simulations that involve liquid-gas interfaces report percentage mass/volume changes.

Additionally, a two-phase dam break problem and two- and three-phase Rayleigh-Taylor instability problems are simulated in Appendix Secs. 6.4 and 6.5, respectively . We compare the percentage volume change of conserved phases with prior numerical studies that also employ the standard level set methodology. Our level set method implementation achieves acceptable mass/volume loss, as shown in the results. We note that mitigating mass/volume loss issues with the level set method is an active area of research; see for example, Howard and Tartakovsky [96] who recently proposed a conservative level set method for N-phase flows that preserves the volume of every phase simultaneously. Our current work focuses primarily on solving the stiff system of equations that result from discretizing volume penalized multiphase flow equations. The proposed preconditioner is agnostic to level set implementation details and applies equally to conservative level set and volume of fluid methods.

The discrete multiphase equations

The discretized form of the multiphase incompressible Navier-Stokes Eqs. (5.6) and (5.7) in conservative form reads as

$$
\begin{aligned}
\frac{\breve{\boldsymbol{\rho}}^{n+1,k+1}\mathbf{u}^{n+1,k+1} - \boldsymbol{\rho}^{n}\mathbf{u}^{n}}{\Delta t} + \mathbf{C}^{n+1,k} &= -\nabla_h\, p^{n+\frac{1}{2},k+1} + (\mathbf{L}_{\mu}\mathbf{u})^{n+\frac{1}{2},k+1} + \mathbf{f}^{n+\frac{1}{2},k+1} \\
&\quad + \frac{\widetilde{\boldsymbol{\chi}}^{n+1,k+1}}{\kappa}\left(\mathbf{u}_b^{n+1,k+1} - \mathbf{u}^{n+1,k+1}\right), \qquad (3.29) \\
\nabla \cdot \mathbf{u}^{n+1,k+1} &= \mathbf{0}, \qquad (3.30)
\end{aligned}
$$

in which $\mathbf{C}^{n+1,k} = \mathbf{C}(\rho^{n+1,k}, \mathbf{u}^{n+1,k})$ is the discretized version of the convective term $\nabla\cdot(\rho\, \mathbf{u}\otimes\mathbf{u})$ and the density approximation $\breve{\boldsymbol{\rho}}^{n+1,k+1}$ is computed by integrating the (auxiliary) mass balance equation to achieve mass and momentum transport consistency at the discrete level. The consistent mass/momentum transport scheme ensures the numerical stability of cases involving high density contrast between solid, liquid and gas phases. This is discussed in greater detail in our previous work [59].

The viscous strain rate in Eq. (3.29) is handled using the Crank-Nicolson (trapezoidal rule) approximation: $(\mathbf{L}_\mu \mathbf{u})^{n+\frac{1}{2},k+1} = \frac{1}{2}\left[(\mathbf{L}_\mu \mathbf{u})^{n+1,k+1} + (\mathbf{L}_\mu \mathbf{u})^{n}\right]$, in which $(\mathbf{L}_\mu \mathbf{u})^{n+1} = \nabla_h \cdot \left[\mu^{n+1}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^\intercal\right)^{n+1}\right]$. The newest approximation to the viscosity $\mu^{n+1,k+1}$ is obtained using the two-stage process as described in Sec. 3.3.3.

Fluid-structure coupling

Next, we describe the Brinkman penalization term that imposes the rigidity constraint in the solid region, and the overall fluid-structure coupling scheme. The Brinkman penalization term is treated implicitly and computed as

$$\boldsymbol{\lambda}^{n+1,k+1} = \frac{\widetilde{\boldsymbol{\chi}}}{\kappa}\left(\mathbf{u}_b^{n+1,k+1} - \mathbf{u}^{n+1,k+1}\right), \tag{3.31}$$

in which the discretized indicator function $\widetilde{\boldsymbol{\chi}} = \mathbf{1} - \widetilde{\mathbf{H}}^{\text{body}}$ is 1 only inside the body domain and defined using the structure Heaviside function $\widetilde{H}^{\text{body}}$ from Eq. (3.23). With $\mathbf{X}_{\text{com}}$ denoting the position of the center of mass of the body, the rigid body velocity $\mathbf{u}_b = \mathbf{U}_r + \mathbf{W}_r \times (\mathbf{x} - \mathbf{X}_{\text{com}})$ can be expressed as a sum of translational $\mathbf{U}_r$ and rotational $\mathbf{W}_r$ velocities. The rigid body velocities can be obtained by integrating Newton's second law of motion

$$M_b \frac{\mathbf{U}_r^{n+1,k+1} - \mathbf{U}_r^{n}}{\Delta t} = \boldsymbol{\mathcal{F}}^{n+1,k} + M_b \mathbf{g}, \tag{3.32}$$

$$\frac{(\mathbf{I}_b \mathbf{W}_r)^{n+1,k+1} - (\mathbf{I}_b \mathbf{W}_r)^{n}}{\Delta t} = \mathcal{M}^{n+1,k}, \tag{3.33}$$

in which M_b is the mass, $\mathbf{I}_b$ is the moment of inertia, $\boldsymbol{\mathcal{F}}$ is the net hydrodynamic force, $\mathcal{M}$ is the net hydrodynamic torque and $M_b\mathbf{g}$ is the net gravitational force acting on the body. Eqs. (3.32) and (3.33) are integrated using an explicit forward Euler scheme to compute $\mathbf{U}_r^{n+1,k+1}$, $\mathbf{W}_r^{n+1,k+1}$ and $\mathbf{X}_{\text{com}}^{n+1,k+1}$. In practice we employ quaternions to integrate Eq. (3.33) in the initial reference frame, which avoids recomputing $\mathbf{I}_b$ as the body rotates in a complex manner in three spatial dimensions.

The multiphase FSI simulations presented in this work consider immersed bodies with only one unlocked translational degree of freedom. In a previous work [41], simultaneous free translation and rotational motions of the body have been considered with the VP approach.

3.4 Solution methodology

3.4.1 Fully-coupled Brinkman penalized Stokes system

Solving for $\mathbf{u}^{n+1,k+1}$ and $p^{n+\frac{1}{2},k+1}$ in Eqs. (3.29) and (3.30) requires the solution of the following block linear system

$$\begin{bmatrix} \frac{1}{\Delta t}\breve{\boldsymbol{\rho}}^{n+1,k+1} + \frac{1}{\kappa}\widetilde{\boldsymbol{\chi}}^{n+1,k+1} - \frac{1}{2}\mathbf{L}_{\boldsymbol{\mu}}{}^{n+1,k+1} & \mathbf{G} \\ -\mathbf{D}\cdot & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1,k+1} \\ p^{n+1,k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{u}} \\ \mathbf{0} \end{bmatrix}, \tag{3.34}$$

in which $\breve{\boldsymbol{\rho}}^{n+1,k+1}$ and $\widetilde{\boldsymbol{\chi}}^{n+1,k+1} = \mathbf{1} - \widetilde{\mathbf{H}}^{\text{body}}$ are diagonal matrices of face-centered densities and body characteristic function corresponding to each velocity degree of freedom, respectively. The right-hand side of the momentum equation is lumped into $\mathbf{f}_{\mathbf{u}}$, which reads as

$$\mathbf{f}_{\mathbf{u}} = \left(\frac{1}{\Delta t}\boldsymbol{\rho}^{n} + \frac{1}{2}\mathbf{L}_{\boldsymbol{\mu}}{}^{n}\right)\mathbf{u}^{n} + \left(\frac{\widetilde{\boldsymbol{\chi}}}{\kappa}\right)^{n+1,k+1}\mathbf{u}_b^{n+1,k+1} - \mathbf{C}^{n+1,k} + \mathbf{f}^{n+\frac{1}{2}}. \tag{3.35}$$

The operator on the left-hand side of Eq. (3.34) is the time-dependent, incompressible staggered Stokes operator with an additional Brinkman penalty term in the (1,1) block. We call this the Brinkman penalized Stokes operator or *Stokes-BP* operator for short. In the next section, we describe the solution of Eq. (3.34) via the GMRES or flexible GMRES (FGMRES) Krylov solver [97] that is preconditioned with an inexact projection solver.

3.4.2 Projection solver for the Brinkman penalized Stokes system

The most popular approach to solving the incompressible Stokes system is the fractional-step projection method. Bergmann and Iolla [11] solved the Stokes-BP system (Eq. (3.34)) using the projection solver by considering spatially uniform density and viscosity in the domain. Recently, Sharaborin et al. [15] solved the Stokes-BP system for variable density and viscosity flows. However, in both works [11, 15] the pressure Poisson equation (PPE) did not include the Brinkman penalty term $\widetilde{\boldsymbol{\chi}}/\kappa$. Our tests suggest that including the penalty term in the projection algorithm ensures robust convergence of the monolithic fluid solver, particularly when κ values are small. To see how the penalty term appears in PPE, the algorithmic derivation of projection method for variable density and viscosity Stokes-BP system is presented next. The special case of spatially uniform density (and viscosity) is also discussed.

The Stokes-BP system of Eq. (3.34) can be succinctly written in the following form

$$\begin{gathered}\mathbf{M}\,\mathbf{x} = \mathbf{b} \\ \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ -\mathbf{D}\cdot & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x_u} \\ \mathbf{x_p} \end{bmatrix} = \begin{bmatrix} \mathbf{b_u} \\ \mathbf{b_p} \end{bmatrix}\end{gathered} \tag{3.36}$$

in which $\mathbf{M}$ denotes the Stokes-BP operator, $\mathbf{A} = \frac{1}{\Delta t}\breve{\boldsymbol{\rho}}^{n+1,k+1} + \frac{1}{\kappa}\widetilde{\boldsymbol{\chi}}^{n+1,k+1} - \frac{1}{2}\mathbf{L}_{\boldsymbol{\mu}}{}^{n+1,k+1}$ denotes the discretization of the temporal, Brinkman penalty and viscous operator, $\mathbf{x_u}$ and $\mathbf{x_p}$ denote the velocity and pressure degrees of freedom, and $\mathbf{b_u}$ and $\mathbf{b_p} = \mathbf{0}$ denote the velocity and pressure right-hand sides.

Formally, the projection method can be derived by approximating the inverse of the Schur complement of the saddle-point system Eq. (3.36). This is shown in the Appendix Sec. 6.3. Algorithmically, in the first step of the projection method, an intermediate approximation to $\mathbf{u}$ is computed by solving

$$\mathbf{A}\widehat{\mathbf{x}}_\mathbf{u} = \mathbf{b_u}. \tag{3.37}$$

Note that this approximation does not in general satisfy the discrete continuity equation i.e., $-\mathbf{D}\cdot\widehat{\mathbf{x}}_{\mathbf{u}} \neq \mathbf{b}_{\mathbf{p}}$. This condition can be satisfied by introducing an auxiliary scalar field $\boldsymbol{\phi}$ and writing out a fractional timestep

$$\left(\frac{\breve{\boldsymbol{\rho}}}{\Delta t} + \frac{\widetilde{\boldsymbol{\chi}}}{\kappa}\right)(\mathbf{x}_{\mathbf{u}} - \widehat{\mathbf{x}}_{\mathbf{u}}) = -\mathbf{G}\boldsymbol{\phi}, \tag{3.38}$$

$$-\mathbf{D}\cdot\mathbf{x}_{\mathbf{u}} = \mathbf{b}_{\mathbf{p}}. \tag{3.39}$$

Multiplying Eq. (3.38) by

$$\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1} = \left(\breve{\boldsymbol{\rho}} + \frac{\widetilde{\boldsymbol{\chi}}\Delta t}{\kappa}\right)^{-1}, \tag{3.40}$$

taking the discrete divergence $\mathbf{D}\cdot$, and substituting in Eq. 3.38 yields the *density and Brinkman penalty*-weighted Poisson problem

$$-(\mathbf{D}\cdot\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G})\boldsymbol{\phi} = -\mathbf{L}_{\rho_\chi}\,\boldsymbol{\phi} = -\frac{1}{\Delta t}\left(\mathbf{b}_{\mathbf{p}} + \mathbf{D}\cdot\widehat{\mathbf{x}}_{\mathbf{u}}\right). \tag{3.41}$$

The updated velocity solution can be computed as

$$\mathbf{x}_{\mathbf{u}} = \widehat{\mathbf{x}}_{\mathbf{u}} - \Delta t\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G}\boldsymbol{\phi}, \tag{3.42}$$

and that of pressure can be computed as

$$\mathbf{x}_{\mathbf{p}} = \boldsymbol{\phi}. \tag{3.43}$$

The main difference between our projection method and that of Bergmann and Iolla and Sharaborin et al. is that we include the stiff Brinkman penalty term $\widetilde{\boldsymbol{\chi}}/\kappa$ in the pressure Poisson Eq. (3.41) and the velocity update Eq. (3.42). This is a small but a crucial detail that leads to robust convergence of the monolithic velocity-pressure solver, particularly when κ is small. In Sec. 3.6, we careful study the effect of the Brinkman penalty term on the solver convergence rate. Note that in Bergmann and Iolla [11] a variable coefficient Poisson solver was not used to solve the PPE. This is because for constant density and without the Brinkman penalty $\boldsymbol{\rho}_{\chi}^{-1} = \frac{1}{\rho}\mathbf{I}$ trivializes to a scalar multiple of the identity matrix $\mathbf{I}$.

3.4.3 Projection preconditioner for the Brinkman penalized Stokes system

Although the Stokes-BP system of Eq. (3.36) can be solved using the projection method, we do not follow this approach here. Instead we use the projection method as a preconditioner to solve the coupled velocity-pressure system. There are several advantages to using the projection method as a preconditioner rather than as a solver. These are discussed in more detail in Griffith [77] and Cai et al. [78]. Below is a summary of the main ones.

- The projection method solves the Stokes system by splitting the velocity and pressure degrees of freedom. The operator-splitting approach requires specifying artificial boundary conditions for the velocity (Eq. (3.37)) and pressure (Eq. (3.41)) fields. This split affects the global order of accuracy of the solution [98]. For example, it is not possible to impose normal traction boundary condition in the projection solver because this requires a linear combination of discretized pressure and velocity variables. This combination can be accounted for in the (Brinkman penalized) Stokes operator $\mathbb{M}$ directly, and the projection preconditioner can still use the artificial boundary conditions in the split velocity and pressure solves. A preconditioner based on artificial boundary conditions does not affect the final outcome of the discretized system it only affects the solver's convergence rate. In Sec. 3.6.1 we consider a test problem with spatially-varying normal traction boundary conditions. The projection preconditioner uses homogeneous Dirichlet and Neumann boundary conditions for the pressure and normal component of velocity, respectively when the normal traction boundary condition is imposed.

- The projection solver is derived under the assumption that certain operators commute; see Appendix Sec. 6.3 for operator commutations associated with the projection method.

These assumptions are typically only satisfied by constant-coefficient operators defined on periodic domains. There is an unavoidable commutator error associated with using the projection method as a solver for variable-coefficient operators.

- Furthermore, using the projection method as a preconditioner is no less efficient than using it as a solver, as Griffith [77], Cai et al. [78], and Nangia et al. [59] demonstrate.

It is relatively straightforward to use the projection solver discussed in Sec. 3.4.2 as a projection preconditioner. In this approach, an outer Krylov solver (e.g., GMRES or FGMRES) is employed that generates a Krylov subspace by applying the action of matrix $\mathbf{M}$ on vectors. When the Krylov solver is preconditioned with the projection solver, it also requires the action of the preconditioner on residual vectors to get estimates on velocity and pressure errors. For the projection preconditioner, the unknowns $\mathbf{x_u}$ and $\mathbf{x_p}$ defined in Eq. (3.36) should be interpreted as errors in velocity and pressure degrees of freedom, respectively, and the right-hand side vectors $\mathbf{b_u}$ and $\mathbf{b_p} \neq \mathbf{0}$ as residuals of momentum and continuity constraint equations, respectively. The projection preconditioner computes the error in velocity ($\mathbf{x_u}$) and pressure ($\mathbf{x_p}$) only approximately. This is achieved by solving Eqs. (3.37) and (3.41) in an inexact manner. Specifically, we solve the velocity and pressure subdomain problems using a Richardson solver that is preconditioned with a single V-cycle of a geometric multigrid solver [99]. For both velocity and pressure problems, 3 iterations of Gauss-Seidel smoothing are performed on each multigrid level.

For the first-order accurate projection method, the pressure solution can be approximated as $\mathbf{x_p} = \boldsymbol{\phi}$ as written in Eq. (3.43); see Brown et al. [98]. In the presence of a spatially-varying viscosity, a more accurate approximation to the pressure solution can be obtained as

$$\mathbf{x_p} = \left(\mathbf{I} - \Delta t\, \boldsymbol{\mu}\, \mathbf{L}_{\rho_{\mathbf{x}}}\right) \boldsymbol{\phi}, \tag{3.44}$$

in which $\boldsymbol{\mu}$ is a diagonal matrix of cell-centered viscosities corresponding to each pressure

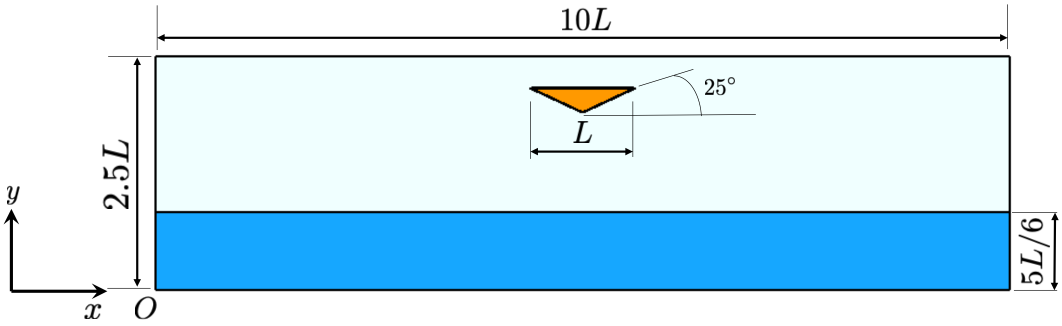


Figure 3.5. Schematic of the 2D wedge slamming on an air-water interface.

degree of freedom. The above form of $\mathbf{x}_\mathrm{p}$ is derived by approximating the inverse of the Schur complement of the saddle-point system Eq. (3.36) as done in Cai et al. [78]. Appendix Sec. 6.3 provides the derivation. In the projection preconditioner we update pressure using Eq. (3.44) instead of Eq. (3.43). Note that in Cai et al. $\mathbf{L}_\rho$ is the density-weighted Laplace operator, whereas in this work $\mathbf{L}_{\rho_\chi}$ is the density and Brinkman penalty-weighted Laplace operator (Eq. (3.41)).

3.5 Validation of the multiphase VP method: Free falling wedge slamming into an air-water interface

As a validation case, we simulate the free fall of a two-dimensional and a three-dimensional wedge slamming against an air-water interface. The top length of the 2D wedge is L = 1.2 m, which is positioned within a computational domain of extents $\Omega = [0, 10L] \times [0, 5L/2]$. The 3D wedge simulation is performed in a computational domain of extents $\Omega = [0, 10L] \times [0, 5L/3] \times [0, 5L/2]$. The top surface of the 3D wedge is a square of extents $L \times L$. The origin O of the computational domains are at the bottom left corner; see Fig. 3.5. The initial coordinates of the lower-most vertex of the 2D wedge is (x_0, y_0) = $(5L, 23L/12)$ and that of the 3D wedge is (x_0, y_0, z_0) = $(5L, 5L/6, 23L/12)$. The wedge makes an angle of 25° with the horizontal and its free fall height is $\Delta s = 13L/12$. There is a distance of $5L/12$ between the wedge vertical sides and the lateral wall of the 3D computational domain. The depth of the water column is $5L/6$ and the remainder of the domain is occupied by air. We assume the density of water is ρ^L = 1000 kg/m^3 and the viscosity is μ_l = 10^{-3} Pa·s. For air, the density is assumed to be ρ^G = 1.2 kg/m^3

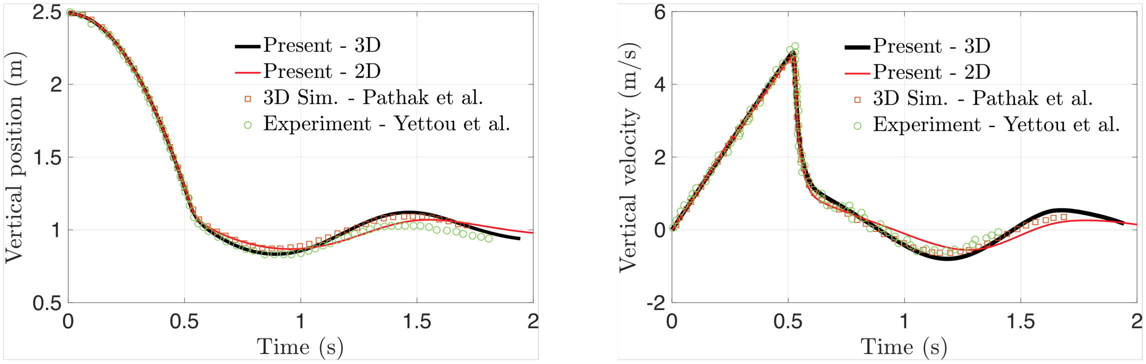
and viscosity to be μ_{g} = $1.8{\times}10^{-5}$ Pa·s. Wedge density is assumed to be ρ^{S} = 466.6 kg/m^3 and its fictitious viscosity μ_{s} is the same as that of water.

The computational domain for the 2D case is discretized into a uniform grid of size 1200×300, which corresponds to 120 grid cells per wedge length. The uniform mesh spacing is $\Delta x = \Delta y = 0.01$ m. A similar grid size in 3D would be computationally very expensive; therefore, we employ an adaptive mesh refinement framework to keep the mesh resolution high in only a few select regions. Among these are regions containing immersed bodies, air-water interfaces, and vorticity of large magnitude. The 3D domain is discretized using $\ell = 3$ grid levels with a refinement ratio $n_{\text{ref}} = 2$. The mesh spacing on the coarsest level is $\Delta x_0 = \Delta y_0 = \Delta z_0 = 0.04$ m and on the finest level is $\Delta x_{\text{min}} = \Delta y_{\text{min}} = \Delta z_{\text{min}} = 0.01$ m. A constant time step size of $\Delta t = 6.25 \times 10^{-5}$ s is used in both cases. As determined by a grid resolution study (see Fig. 3.12), the mesh and time step size used are adequate to resolve the FSI dynamics of the freely falling wedge.

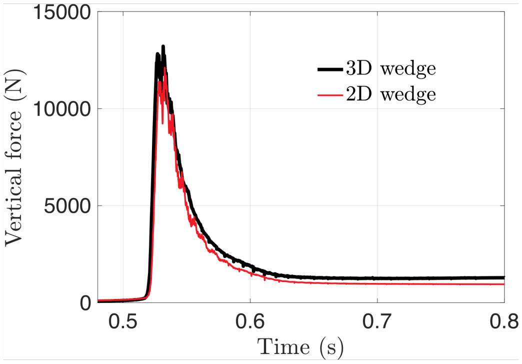
Figs. 3.6(A) and 3.6(B) illustrate the temporal evolution of the wedge's vertical position and velocity, respectively. The results are in good agreement with previous 3D volume of fluid simulations of Pathak et al. [2] [3] and experimental study of Yettou et al. [3]. There is some mismatch between the 2D simulations and experimentally measured dynamics during later times. In contrast, the present 3D simulation agrees better with the 3D simulations of Pathak et al. [2] and the experimental results of Yettou et al. [3]. Fig. 3.6(B) illustrates how the vertical velocity of the wedge decreases as the wedge penetrates the water and how it reverses directions due to buoyancy forces. Fig. 3.6(C) compares the vertical hydrodynamic force (viscous and pressure forces) on the wedge surface over time for both 2D and 3D cases. Some differences are observed. A peak load around the time of slamming ($t \approx 0.52$ s) can be observed in the figure. Fig. 3.7 shows the evolution of wake behind the falling 2D wedge and the vortex shedding upon slamming. This type of wake has also been observed in experiments with falling cones [100]. In Fig. 3.8 we show pressure in the domain at various times. As the 2D wedge impacts, a high

---

[3] Pathak et al. also impose a contact angle condition at the three material points in their volume of fluid simulation.



(A) Vertical postion (B) Vertical velocity

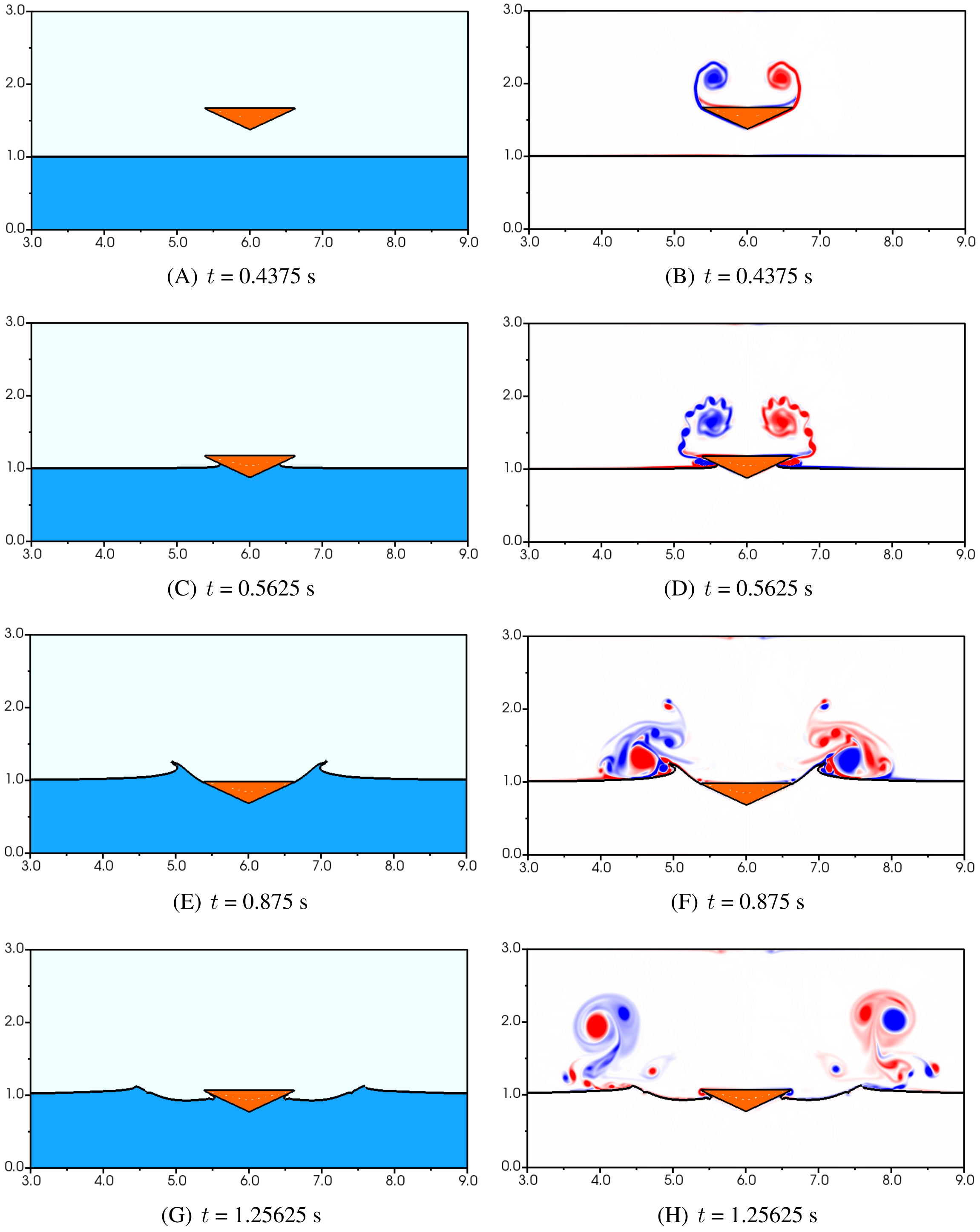


(C) Vertical force

Figure 3.6. Temporal evolution of a free falling wedge slamming against an air-water interface: (A) vertical position; (B) vertical velocity; and (C) vertical component of the hydrodynamic force for 2D and 3D wedge case. The simulation results are compared against previous 3D volume of fluid simulations of Pathak et al. [2] and experimental study of Yettou et al. [3].

pressure region forms at its bottom tip. When the wedge penetrates further into the water, the high pressure region shifts to its inclined surface. After a period of time, the pressure in the domain becomes hydrostatic (increases linearly with depth).

Fig. 3.9 shows the fluid-structure interaction of the 3D wedge case simulated using AMR at three distinct time instances: (left column) density plot; (center column) mesh levels; (right column) shed vortex structures shown on a 2D slice of the domain taken at $y = 5L/6$. Initially, the finest mesh level captures only the air-water interface and the region around the 3D wedge. More fine mesh regions are dynamically generated when the wedge slams the air-water interface,



(A) $t = 0.4375$ s

(B) $t = 0.4375$ s

(C) $t = 0.5625$ s

(D) $t = 0.5625$ s

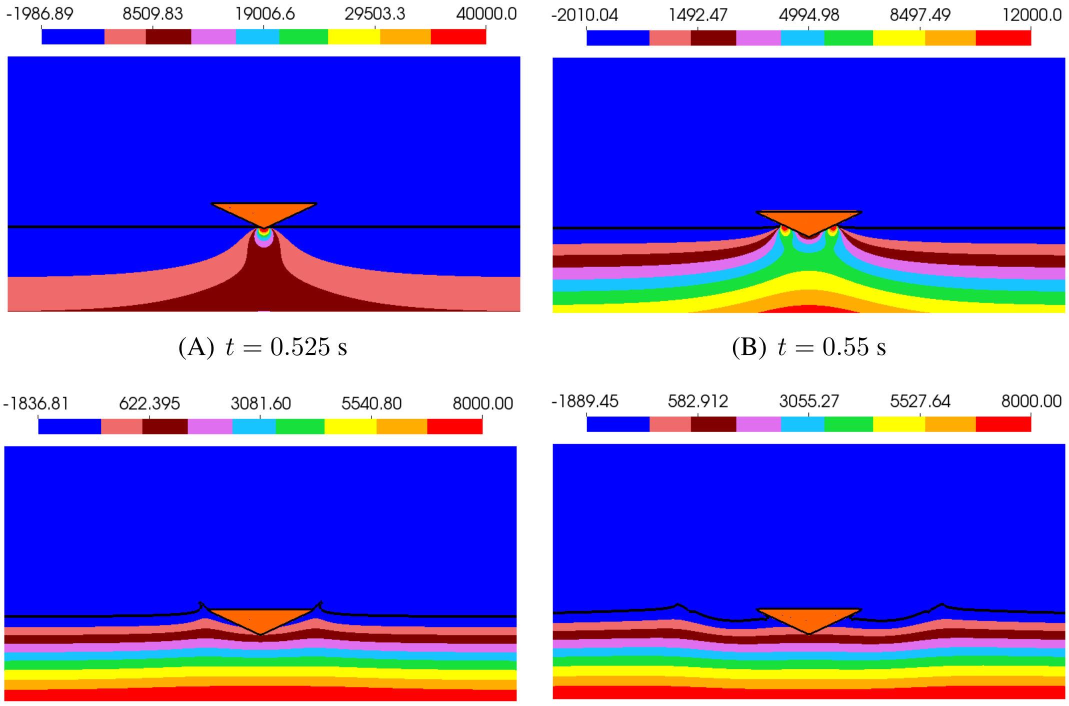
(E) $t = 0.875$ s

(F) $t = 0.875$ s

(G) $t = 1.25625$ s

(H) $t = 1.25625$ s

Figure 3.7. Temporal evolution of free falling 2D wedge slamming into an air-water interface: (left) density and (right) vorticity generated in the range -100 to 100 s^{-1}.



(A) $t = 0.525$ s

(B) $t = 0.55$ s

(C) $t = 0.65$ s

(D) $t = 1.25$ s

Figure 3.8. Temporal evolution of pressure in the domain for a free falling 2D wedge slamming on an air-water interface.

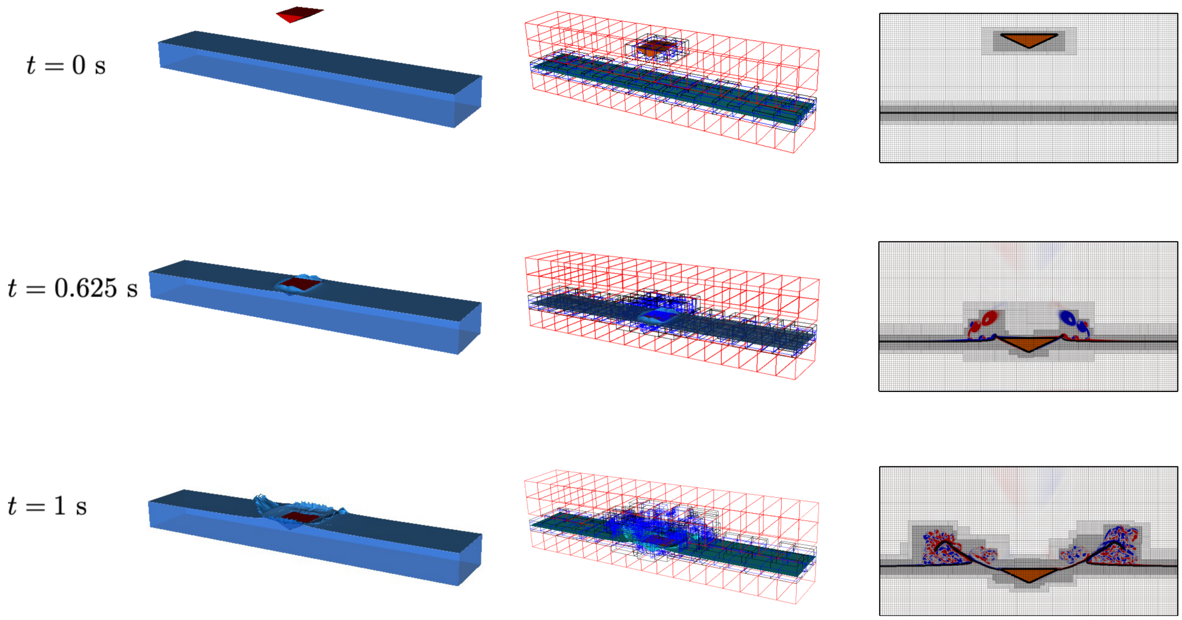


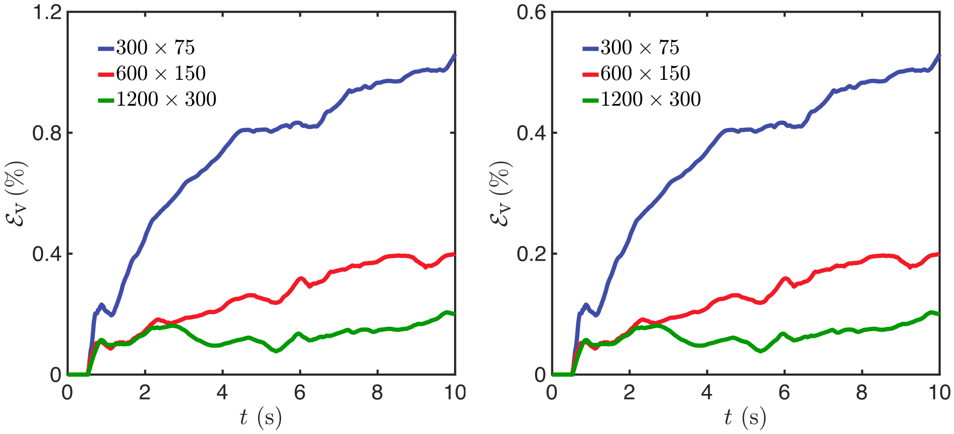
Figure 3.9. Temporal evolution of a free falling 3D wedge slamming on an air-water interface using AMR: (left) density; (center) different mesh levels with the coarsest level shown in red boxes and the finest level shown in blue boxes; and (right) vorticity generated in the air region shown on a 2D slice of the domain at $y = 5L/6$. Vorticity is in the range -50 to 50 s^{-1}.

so that larger vortex structures shed in the air phase can be resolved accurately.

Fig. 3.10 shows the percentage volume (or mass) change for the liquid and gas phases in the 2D wedge case. The percentage error $\mathcal{E}_{\text{v}}$ is defined as $\mathcal{E}_{\text{v}} = \frac{|v(t) - v_0|}{v_0} \times 100$, in which $v(t) = \int_\Omega H^{\text{flow}} \, \mathrm{d}\Omega$ is the volume occupied by the gas phase at time t, and $v_0 = v(t = 0)$ is its volume at the beginning of the simulation. The liquid volume/mass can be defined similarly. The errors are below 1% and they decrease as the grid refines.

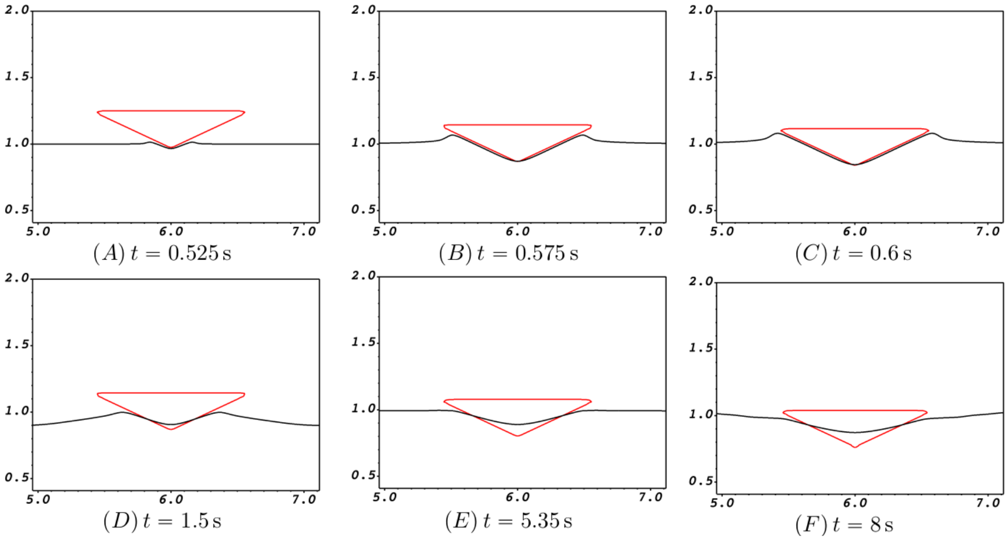
3.5.1 Choosing the right κ value for the multiphase FSI model and its effect on the contact line

In both 2D and 3D wedge problems considered in this section, the permeability coefficient is taken to be $\kappa = \Delta t/\rho^{\text{S}} \sim \mathcal{O}(10^{-8})$. While this value is small, it does not approach machine precision. In the case of a small but non-zero value of κ, the Brinkman penalty term only weakly imposes the no-slip condition on the solid surface. Weak no-slip conditions on solid surfaces can also allow fluid to penetrate into porous solids and wet them. It is also possible to observe this



(A) Percentage change in liquid volume (B) Percentage change in gas volume

Figure 3.10. Percentage volume (or mass) change $\mathcal{E}_\mathrm{v}$ of liquid and gas as a function of time for the two-dimensional wedge case under grid refinement.



$(A)\, t = 0.525\,\mathrm{s}$ $(B)\, t = 0.575\,\mathrm{s}$ $(C)\, t = 0.6\,\mathrm{s}$

$(D)\, t = 1.5\,\mathrm{s}$ $(E)\, t = 5.35\,\mathrm{s}$ $(F)\, t = 8\,\mathrm{s}$

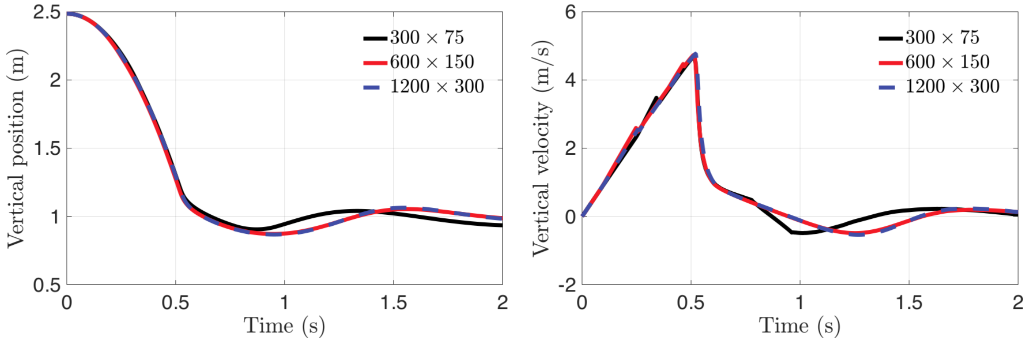
Figure 3.11. Evolution of the zero contours of fluid ϕ (black line) and solid ψ (red line) level set functions.

wetting phenomenon in Fig. 3.11, which illustrates the temporal evolution of the zero contours of the fluid and solid level set functions. When the wedge impacts the water surface, the liquid-gas interface deforms and conforms to the wedge shape. Over time, the liquid-gas interface gradually wets the wedge to conserve mass and to achieve force balance (weight of the wedge = buoyancy force). Once the system reaches mechanical equilibrium (when velocity is zero everywhere after a long time), the air-gas interface flattens out and returns to its original shape.

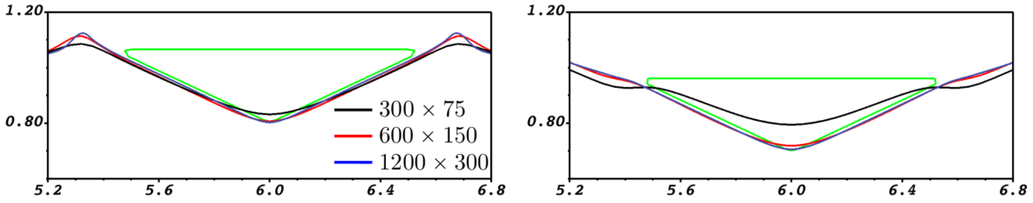
Fluid leakage into an immersed structure is a common trait of immersed boundary (IB) methods that use penalty forces to enforce no-slip conditions. Most IB methods in this category use delta functions proposed by Peskin; for a description of the leakage problem for single phase IB methods see [38, 101, 81, 102, 103]. By lowering κ, it is possible to control flow leakage within the Brinkman/volume penalization approach. However, this will make the system of equations stiff. For the proposed preconditioner, the stiff system of equations is not an issue. The solver remains robust even at very low values of $\kappa \sim \mathcal{O}(10^{-12})$, as demonstrated in the next Sec. 3.6. Nevertheless, very small κ values break a key assumption behind the multiphase FSI model employed here: to conserve mass and to achieve force balance, we need the fluid to penetrate the structure. Moreover, we need to relax the no-slip boundary condition for the tangential velocity components to avoid singularities in tangential stress and pressure at the contact line. Considering the falling wedge problem, we can understand the need for normal penetration of the fluid into the impacting wedge as follows:

The enclosed domain consists of liquid and gas phases at $t = 0$. As far as material mass is concerned, the wedge is not present (recall that the structure domain is fictitious and the immersed body interacts only through the penalty term in the momentum equation). Imagine that the wedge comes to rest after a very long time, and we have imposed the no-penetration boundary condition perfectly throughout the simulation. For the tangential components of velocity we can assume either a slip or no-slip condition for the purpose of this discussion. Eventually, the air-water interface will conform to the wedge geometry (also called the “dry” contact line or the 180 degree contact angle condition), and both the stationary water level and wedge will be in equilibrium. However, this implies that the wedge has displaced the liquid and placed air in the region where it is submerged. At the stationary air-water interface comprising both horizontal and wedge-shaped regions, the hydrostatic pressure level is the same. Physically, this is not possible as the horizontal and curved-down regions of the interface are at different elevations. Instead, the method allows liquid and gas to penetrate the wedge, conserving both phases' mass and achieving the same (hydrostatic) pressure at the air-water interface. In the model, if κ is not too small, this mechanism is permitted. Fluid leakage also means contact angle conditions cannot be controlled directly, and they emerge numerically from mass and force balance conditions. Our simulations demonstrate this in Fig. 3.11. Initially, the contact angle appears to be 180 degrees. However, over time, the fluid penetrates into the porous solid at some numerical or apparent angle less than 180 degrees. A further discussion on the issue of fluid entering the immersed structure is provided at the end of this section.

Based on empirical tests, $\kappa = \Delta t/\rho^{\mathrm{S}}$ represents a robust choice for multiphase FSI models. This results in a continuous air-water interface near the triple points and within the solid, as well as converged FSI dynamics. In Fig. 3.12, we show the convergence of 2D wedge velocity and position, and liquid-gas interface at three grid resolutions: coarse, medium, and fine. Numerical instability occurs when κ is lowered by a factor of 2 or more. In the simulations we observe a sudden large rise in pressure and velocity magnitude at triple points when the wedge meets the air-water interface; very low κ values do not affect the stability of the simulations when the wedge is completely in the air phase. This is because the no-slip condition for the tangential velocity components at the contact line leads to singularities in shear stress and pressure. Huh and Scriven analyzed this situation analytically in their 1971 paper [104], where they considered a solid driven into phase B from phase A (similar to the wedge problem considered in this section). The authors showed that the classical no-slip condition (for the tangential velocity) breaks down as it leads to infinite stress and pressure at the contact line and remarked that "*not even Herkales [Hercules] could sink a solid if the physical model [i.e., the no-slip model] were entirely valid, which it is not.*" As a possible means to remedy this situation, Huh and Scriven



(A) Vertical postion 2D wedge (B) Vertical velocity 2D wedge



$(A)\, t = 0.65\,\mathrm{s}$ $(B)\, t = 1\,\mathrm{s}$

(C) Liquid-gas interface

Figure 3.12. Grid convergence study for the free falling 2D wedge case: (A) vertical position; (B) vertical velocity; and (C) liquid-gas interface at two different time instants. $\kappa = \Delta t/\rho^{\mathrm{S}}$ is used as the penalty factor. A uniform time step size of $\Delta t = 2.5 \times 10^{-4}$ s is employed for the coarse grid, which is halved for each successively refined grid.

suggest using a slip model (see Eq. 37 of [104]) in the tangential direction, which is what the Brinkman penalty term is also doing. Furthermore, the authors in [104] also suggest relaxing the no-slip condition in the normal direction, where they remark "*Relaxing the normal component of the adherence condition, Eqs. [2] and [5], in the immediate vicinity of the contact line—to allow for fracture, for example–is another possibility, but seems to demand slip anyway, along the fluid interface.*" Thus, low but non-zero κ values in the volume penalization approach allow tangential slip and mass flux into the solid. The simulations presented in this section confirm Huh and Scriven's theoretical analysis. Multiple viewpoints (e.g., force and mass balance at the final equilibrium position and singularities in shear stress and pressure) lead to the same conclusion that κ should not be taken too low. Further physical insights on the effect of κ on the contact line/angle are provided in Sec. 3.6.2, where we consider a different multiphase FSI problem (a rigid cylinder heaving on an air-water interface).

Last but not least, we discuss the usefulness of our multiphase FSI model for certain classes of problems (e.g., wave energy converters, water entry/exit of marine structures), despite the fact that a specified contact angle condition cannot be directly imposed and that fluid penetrates the structure. For the former issue, at the scale of these ocean engineering problems, the contact angle boundary condition, or for that matter, the surface tension force as a whole does not affect the device/structure dynamics. As a concrete example, let's consider the water entry of the wedge simulated in this section. The structure length scale is $L = 1.2$ m, whereas the capillary length scale is $l_c \propto \sqrt{\frac{\gamma}{(\rho^{\rm L}-\rho^{\rm G})g}} \approx 2.7$ mm. The structure is $\approx$ 500x larger than the capillary length. The grid resolution that we used (same as that of Pathak et al. [2], who impose static contact angle conditions [the validity of this assumption is questionable during the dynamic phase of impact] over the surface of the impacting wedge) to capture the rigid body dynamics of the wedge is $\Delta x = \Delta y = \Delta z = 0.01$ m or 10 mm, which is 4x larger than the capillary length scale. Now imagine that the wedge comes to rest after a long period of time and the entrained air near the contact line has either dissolved or escaped. In this scenario the static contact angle condition becomes valid. Assuming the wedge is metallic (hydrophilic surface with $\theta_{\rm static} = 60^\circ < \pi/2$) the air-water interface/meniscus will rise by a height of $l_c \cot(\theta_{\rm static}) \approx 1.55$ mm, which is $\approx$ 777x smaller than the wedge dimensions, and falls in the subgrid scale ($\approx$ 10x smaller than the cell size). In more practical ocean engineering problems, marine structures typically span a length of 25 - 500 m. Therefore, it is computationally unfeasible/impractical to resolve capillary length scales/triple points along with structure and wave dynamics. Furthermore, the dynamic contact angle conditions under these highly turbulent and unsteady conditions are not well known/studied.

Regarding the issue of fluid penetrating the structure, technically speaking, this should not happen. If this happens it means that some fluid is taken from the surrounding reservoir. For ocean engineering problems, the reservoir volume is very large in comparison to how much fluid penetrates the structure. Fluid-structure interaction is typically not affected by this. There are specialized numerical techniques (e.g., cut-cell methods) that prevent this from happening [71, 72]. These methods also employ discontinuous interface capturing methods like the volume of fluid method to truncate the air-water interface on the surface of the body. The VP method employing the level set method assumes the air-water interface is continuous which does not truncate on the structure surface. It remains a future endeavor to develop a VP/fictitious domain method that (i) does not permit fluid from entering the structure; (ii) satisfies mass balance and momentum equations; (iii) imposes a specified contact angle condition; and (iv) keeps the air-water interface continuous/smooth. To summarize, the current multiphase FSI model can only be used for problems that do not depend critically upon imposing a specific contact angle condition (i.e., problems with length scale much larger than the capillary length scale), and where the reservoir volume is much larger than the structure volume. As a final note on the multiphase FSI model, there have been several studies in the literature which have ignored contact angle conditions at material triple points, and have allowed the liquid-gas interface to penetrate/exist within the solid region; see for example [105, 106, 107, 15, 16, 108]. In [106] the authors mention that "*In the present work, we assume that the solids are filled with a "virtual" fluid with density and viscosity equal to the largest among all fluids in the domain.*" In [15] the authors show the penetration of the liquid-gas interface into the initially dry solid for their coupled volume of fluid and Brinkman penalization approach. And in [108] the authors show the air-water interface passing through a floating buoy in a numerical wave tank. A coupled level set and immersed boundary method (similar to the VP method) was used in [108].

3.6 Results and discussion

In this section, we consider two nontrivial test problems to demonstrate the efficacy of the projection preconditioner to solve the coupled velocity-pressure system written in Eq. (3.36). In the first problem, a uniform density and viscosity flow is considered in a complex domain. Using the method of manufactured solutions, we compute the spatial order of accuracy of the solution ($\mathbf{u}$ and p), and monitor the number of iterations the outer FGMRES solver takes to converge with decreasing values of κ. In the second case, we study the free-decay of a rigid cylinder heaving on an air-water interface. The heave displacement of the cylinder is compared against literature to assess the accuracy of the FSI solution. The number of iterations taken by the Krylov solver to converge are monitored for this case as well. In contrast to the first problem, the density and viscosity of the three phases (solid, liquid, gas) differ by orders of magnitude in the second problem.

In the tests, the outer FGMRES solver is deemed to be converged if a value of 10^{-9} or below is reached for the norm of the relative residual

$$\mathcal{R} = \frac{||\mathbf{r}||}{||\mathbf{b}||} = \frac{||\mathbf{b} - \mathbf{M}\mathbf{x}||}{||\mathbf{b}||}. \tag{3.45}$$

For velocity and pressure subdomain problems, the inner Richardson solver (that is preconditioned with a single multigrid V-cycle) is set to use only a single iteration. We remark that if the preconditioner employs a fixed number of Richardson iterations (as considered here, which is equal to one), then it is also possible to use the more memory-efficient GMRES solver as the outer Krylov solver. However, here we report the convergence rate of the FGMRES solver (which requires roughly twice the amount of memory compared to GMRES), as it exhibits more uniform convergence behavior across a wide range of thermo-physical parameters, i.e., μ and ρ.

3.6.1 Uniform density and viscosity flow in a complex domain

Consider a computational domain $\Omega \in [0, 2\pi]^2$ which embeds a circular cylinder of radius $R = 1.5$ at its center (π, π) as shown in Fig. 3.13. A steady state manufactured solution for velocity $\mathbf{u}$ and pressure p

$$u_{\text{exact}}(\mathbf{x}, t \to \infty) = \sin(x)\cos(y), \tag{3.46}$$

$$v_{\text{exact}}(\mathbf{x}, t \to \infty) = -\cos(x)\sin(y), \tag{3.47}$$

$$p_{\text{exact}}(\mathbf{x}, t \to \infty) = \sin(x)\sin(y), \tag{3.48}$$

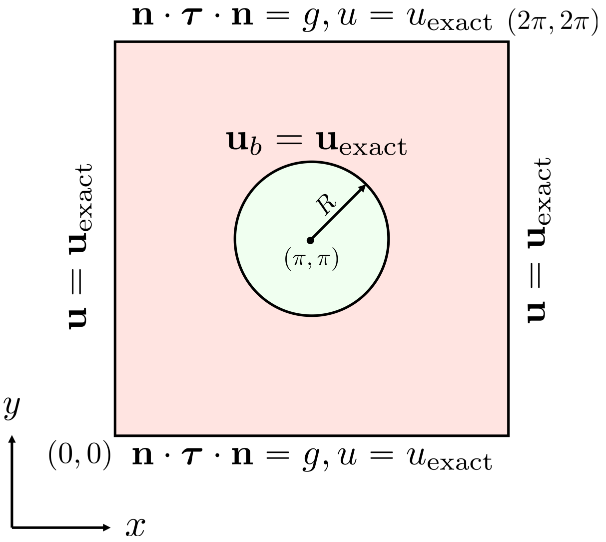


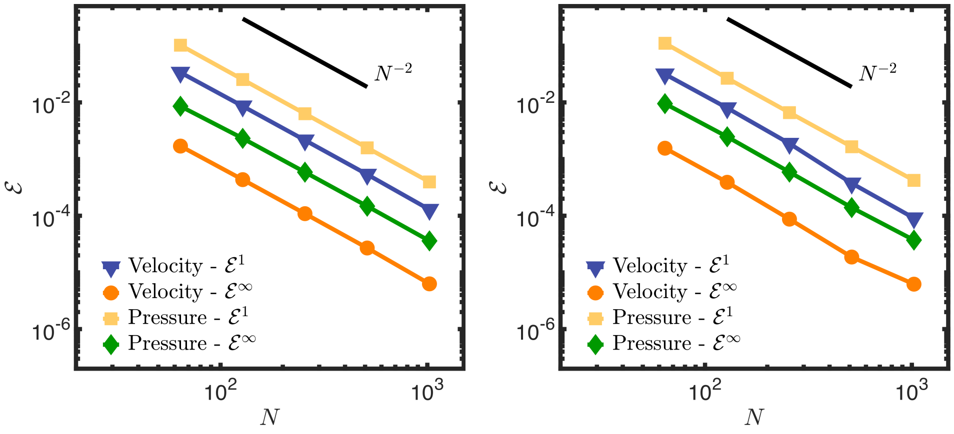
Figure 3.13. Schematic of the computational domain. Here, $g = -p + 2\mu\frac{\partial v}{\partial y} = \sin(x)\sin(y) - 2\cos(x)\cos(y)$ is the imposed normal traction on the top and bottom boundaries. For the top boundary $\mathbf{n} = (0, 1)$ and for the bottom boundary $\mathbf{n} = (0, -1)$.

is used to drive a constant density $\rho(\mathbf{x}, t) \equiv 1$ and viscosity $\mu(\mathbf{x}, t) \equiv 1$ flow in the domain. Specifically, Eqs. (3.46)-(3.48) are plugged into the momentum Eq. (5.6) to determine the body force $\mathbf{f}$ that drives the flow. The manufactured solution is also used to impose boundary conditions on $\partial\Omega$ and inside the fictitious cylinder Ω_b. On the left and right ends of the domain, we impose velocity boundary conditions $u = u_{\text{exact}}$ and $v = v_{\text{exact}}$. On the top and bottom boundaries a combination of normal traction $\mathbf{n}\cdot\boldsymbol{\tau}\cdot\mathbf{n} = g = -p + 2\mu\frac{\partial v}{\partial y} = \sin(x)\sin(y) - 2\cos(x)\cos(y)$ and tangential velocity $u = u_{\text{exact}}$ condition is imposed. Here, $\boldsymbol{\tau} = -p\mathbf{I} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathsf{T}}\right)$ denotes the hydrodynamic stress tensor. The velocity inside the cylinder is prescribed to be $\mathbf{u}_b = \mathbf{u}_{\text{exact}}$. The initial conditions for velocity and pressure are taken to be zero.

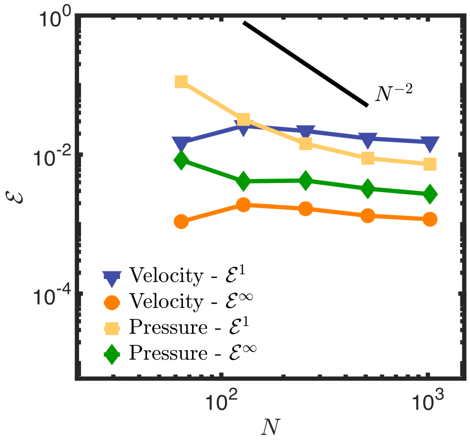
Five grid sizes $N \times N = \{64^2, 128^2, 256^2, 512^2, 1024^2\}$ are used to run the simulations starting from $t = 0$ till steady state is reached. $n_{\text{smear}} = 1$ is used in Eq. (3.23) to smear the fluid-solid interface. A constant time step size of $\Delta t = 1 \times 10^{-3}$ is used in all simulations, which maintains convective CFL number below 0.35 for all grid sizes N. The permeability coefficient κ should be kept small, but not too small to avoid the plateauing of spatial discretization errors [7, 15]. To study the effect of κ on the spatial order of accuracy of $\mathbf{u}$ and

p solutions, as well as on the solver convergence rate, we consider three different values for $\kappa = \{\Delta t/\rho,\ \Delta t/100\rho,\ \Delta t/10000\rho\}$ in the numerical experiments. Note that $\Delta t/\rho$ is the maximum value which κ can or should take as per the inertial scale $\chi/\kappa \sim \rho/\Delta t$. Another possibility is to select κ based on the viscous scale $\chi/\kappa \sim \mu/h^2$, in which h is the uniform cell size. Here, our strategy is to start with the maximum value of κ based on the inertial scale, and then reduce it progressively till no further improvement in the solution is observed.

We first present the spatial order of accuracy of the Brinkman penalized $\mathbf{u}$ and p solutions. The error between steady state numerical and analytical solutions is denoted $\mathcal{E}$. As can be observed in Fig. 3.14, second-order pointwise (L^∞-norm) convergence rate is obtained for both velocity and pressure errors when $\kappa = \Delta t/\rho,\ \Delta t/100\rho$. For a very small value of $\kappa = \Delta t/10000\rho = 1 \times 10^{-7}$, the error in velocity and pressure saturates after a certain grid size ($N = 128$ in Fig. 3.14(C)). To understand this trend, we compare the magnitudes of the Brinkman penalty coefficient χ/κ and the discrete inertial and viscous scales, by plotting the latter two as a function of grid size N in Fig. 3.15. The discrete inertial scale $\rho/\Delta t$ remains constant (because of the constant time step size Δt), whereas the discrete viscous scale varies quadratically (linearly on a log scale) with N. The magnitude of the inertial scale is larger than the viscous scale for the first two grids, and vice versa for the remaining grids. For the largest value of $\kappa = \Delta t/\rho$, although the penalty coefficient $\chi/\kappa = \rho/\Delta t$ is not always larger than the viscous scale, second-order convergence is still observed. When $\kappa = \Delta t/100\rho$, the Brinkman penalty coefficient is significantly larger than the inertial scale and comparable with the viscous scale. Second-order pointwise convergence is obtained for this κ value as well. However, when κ is decreased further to $\Delta t/10000\rho$, the Brinkman penalty becomes four orders larger than the inertial scale and two orders larger than the viscous scale. This large penalty value causes the errors to saturate. Note that in this problem we imposed spatially-varying traction boundary condition on the top and bottom boundaries of $\partial\Omega$, and obtained second-order convergence rates for velocity and pressure solutions. This is not possible to achieve if pressure and velocity are solved in a split manner using projection method as a solver.



(A) $\kappa = \Delta t/\rho$ (B) $\kappa = \Delta t/100\rho$



(C) $\kappa = \Delta t/10000\rho$

Figure 3.14. Spatial order of convergence of the volume penalized Navier-Stokes system using manufactured solutions and different values of κ.

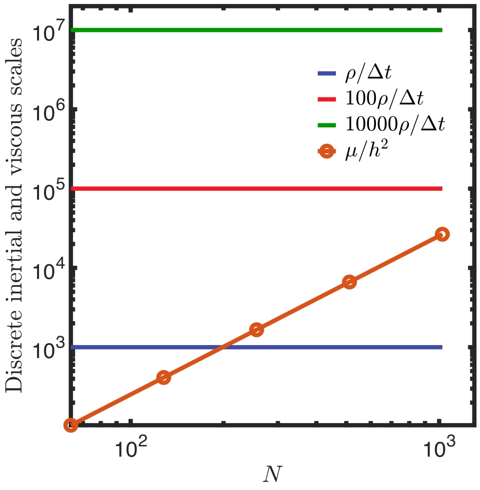
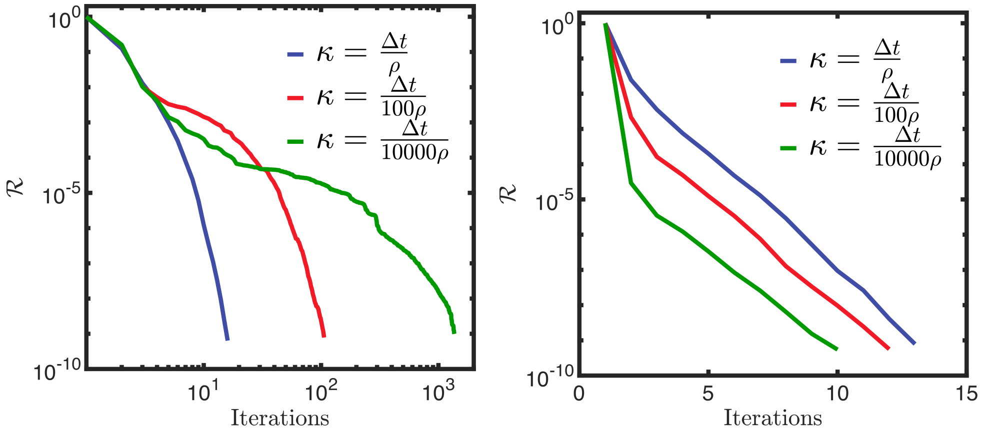


Figure 3.15. Discrete inertial and viscous scales as a function of grid size N.



(A) Relative residual when $\theta = 0$ (B) Relative residual when $\theta = 1$

Figure 3.16. Convergence rate of the preconditioned FGMRES solver during the first time step of the simulation (A) without and (B) with the Brinkman penalty term in the projection preconditioner. The grid size is $N = 256$.

Next, we study the impact of κ on the convergence rate of the preconditioned FGMRES solver. For this problem, the first time step poses the most difficultly for the (iterative) Krylov solver as the initial zero guess for velocity and pressure is far-off from the true solution at $t = \Delta t$. Therefore, it suffices to monitor the solver performance at the first time step only to evaluate the efficacy of the preconditioner. We run this test case with and without the Brinkman penalty term $\widetilde{\boldsymbol{\chi}}/\kappa$ in the projection preconditioner. In practice, we re-define $\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}$ with the help of a boolean parameter θ

$$\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1} = \left(\breve{\boldsymbol{\rho}} + \theta \frac{\widetilde{\boldsymbol{\chi}}\Delta t}{\kappa}\right)^{-1}, \tag{3.49}$$

so that by setting $\theta = 1$ in $\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}$ we obtain the new projection method and by setting $\theta = 0$ we revert to the projection methods of Bergmann and Iolla [11] and Sharaborin et al. [15].

Fig. 3.16 compares the convergence rate of the projection method preconditioned FGMRES solver for different κ values. The grid size is taken to be $N = 256$. When the Brinkman penalty is excluded from the projection step (i.e., $\theta = 0$ in Eq. (3.49)), we note from Fig. 3.16(A) that the number of iterations required to convergence to a relative residual of $\mathcal{R} = 10^{-9}$ increases approximately by a factor of 10 with decreasing values of κ. However, with the proposed

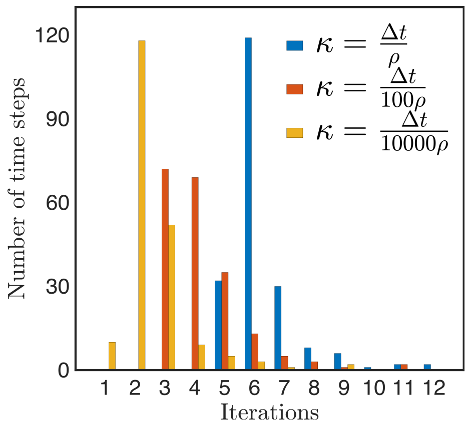


Figure 3.17. Performance of the preconditioned ($\theta = 1$) FGMRES solver during the first 200 time steps of the simulations. The average number of iterations required to converge for $\kappa = \Delta t/\rho$, $\kappa = \Delta t/100\rho$, and $\kappa = \Delta t/10000\rho$ are 6, 4 and 2, respectively. The grid size is $N = 256$. Here, only a single-cycle of fixed-point iterations is considered.

projection method (i.e., $\theta = 1$ in Eq. (3.49)) the convergence of the FGMRES solver remains robust. This can be observed in Fig. 3.16(B) where the solver converges with approximately 10 iterations for all three κ values. This clearly demonstrates the importance of including the Brinkman term in the projection method. In fact, with decreasing κ (or increasing penalty) values, the convergence rate of the solver actually improves. This can be attributed to $\mathbf{A}$ matrix of Eq. (3.37) which becomes diagonally dominant when the penalty coefficient is larger than the viscous scale. This in turn makes the velocity subdomain problem "easier" to solve.

To present a more complete picture of the solver performance as the simulation progresses, Fig. 3.17 presents the number of iterations to converge for the first 200 time steps. Here, only a single cycle of fixed-point iterations is employed as the number of FGMRES iterations reduces (substantially) at iteration 2 and beyond. Thus, the "worst-case-scenario" is considered. The grid size is taken to be $N = 256$ and the same three κ values are considered. It is clearly seen that with decreasing κ values or conversely with increasing penalty values, the average number of iterations to converge decreases.

Thus far in this section, we demonstrated the efficacy of the new projection preconditioner on a single grid $N = 256$. To demonstrate that the proposed preconditioner is scalable, Fig. 3.18

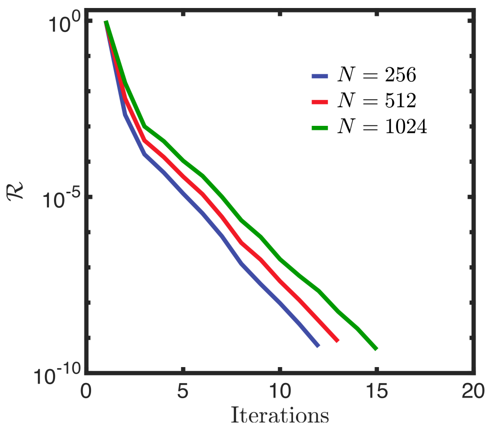


Figure 3.18. Convergence rate of the preconditioned ($\theta = 1$) FGMRES solver during the first time step of the simulation on three grids: $N = 256, 512, 1024$. The permeability parameter is taken to be $\kappa = \Delta t/100\rho$.

reports the convergence rate of the preconditioned FGMRES solver on three grids: $N = 256, 512, 1024$. The permeability parameter is taken to be $\kappa = \Delta t/100\rho$, which is larger than both inertial and viscous scales as discussed earlier. In Fig. 3.18 it can be observed that the solver's convergence rate remains approximately the same, even when the degrees of freedom, $\mathbf{x_u}$ and $\mathbf{x_p}$, increase substantially with increasing grid size.

Based on the results presented in this section we conclude that: (**1**) the proposed projection method is a scalable preconditioner for the volume penalized incompressible Navier-Stokes system; (**2**) it is possible to achieve pointwise second-order accuracy in velocity and pressure solutions with nontrivial traction boundary conditions without sacrificing computational efficiency; and (**3**) a reasonable starting value for the permeability parameter is $\kappa = \Delta t/\rho$. In the next section we consider a three phase FSI problem to demonstrate that the proposed projection preconditioner remains effective even with spatially varying ρ and μ.

3.6.2 Free-decay of a rigid cylinder heaving on an air-water interface

We consider a two-dimensional computational domain of extents $\Omega \in [0, L] \times [0, H]$ to simulate the heaving motion of a 2D cylinder on an air-water interface; see Fig. 3.19. The length of the domain is L = 10 m, height is $H = 0.2L$, and the origin O is located at the bottom

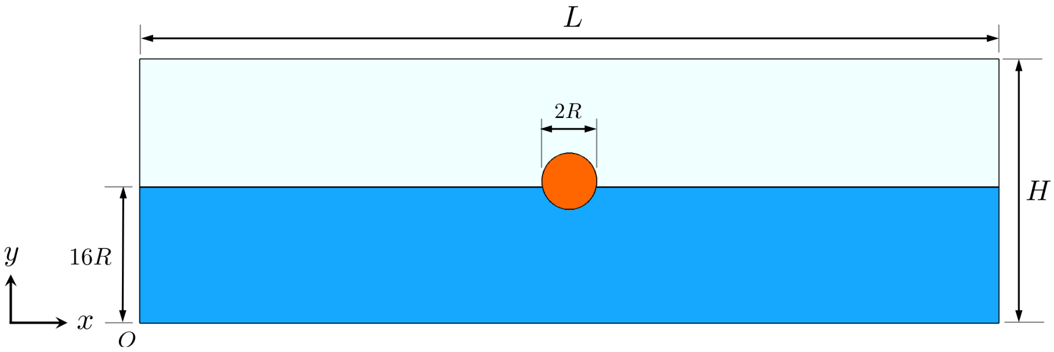


Figure 3.19. Schematic of a 2D cylinder heaving on an air-water interface.

left corner. The radius of the cylinder is R = 0.0762 m. Water of density ρ^{L} = 1000 kg/m^3 and viscosity $\mu_{\text{l}} = 10^{-3}$ Pa·s occupies the computational domain from y = 0 to y = $16R$ and air of density ρ^{G} = 1 kg/m^3 and viscosity $\mu_{\text{g}} = 1.8 \times 10^{-5}$ Pa·s occupies the domain from $y = 16R$ to $y = H$. The solid cylinder is half-buoyant and has a density of ρ^{S} = 500 kg/m^3. Its fictitious viscosity μ_{s} is taken to be same as that of water. The initial center of the cylinder is located slightly above the air-water interface $(x_c, y_c) = (L/2, 16R + R/3)$ from where it is released. The cylinder decays freely under the action of gravity and hydrodynamic forces. The cylinder's heave (vertical) degree of freedom is free, whereas its surge (horizontal) and pitch (rotational) motions are locked. No-slip (zero-velocity) boundary conditions are imposed along $\partial\Omega$. To smoothly transition between different material properties while keeping the liquid-gas and fluid-solid interfaces sharp, one grid cell on either side of the interface is considered ($n_{\text{smear}} = 1$ in Eqs. (3.22) and (3.23)). The surface tension of water does not affect the dynamics of the cylinder at this scale and is neglected in the simulation.

The domain is discretized with a uniform grid of size 6000 $\times$ 1200 such that there are 46 grid cells per radius of the cylinder. A constant time step of $\Delta t = 10^{-3}$ s is used which maintains the convective CFL number below 0.5. The permeability coefficient is taken to be $\kappa = \Delta t/\rho^{\text{S}}$. Newton's second law is used to compute the rigid body velocity of the cylinder $\mathbf{u}_b$, which requires integrating hydrodynamic traction over the irregular surface of the immersed body. This procedure is explained in our prior works [41, 40]. Fig. 3.20 shows the temporal

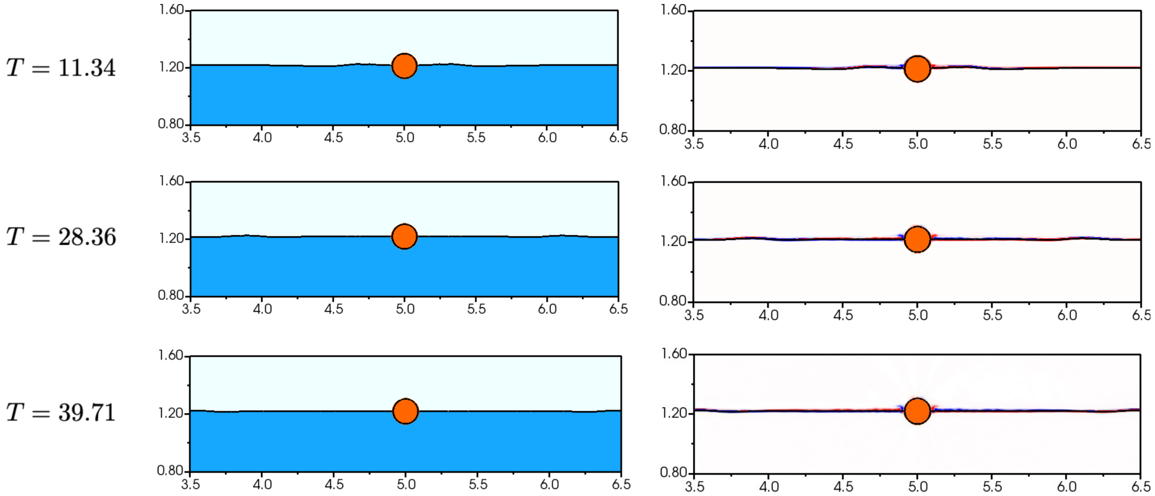


Figure 3.20. Temporal evolution a cylinder heaving on an air-water interface: (left) density and (right) vorticity generated in the range -1 to 1 s^{-1}.

evolution of the heaving cylinder at various time instances. The vertical displacement of the cylinder is shown in Fig. 3.28 and compared with prior immersed boundary (IB) simulations of Nangia et al. [4]. Here, $T = t\sqrt{g/R}$ and $\bar{y} = 3y/R$ denote the non-dimensional time and vertical displacement of the cylinder, respectively. An excellent agreement is obtained with the prior study. After a while, the heaving motion of the cylinder ceases and its center coincides with the free water surface. An illustration of the temporal evolution of the zero contours of the fluid ϕ and solid ψ level set functions can be found in Fig. 3.22. Within the cylinder and near the material triple points, the air-water interface remains continuous.

Fig. 3.23 shows the percentage volume/mass change of liquid and gas over time. $\kappa = \Delta t/\rho^{\mathrm{S}}$ is taken and $\mathcal{E}_{\mathrm{v}}$ is computed numerically as described in Sec. 3.5. There is a very small percentage change in volume, less than 0.006%. The $\mathcal{E}_{\mathrm{v}}$ value decreases further under grid refinement.

Effect of κ on the contact line motion and solver performance

Unlike the falling wedge problem considered in Sec. 3.5, where the solid evolved from a non-wet state to a wet state, here the rigid cylinder starts out in the wet state at $T = 0$. Recall that the liquid-gas interface could not penetrate the wedge when κ was lowered below $\Delta t/\rho^{\mathrm{S}}$ (by

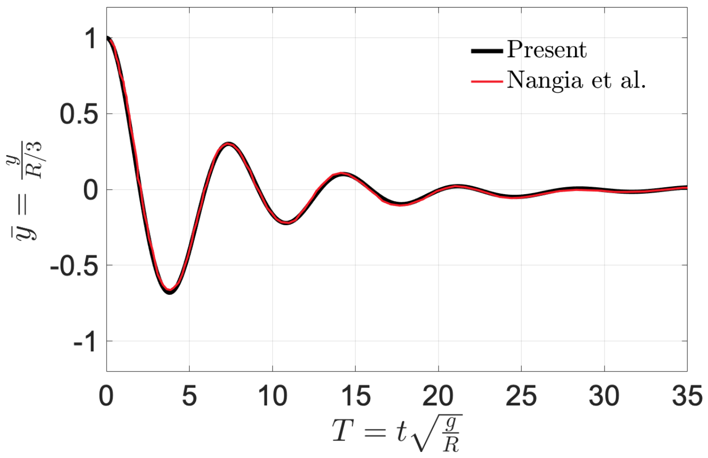


Figure 3.21. Temporal evolution of the non-dimensional vertical displacement $\bar{y}$ of the cylinder. The heave displacement of the cylinder is compared against the prior simulations of Nangia et al. [4]

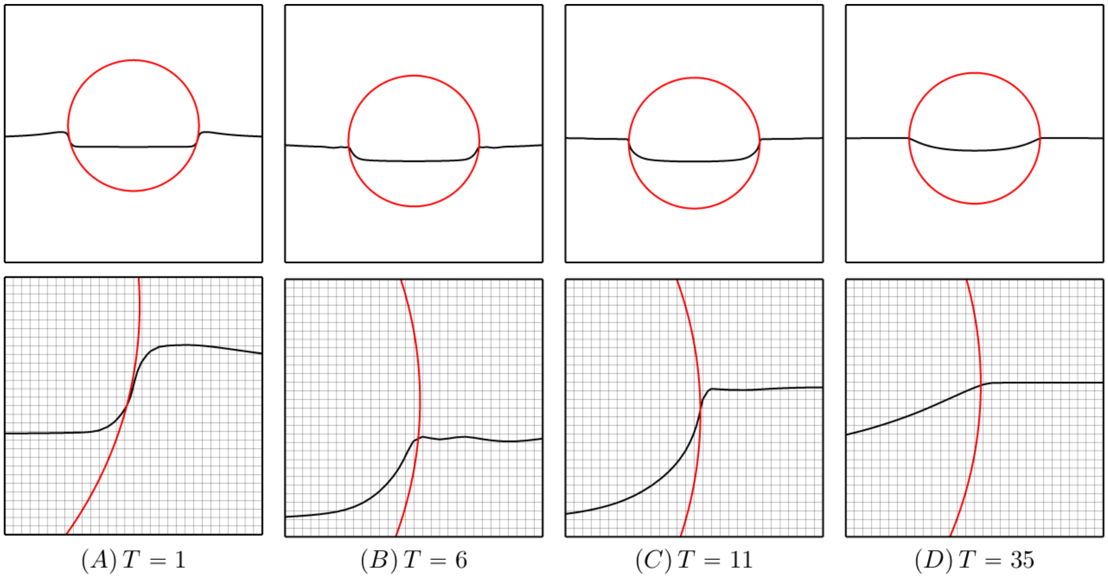
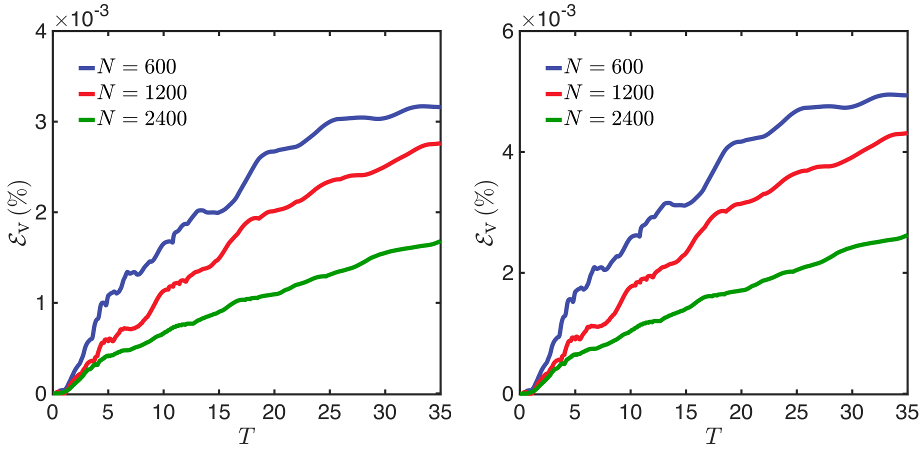


Figure 3.22. Evolution of the zero contours of fluid ϕ (black line) and solid ψ (red line) level set functions. An enlarged view of the liquid-gas interface near the cylinder's right side is shown in the bottom row.



(A) Percentage change in liquid volume (B) Percentage change in gas volume

Figure 3.23. Percentage volume change $\mathcal{E}_{\rm v}$ of liquid and gas over time for the heaving cylinder case. Three different grid resolutions are considered. A uniform time step size of $\Delta t = 2 \times 10^{-3}$ is used for the coarsest grid $N = 600$ and it is halved for each successive finer grid. The permeability coefficient is taken to be $\kappa = \Delta t/\rho^{\rm S}$.

a factor of 2 or more) and numerical instability occurred due to the breakdown of the classical no-slip condition for the contact line. The heaving cylinder case also presents a similar scenario. The wet cylinder attempts to reach a dry state by conforming the contact line around its outer periphery. This is observed when κ is further lowered to $\kappa = \{\Delta t/10\rho^{\rm S}, \Delta t/10000\rho^{\rm S}\}$. An illustration of the situation is shown in Fig. 3.24, which compares contact line dynamics at three different κ values. Numerical instabilities arise around $T = 25$ with $\kappa = \Delta t/10000\rho^{\rm S}$. Despite trying to dry out, the cylinder is unable to do so due to the combination of mass and force balance conditions. In the intermediate case of $\kappa = \Delta t/10\rho^{\rm S}$, numerical instabilities do not occur due to imperfectly imposed no-slip conditions. One can clearly see, however, the contact line's tendency to align with the cylinder surface (the 180 degree contact angle condition) in the intermediate case as well. The heave dynamics of the cylinder for the three κ values are compared in Fig. 3.25. There is a very good match between all three curves and the results reported in the literature (except for the numerical instability in the lowest κ case).

Next, we examine the effect of κ on the convergence rate of the preconditioned FGMRES solver by considering three different values of $\kappa = \{\Delta t/\rho^{\rm S},\ \Delta t/100\rho^{\rm S},\ \Delta t/10000\rho^{\rm S}\}$. Here

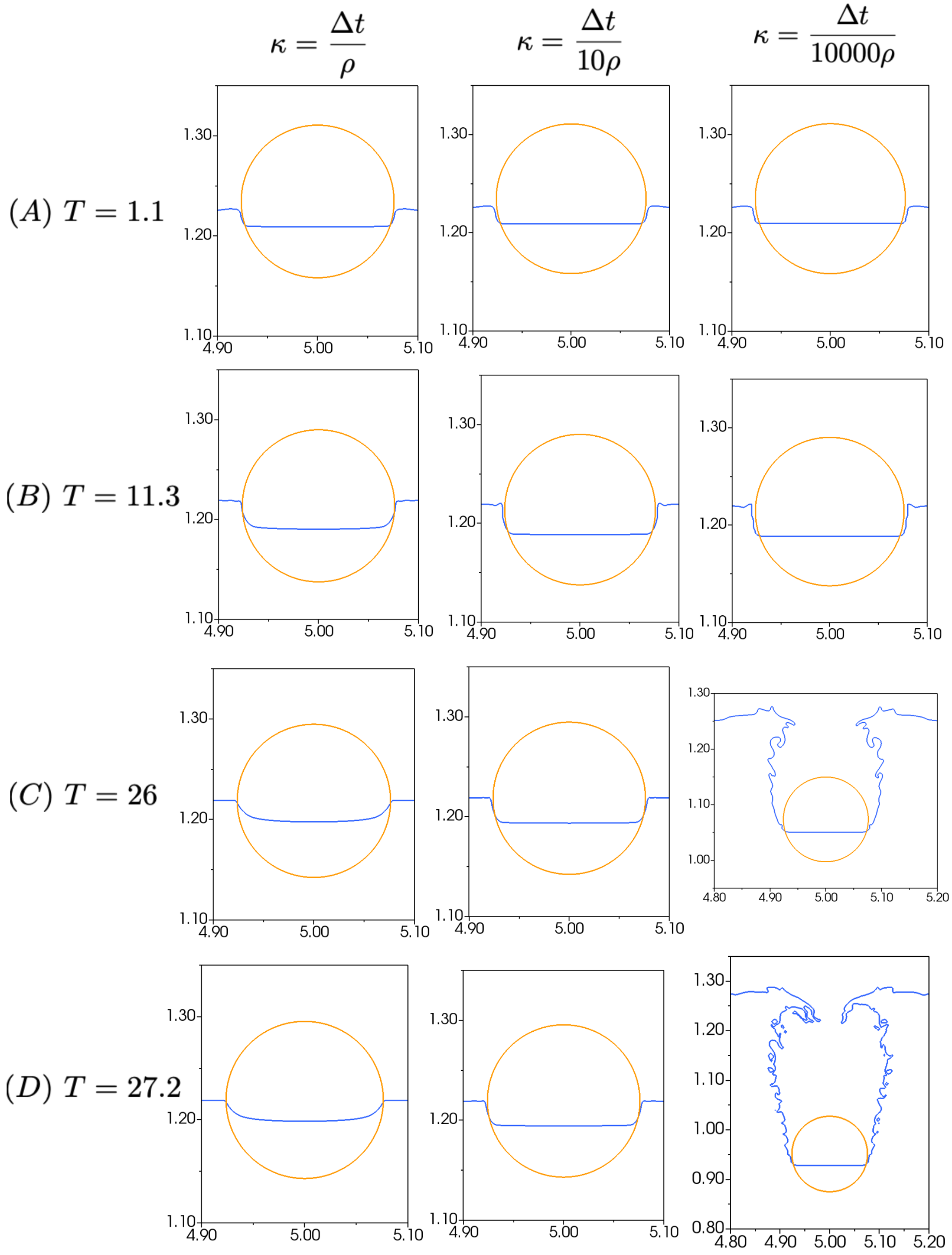


Figure 3.24. Evolution of the zero contours of fluid ϕ (blue line) and solid ψ (mustard line) level set functions using three different values of $\kappa = \{\Delta t/\rho^{\mathrm{S}}, \Delta t/10\rho^{\mathrm{S}}, \Delta t/10000\rho^{\mathrm{S}}\}$.

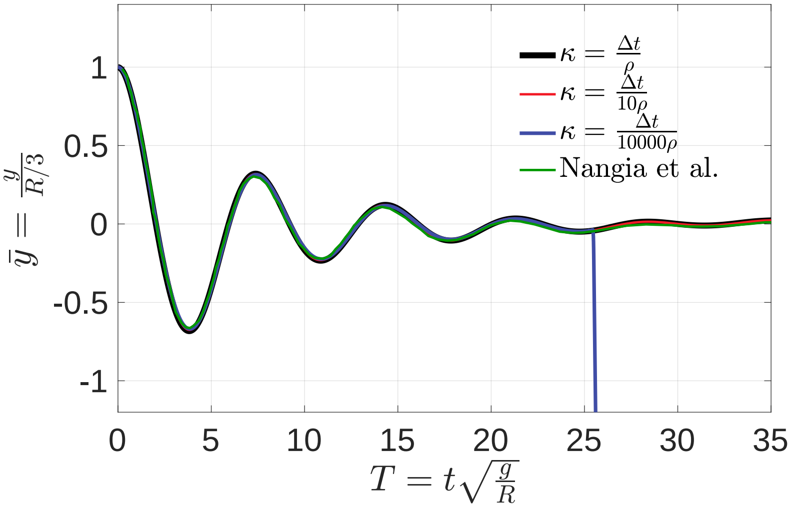
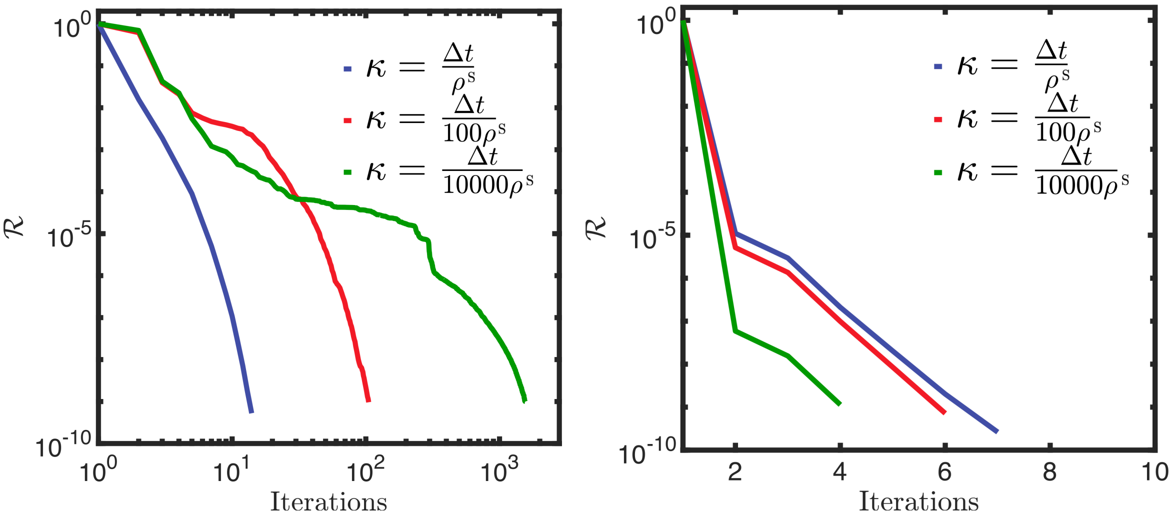


Figure 3.25. Temporal evolution of the non-dimensional vertical displacement $\bar{y}$ of the cylinder using three different values of $\kappa = \{\Delta t/\rho^{\mathrm{S}}, \Delta t/10\rho^{\mathrm{S}}, \Delta t/10000\rho^{\mathrm{S}}\}$. The heave displacement of the cylinder is compared against the prior study of Nangia et al. [4]. The sudden drop in the cylinder displacement is due numerical instabilities that arise for the lowest κ case.



(A) Relative residual when $\theta = 0$

(B) Relative residual when $\theta = 1$

Figure 3.26. Convergence rate of the FGMRES solver during the first time step of the simulation for the multiphase case of a cylinder heaving on the air-water interface.

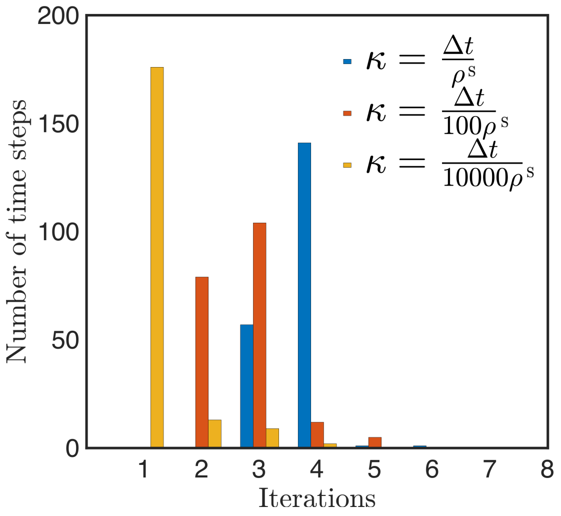


Figure 3.27. Performance of the preconditioned ($\theta = 1$) FGMRES solver during the first 200 time steps of the simulations for the multiphase case of cylinder heaving on the air-water interface. The average number of iterations required to converge for $\kappa = \Delta t/\rho^{\mathrm{S}}$, $\kappa = \Delta t/100\rho^{\mathrm{S}}$, and $\kappa = \Delta t/10000\rho^{\mathrm{S}}$ are 4, 3 and 1, respectively. Here, only a single cycle of fixed-point iterations is employed.

we are interested in testing the performance of the preconditioner at extremely low values of κ, despite the fact that very low values of κ (which lead to the no-slip condition for the contact line) are not consistent with the multiphase FSI model [104]. The solver performance is monitored only at the first time step for the reasons explained in the previous Sec. 3.6.1. The tests are run with ($\theta = 1$) and without ($\theta = 0$) the Brinkman penalty term $\widetilde{\chi}/\kappa$ in the projection preconditioner; see Eq. 3.49. Fig. 3.26 compares the convergence rate of the solver for the three κ values. As observed in Fig. 3.26(A), without the penalty term, the number of iterations required to converge to a relative residual of $\mathcal{R} = 10^{-9}$ increases approximately by a factor of 10 for decreasing values of κ. With the new projection algorithm, the convergence of the solver remains robust and it convergences within 7 iterations as illustrated in Fig. 3.26(B). Further, Fig. 3.27 shows the number of iterations to converge for the first 200 time steps for the three κ values. Here, only a single cycle of fixed-point iterations is employed as the number of FGMRES iterations reduces (substantially) at iteration 2 and beyond. Similar to the results of the previous section, with decreasing κ values, the average number of iterations to converge decreases. Based upon

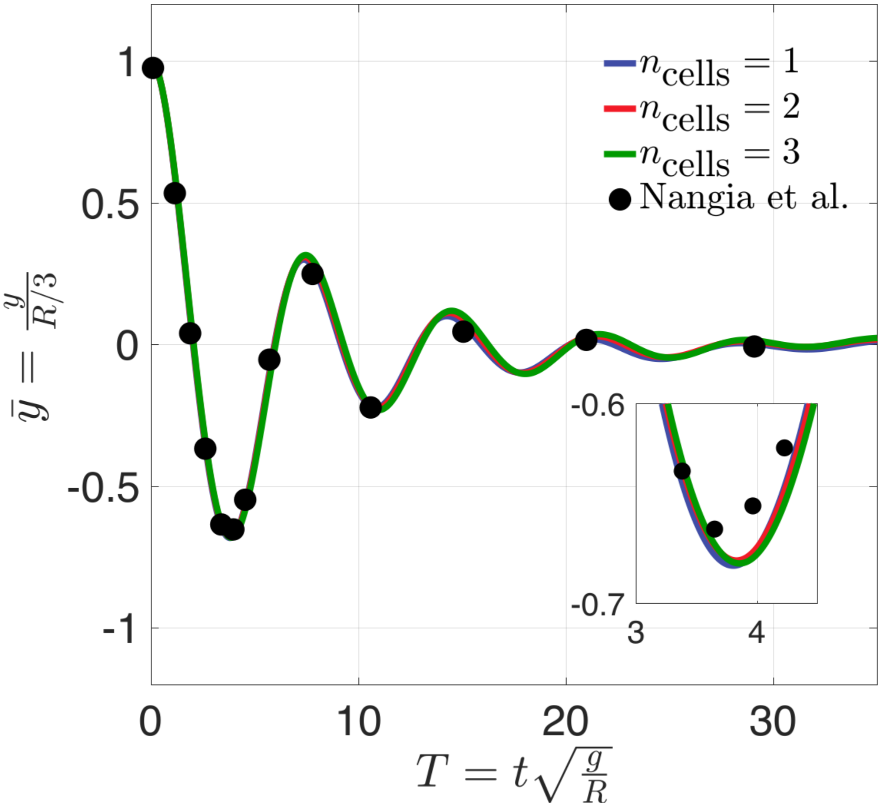


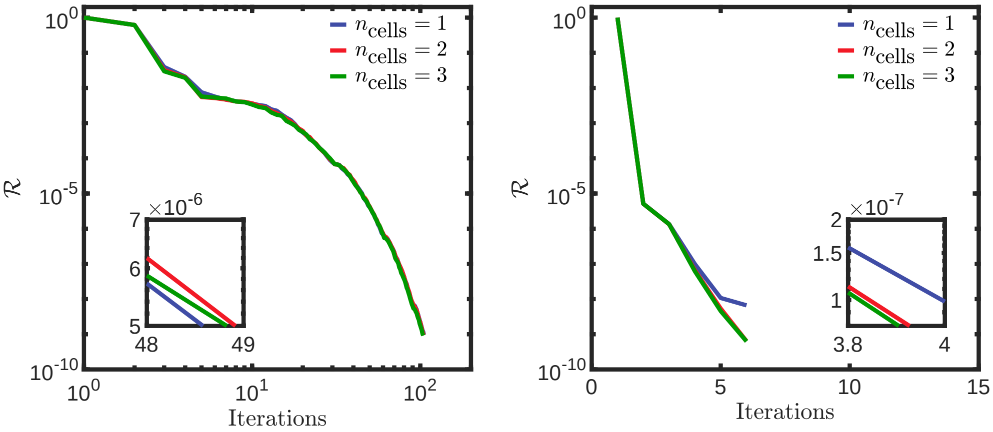
Figure 3.28. Temporal evolution of the non-dimensional vertical displacement $\bar{y}$ of the cylinder for different n_{smear} values. The heave displacement of the cylinder is compared against the prior immersed boundary (IB) simulations of Nangia et al. [4]

the results of this and the previous section, it can be concluded that the projection preconditioner is an effective and scalable strategy for both single and multiphase VP INS systems.

Effect of n_{smear}

We investigate how n_{smear} values affect the solution accuracy, linear solver convergence rate, and mass/volume of conserved phases. We consider three different n_{smear} values, n_{smear} = 1, 2, and 3, to solve the heaving cylinder problem on a uniform 6000×1200 grid. $\kappa = \Delta t/\rho^{\text{S}}$ for all cases considered in this subsection. The vertical displacement of the cylinder using different n_{smear} values is plotted in Fig. 3.28. As can be observed, there is no significant change in the numerical solution and it is similar to the prior IB simulations of Nangia et al. [4].

Using different values of n_{smear}, we compare the convergence rate of the linear solver



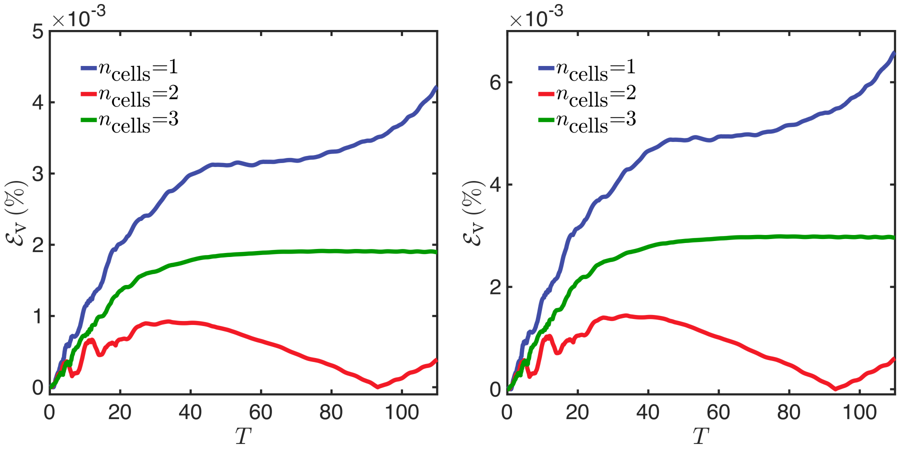
(A) Relative residual when $\theta = 0$ (B) Relative residual when $\theta = 1$

Figure 3.29. Convergence rate of the preconditioned FGMRES solver during the first time step of the simulation (A) without and (B) with the Brinkman penalty term in the projection preconditioner. The grid size is $N = 256$.

during the first time step. Fig. 3.29(A) illustrates the linear solver's convergence rate with $\theta = 0$. When $\theta = 0$, the solver requires more iterations to converge. However, the convergence rates are not significantly affected by n_{smear} values. We achieve a robust convergence rate when the Brinkman penalty is included in the projection preconditioner, i.e., when $\theta = 1$ as shown in Fig. 3.29(B). There is no significant difference in convergence rates in this case either with different values of n_{smear}.

Finally, we compute the percentage volume change in liquid and gas over time as a function of n_{smear}, and is shown in Fig. 3.30. A longer duration of simulation time T is considered here. We consider a uniform grid of size 6000×1200 with a constant time step size of $\Delta t = 10^{-3}$ s. Again, n_{smear} values do not significantly affect $\mathcal{E}_{\text{v}}$. The errors appear in the same order for all n_{smear} values.

While increasing the n_{smear} value does not affect the numerical solution quantitatively, it does make the solid-liquid and liquid-gas interfaces more diffuse. This produces diffuse vortex structures as well. Therefore, we do not recommend using $n_{\text{smear}} > 2$ in the simulations.



(A) Percentage change in liquid volume (B) Percentage change in gas volume

Figure 3.30. Percentage change $\mathcal{E}_{\text{v}}$ in the volume of liquid and gas over time for the heaving cylinder case with three n_{smear} values.

3.7 Conclusions

This work presented a scalable preconditioner for the monolithic solution of the volume penalized single and multiphase incompressible Navier-Stokes equations. The same preconditioner can also be used to solve volume penalized low Mach Navier-Stokes equations monolithically. The preconditioner uses a projection algorithm that correctly accounts for the Brinkman penalty term in both pressure Poisson and velocity update equations. The accuracy and scalability of the solver under grid refinement were discussed. In contrast to prior experience, the solver convergence rate improves when the permeability parameter is decreased. However, the penalized solution's error saturates at extremely low values of permeability. The inertial scale ($\kappa = \Delta t/\rho$) can be used as a reasonable starting point for permeability parameter, which can be gradually lowered until no further improvement in the penalized solution is observed. Our robust preconditioner allowed us to consider low κ values for the multiphase FSI model. It was demonstrated that a dry contact line condition is imposed over the surface of the immersed body at low values of κ. In addition, low κ values imply a no-slip condition, which leads to singularities in shear stress and pressure at the contact line [104]. In multiphase FSI simulations, low but not too small κ values should be used. $\kappa = \Delta/\rho^{\mathrm{S}}$ is a robust choice. The robust preconditioner enabled us to select the right penalty value to arrest the solidified surface in the phase change problem. Additionally, we studied how κ affected solution accuracy for manufactured solutions with our proposed preconditioner. We also showed that the use of the standard level set method leads to (an acceptable level of) volume loss of the conserved fluid phases. This loss, however, is not due to the Brinkman penalty. This can be further confirmed from the coupled Brinkman penalization and volume of fluid (VoF) simulations of Sharaborin et al. [15]. These VoF simulations demonstrate mass/volume loss close to machine accuracy. Lastly, we discussed the limitations of the multiphase FSI model used in this and other works [105, 106, 107, 15, 16] in Sec. 3.5.1, which are (i) fluid penetrating into the solid; and (ii) inability to impose contact angle conditions. It was discussed that despite these limitations the model gives a converged FSI solution for a certain class of problems (e.g., in ocean engineering) that have length scales much larger than the capillary length scale and where the reservoir volume is much larger than the structure volume. It remains a future endeavor to develop a VP/fictitious domain method that overcomes these two limitations while keeping the air-water interface smooth/continuous and satisfying the mass balance and momentum equations.

Acknowledgement

Chapter 3, in full, is a reprint of the material as it appears in An effective preconditioning strategy for volume penalized incompressible/low Mach multiphase flow solvers. Thirumalaisamy, Ramakrishnan; Khedkar, Kaustubh; Ghysels, Pieter; Bhalla, Amneet Pal Singh, Journal of Computational Physics, vol. 490, 112325, 2023. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4

A low Mach enthalpy method to model non-isothermal gas–liquid–solid flows with melting and solidification

This chapter presents a novel low Mach enthalpy method for simulating solidification and melting problems with variable thermophysical properties, including density. Additionally, this formulation allows coupling a solid–liquid PCM with a gas phase in order to simulate the free surface dynamics of PCMs undergoing melting and solidification. We revisit the two-phase Stefan problem involving a density jump between two material phases. We propose a possible means to include the kinetic energy jump in the Stefan condition while still allowing for an analytical solution. The new low Mach enthalpy method is validated against analytical solutions for a PCM undergoing a large density change during its phase transition. Additionally, a few simple sanity checks are proposed to benchmark computational fluid dynamics (CFD) algorithms that aim to capture the volume change effects of PCMs. Furthermore, we solve the enthalpy equation in conjunction with the Navier-Stokes equation to solve surface tension-driven flows. Additionally, we propose a robust technique for adaptively refining grids near the liquid-solid interface, enhancing the accuracy and efficiency of our simulations in critical regions.

4.1 Introduction

The numerical modeling and simulation of phase change materials (PCMs) is a very active area of research because they play a key role in energy systems (e.g., concentrated solar power plants and latent thermal energy storage units [20, 21, 22, 23, 24]), geophysical processes (e.g., sea ice formation and glacier melting [25, 26]), and manufacturing technologies (e.g., casting, welding, and metal 3D printing [27, 28, 29, 30]). The numerical modeling of PCMs is difficult because the energy equation is nonlinear, and most problems involve liquid flows, and some also involve gas flows and solid motion. As an example, consider selective laser melting (SLM) of metal powder as PCM for additively printing complex parts. During the SLM process, a thin powder layer of thickness 20–100 μm is deposited with the aid of a roller or a blade, and then fused by a directed laser source that typically scans at a rate of 0.1–1 m/s. After each laser pass over the powder bed, the solid powder particles melt and evaporate, and the molten metal pool solidifies to create the print [109]. The solidifying metal can entrain gas plumes from the surface, causing porosity and keyhole defects [110, 111, 112, 29] in the finished product. The aforementioned application of metal (powder) as a PCM illustrates the importance of resolving large density gradients (gas-metal density ratio is $\sim 10^4$) within multiphysics PCM simulations, which pose stability issues for numerical schemes [113, 114, 115].

Computational fluid dynamics (CFD) models for simulating the phase change of materials began to be developed in the late 80s and early 90s. According to the way they handle the moving phase boundary, these models can be categorized into two main groups: (**1**) deforming and (**2**) fixed grid schemes. Fixed grid schemes offer greater flexibility for incorporating additional physics into heat transfer problems (e.g., fluid flow) and are easier to implement than deforming grid schemes. They also naturally handle complex topological changes of the interface (merging, pinching, break-up, self-folding), which the deforming grid methods cannot. In the early development of fixed grid numerical methods, the volume change in PCM with melting or solidification was ignored and the variable density in the liquid was described by the Boussinesq approximation. This is still practiced today [31, 32, 33, 34, 35, 36, 37], despite significant improvements in numerical methodologies and the fact that solid–liquid phase changes are always accompanied by a volume change. Since most existing numerical methods can only deal with constant density PCMs, the gas phase is neglected and only melting and solidification are simulated. In the context of metal AM simulations, a recent review paper by Cook and Murphy [5] demonstrates that in most of the existing studies, the effect of the surrounding gas flow has been ignored. This is mainly due to the assumption that the powder or substrate cannot move. In reality, surrounding gas flows are prominent near the laser-interaction zone, where extreme thermo-capillary forces can drive gaseous pores out of the melt-pool and/or entrain particles in the melt-pool, resulting in deeper keyhole formation, which has been observed experimentally [116, 30].

Surface tension plays a crucial role in AM simulations, impacting the curvature of liquid interfaces, pore shape within the liquid, and melt pool behavior. Specifically, it contributes to phenomena such as the balling of the melt pool due to Plateau–Rayleigh instability and the movement/coalescence of partially melted powder due to capillary forces. The Marangoni effect, caused by temperature or species concentration variations affecting surface tension, is a significant driver of liquid flow dynamics in AM processes [5]. Surface tension is strongly influenced by temperature and concentration, and numerous applications involve temperature/concentration-dependent surface tension. For example, thermal fluctuations in the atomization of liquids in the combustion process significantly impact the liquid-gas interface [117]. For instance, thermal fluctuations during the atomization of liquids in combustion processes significantly affect the liquid-gas interface [117]. Similarly, using surfactants to reduce the surface tension of liquids is common in applications such as painting, lubrication, and pesticides [118]. Implementing surface tension in simulations faces challenges such as accurately defining surface normals and curvatures. Most studies employ the continuum surface tension model proposed by Brackbill et al. [119] to include this surface force as a volumetric force in the momentum equation. The temperature equation is solved in conjunction with the Navier-Stokes equation to simulate thermocapillary flows [120, 117, 121, 122, 123] with the assumption that surface tension σ is dependent only on temperature, i.e., $\sigma = f(T)$.

In this chapter, a fixed-grid low Mach enthalpy method is developed to capture density change-induced flow during PCM melting and solidification. In this formulation a gas phase is also incorporated and coupled to the solid-liquid PCM region. Our ultimate goal is to develop a simulation method that can handle simultaneous occurrences of evaporation, condensation, melting, and solidification. A method such as this would allow realistic modeling of manufacturing processes such as metal additive manufacturing, in which all four modes of phase change are present at the same time. The original enthalpy method (EM) introduced by Voller and colleagues [124, 125] for modeling melting and solidification of PCMs assumes the two material phases have the same density. Some recent works [126, 127, 128, 129] have relaxed the constant density requirement of the enthalpy method, but none to our knowledge have explicitly or carefully accounted for the volume change effect in this technique. This becomes even more critical for CFD models [130, 131, 31] that consider three phase gas-liquid-solid flows. Furthermore, existing numerical works have not verified whether using variable thermophysical properties (density, specific heat, thermal conductivity) in the enthalpy method produces an accurate numerical solution when such properties vary widely between phases[1]. The new low Mach EM is validated using the analytical solution to the two-phase Stefan problem for a PCM that undergoes a substantial volume change during solidification. The two-phase Stefan problem has a non-standard Stefan condition that involves additional jumps in the specific heat and kinetic energy. To derive analytical solutions, the kinetic energy jump term is dropped from the Stefan condition as it is usually small in comparison to latent heat. We discuss a way to retain it in the Stefan condition while still allowing for an analytical solution. Moreover, we use the low Mach enthalpy method to simulate thermocapillary flows. However, the significant difference between the previous and current studies is that the temperature is constructed from enthalpy rather than

---

[1] This is primarily due to the fact that in enthalpy methods, any thermophysical property is expressed as a function of a liquid fraction variable that also needs to be solved for. Variable thermophysical properties increase the nonlinearity of enthalpy methods.

obtained from the temperature equation. Still, the results are promising.

4.2 Jump conditions across the phase-changing interface

We follow Myers et al. [132] to derive the jump conditions across the phase-changing liquid-solid interface. The enthalpy equation and energy equation are distinguished at the end of this section. In addition, we will see how certain jump terms get omitted if the enthalpy equation is used as the starting point for the derivation. To derive the jump conditions, it is convenient to express the governing equations for mass balance, momentum, and energy in conservative form and apply the Rankine-Hugoniot condition across the interface. The conservation laws in differential form[2] read as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{4.1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\mathbb{I}) - \sigma \mathcal{C} \delta(\mathbf{x} - \mathbf{s})\mathbf{n} = 0 \tag{4.2}$$

$$\frac{\partial}{\partial t}\left(\rho\left[e + \frac{1}{2}|\mathbf{u}|^2\right]\right) + \nabla \cdot \left(\rho\left[e + \frac{1}{2}|\mathbf{u}|^2\right]\mathbf{u} + \mathbf{q} + p\mathbf{u}\right) - \sigma \mathcal{C} \delta(\mathbf{x} - \mathbf{s})\mathbf{u} \cdot \mathbf{n} = 0. \tag{4.3}$$

In momentum Eq. (4.2), σ is the surface tension coefficient between the two phases (liquid and solid in this context), $\mathcal{C}$ represents the mean local curvature of the interface, s represents the position of the interface, δ is the Dirac delta distribution, and $\mathbf{n}$ is the outward unit normal vector of the interface (pointing outwards from the solid and into the liquid phase in Fig. 4.1 (A)). In energy Eq. (4.3), e denotes the internal energy and $\mathbf{q} = -\kappa \nabla T$ is the conductive heat flux. Integrating Eqs. (4.1)-(4.3) across a finite region around the liquid-solid interface and then letting the volume of the region tend to zero, gives

1. Mass jump across the interface:

$$(\rho^{\mathrm{L}} - \rho^{\mathrm{S}})u^* = (\rho^{\mathrm{L}}\mathbf{u}^{\mathrm{L}} - \rho^{\mathrm{S}}\mathbf{u}^{\mathrm{S}}) \cdot \mathbf{n} = (\rho^{\mathrm{L}} u_{\mathrm{n}}^{\mathrm{L}} - \rho^{\mathrm{S}} u_{\mathrm{n}}^{\mathrm{S}}). \tag{4.4}$$

---

[2] Alternatively, one can start with the integral form of the conservation principle and derive the jump condition(s); see, for example, Delhaye [133].

Here, $u^* = \mathbf{u} \cdot \mathbf{n}$ represent the normal velocity of the interface, and $u_{\rm n}^{\rm L} = \mathbf{u}^{\rm L} \cdot \mathbf{n}$ and $u_{\rm n}^{\rm S} = \mathbf{u}^{\rm S} \cdot \mathbf{n}$ represent the near-interface normal component of velocity of the liquid and solid phase, respectively. The following relation can be obtained by taking the velocity in the solid to be zero $\mathbf{u}^{\rm S} = 0$

$$u_{\rm n}^{\rm L} = \left(1 - \frac{\rho^{\rm S}}{\rho^{\rm L}}\right) u^*. \tag{4.5}$$

2. Momentum jump across the interface:

$$(\rho^{\rm L}\mathbf{u}^{\rm L} - \rho^{\rm S}\mathbf{u}^{\rm S})u^* = \left(\rho^{\rm L}\mathbf{u}^{\rm L} \otimes \mathbf{u}^{\rm L} + p^{\rm L}\mathbb{I}\right) \cdot \mathbf{n} - \left(\rho^{\rm S}\mathbf{u}^{\rm S} \otimes \mathbf{u}^{\rm S} + p^{\rm S}\mathbb{I}\right) \cdot \mathbf{n} + \sigma\mathcal{C}\mathbf{n}. \tag{4.6}$$

The above vector equation can be expressed in terms of normal and tangential jumps. Taking an inner product of Eq. (4.6) with the unit normal vector $\mathbf{n}$ yields

$$\begin{aligned} (\rho^{\rm L}u_{\rm n}^{\rm L} - \rho^{\rm S}u_{\rm n}^{\rm S})u^* &= \rho^{\rm L}(u_{\rm n}^{\rm L})^2 - \rho^{\rm S}(u_{\rm n}^{\rm S})^2 + p^{\rm L} - p^{\rm S} + \sigma\mathcal{C} \\ \hookrightarrow\ p^{\rm S} - p^{\rm L} &= \rho^{\rm L}u_{\rm n}^{\rm L}(u_{\rm n}^{\rm L} - u^*) + \sigma\mathcal{C} = -\rho^{\rm S}u_{\rm n}^{\rm L}u^* + \sigma\mathcal{C}, \end{aligned} \tag{4.7}$$

in which we used Eq. (4.4) to express pressure jump in terms of $\rho^{\rm S}$. Likewise, jump in the tangential momentum across the interface is obtained by taking the inner product of Eq. (4.6) with the unit tangent vector $\mathbf{t}$

$$\begin{aligned} (\rho^{\rm L}u_{\rm t}^{\rm L} - \rho^{\rm S}u_{\rm t}^{\rm S})u^* &= \rho^{\rm L}u_{\rm n}^{\rm L}u_{\rm t}^{\rm L} - \rho^{\rm S}u_{\rm n}^{\rm S}u_{\rm t}^{\rm S} \\ \hookrightarrow\ (\rho^{\rm L} - \rho^{\rm S})u^* &= \rho^{\rm L}u_{\rm n}^{\rm L} - \rho^{\rm S}u_{\rm n}^{\rm S}. \end{aligned} \tag{4.8}$$

Here, $u_{\rm t}^{\rm L} = \mathbf{u}^{\rm L} \cdot \mathbf{t}$ and $u_{\rm t}^{\rm S} = \mathbf{u}^{\rm S} \cdot \mathbf{t}$ denote the tangential velocity component of the liquid and solid phase, respectively. In addition, we invoked the no-slip condition at the interface to equate the tangential velocities of the two phases, $u_{\rm t}^{\rm L} = u_{\rm t}^{\rm S}$. The no-slip assumption makes the mass and tangential momentum jumps equivalent.

3. Energy jump across the interface:

$$\rho^{\mathrm{L}}\left(e^{\mathrm{L}}+\frac{|\mathbf{u}^{\mathrm{L}}|^{2}}{2}\right)(u^{*}-u^{\mathrm{L}}_{\mathrm{n}})-\rho^{\mathrm{S}}\left(e^{\mathrm{S}}+\frac{|\mathbf{u}^{\mathrm{S}}|^{2}}{2}\right)(u^{*}-u^{\mathrm{S}}_{\mathrm{n}})=[(\mathbf{q}^{\mathrm{L}}-\mathbf{q}^{\mathrm{S}})$$
$$+(p^{\mathrm{L}}\mathbf{u}^{\mathrm{L}}-p^{\mathrm{S}}\mathbf{u}^{\mathrm{S}})]\cdot\mathbf{n}+\sigma\mathcal{C}u^{*}$$
$$\hookrightarrow\ \left[\rho^{\mathrm{S}}\left(e^{\mathrm{L}}-e^{\mathrm{S}}+\frac{|\mathbf{u}^{\mathrm{L}}|^{2}}{2}\right)-\sigma\mathcal{C}-\left(1-\frac{\rho^{\mathrm{S}}}{\rho^{\mathrm{L}}}\right)p^{\mathrm{L}}\right]u^{*}=[(\mathbf{q}^{\mathrm{L}}-\mathbf{q}^{\mathrm{S}})]\cdot\mathbf{n}, \tag{4.9}$$

in which Eqs. (4.4) and (4.5) have been used to simplify some terms.

The Stefan condition is obtained from the energy jump Eq. (4.9) by expressing internal energy in terms of temperature T and latent heat L

$$e^{\mathrm{S}}=C^{\mathrm{S}}(T^{\mathrm{S}}-T_{r})-\frac{p^{\mathrm{S}}}{\rho^{\mathrm{S}}}, \tag{4.10}$$
$$e^{\mathrm{L}}=C^{\mathrm{L}}(T^{\mathrm{L}}-T_{r})+L-\frac{p^{\mathrm{L}}}{\rho^{\mathrm{L}}}. \tag{4.11}$$

Substituting Eqs. (4.10) and (4.11) into Eq. (4.9) and using Eqs. (4.7) and (4.4) for further simplifications, yields the Stefan condition

$$\rho^{\mathrm{S}}\left[(C^{\mathrm{L}}-C^{\mathrm{S}})(T_{m}-T_{r})+L-\frac{1}{2}\left(1-\left(\frac{\rho^{\mathrm{S}}}{\rho^{\mathrm{L}}}\right)^{2}\right)(u^{*})^{2}\right]u^{*}=[\kappa^{\mathrm{S}}\nabla T^{\mathrm{S}}-\kappa^{\mathrm{L}}\nabla T^{\mathrm{L}}]\cdot\mathbf{n}. \tag{4.12}$$

Here, T_m denotes the melting/solidification temperature attained by the two phases at the interface and T_r is the phase change temperature in the bulk measured at a reference pressure.

Note that the Stefan condition (Eq. (4.12)) is derived from the energy jump Eq. (4.9), which considers the change in kinetic energy between liquid and solid across the interface. Several textbooks derive the Stefan condition from the enthalpy equation and do not include jumps related to kinetic energy. The reason for this can be traced back to the derivation of the enthalpy equation, which is determined by subtracting the kinetic energy equation from the energy equation. The subtraction eliminates terms like $\frac{1}{2}|\mathbf{u}|^2$. The kinetic energy equation is

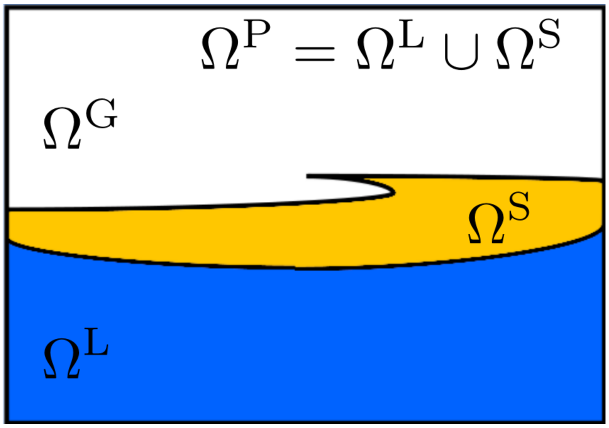


Figure 4.1. Schematics of the (A) two-phase and (B) three-phase problems examined in this study. A liquid phase is represented by blue, a solid phase by yellow, and a gas phase by white. Gas and PCM regions are tracked using the Heaviside function H, which is defined to be 1 in the PCM domain and 0 in the gas domain. Liquid and solid phases are tracked by the liquid fraction variable φ, which is equal to 1 in the liquid phase and 0 in the solid phase. A mushy region is defined by $0 < \varphi < 1$.

obtained by taking an inner product of velocity with the momentum equation and manipulating derivatives of velocity and pressure to obtain terms like $\nabla \cdot (\frac{1}{2}|\mathbf{u}|^2)$ and $\nabla \cdot (p\mathbf{u})$. It is not possible to manipulate derivatives for a phase change problem since both velocity and pressure are discontinuous across the interface.

4.3 Analytical solution to the two phase Stefan problem with density change

In this section, we revisit the two-phase Stefan problem involving a density jump from Alexiades and Solomon's textbook [134]. Unfortunately, despite its simplicity, this problem has not received much attention in the CFD literature. Meanwhile, the single phase Stefan problem, where only the heat equation is used and no density change-induced fluid flow is involved, remains the gold standard for validating advanced CFD algorithms for modeling melting and solidification [135, 130, 136] and boiling and condensation [137, 138] phenomena. In their textbook [134], the authors drop the kinetic energy jump term from the Stefan condition because it "destroys" the similarity solution. As we show in this section, this is not the case. A method for including it in the analytical solution is discussed. In the early stages of solidification, the interface velocity is infinite and the kinetic energy term dominates. Likewise, when a cylindrical or spherical material melts, its melting rate tends to infinity towards the end of the process. There are situations (certain time periods and length scales) where the kinetic energy term becomes important, and our analytical solution may prove useful in those instances.

Consider solidification of a liquid in a large static domain $\Omega := 0 \leq x \leq l$ that is closed on the left and open on the right. The same setup and analytical methodology can also be applied to the evaporation/condensation problem. Liquid occupies the whole domain at $t = 0$ and has a uniform temperature T_i greater than that of solidification temperature T_m. The temperature at the left boundary ($x = 0$) is suddenly lowered to $T_o < T_m$ at $t = 0^+$, which remains constant thereafter. The solidification front having position $x^* = s(t)$ moves in the positive x-direction as shown in Fig. 4.1(A). The thermophysical properties, i.e., density, specific heat, and thermal conductivity of the liquid phase are denoted ρ^{L}, C^{L}, and κ^{L}, respectively. The respective quantities for the solid phase are denoted ρ^{S}, C^{S}, and κ^{S}. Solid-to-liquid density ratio is denoted $R_\rho = \rho^{\mathrm{S}}/\rho^{\mathrm{L}}$. In the case where $R_\rho < 1$, the liquid expands as it solidifies, causing an additional flow in the direction of solidification. Alternatively, if $R_\rho > 1$, the fluid shrinks as it solidifies, causing a flow in the opposite direction. We seek analytical solutions for temperature, velocity, and pressure in the liquid and solid domains. Starting with the Rankine-Hugoniot relation for the mass balance equation (see Eq. (4.4)) the jump in velocity across the interface is obtained

$$[\![u]\!] = u^{\mathrm{L}}(s^+, t) - u^{\mathrm{S}}(s^-, t) = u^{\mathrm{L}}(s^+, t) = (1 - R_\rho)\frac{\mathrm{d}s}{\mathrm{d}t}. \tag{4.13}$$

Here, s^+ and s^- represent spatial locations just ahead and behind the interface, and $u^* = \mathrm{d}s/\mathrm{d}t$ represents interface speed. The velocity in the solid domain $\Omega^{\mathrm{S}}(t) := 0 \leq x < s(t)$ is taken to be zero[3], i.e., $u^{\mathrm{S}}(\Omega^{\mathrm{S}}, t) \equiv 0$, while in the liquid domain $\Omega^{\mathrm{L}}(t) := s(t) < x \leq l$, it is uniform

---

[3] In the case of evaporation and condensation, the solid phase will be replaced by gas/vapor in the analytical model. In the vapor phase, the velocity is also zero everywhere $u^{\mathrm{G}} \equiv 0$. It follows from the continuity equation $\partial u^{\mathrm{G}}/\partial x = 0$ and the zero-velocity boundary condition imposed at the left end ($x = 0$).

as per the continuity equation $\partial u^{\mathrm{L}}/\partial x = 0$. Therefore, liquid velocity can be obtained directly from interface speed as $u^{\mathrm{L}}(\Omega^{\mathrm{L}}, t) \equiv u^{\mathrm{L}}(s^{+}, t)$. Once the solid and fluid velocities have been determined, it is possible to solve the phase change problem with volume change effects by solving the energy equation for both solid and liquid phases

$$\rho^{\mathrm{S}} C^{\mathrm{S}} \left(\frac{\partial T^{\mathrm{S}}}{\partial t} + u^{\mathrm{S}} \frac{\partial T^{\mathrm{S}}}{\partial x} \right) = \kappa^{\mathrm{S}} \frac{\partial^2 T^{\mathrm{S}}}{\partial x^2} \ \in \Omega^{\mathrm{S}}(t), \tag{4.14}$$

$$\rho^{\mathrm{L}} C^{\mathrm{L}} \left(\frac{\partial T^{\mathrm{L}}}{\partial t} + u^{\mathrm{L}} \frac{\partial T^{\mathrm{L}}}{\partial x} \right) = \kappa^{\mathrm{L}} \frac{\partial^2 T^{\mathrm{L}}}{\partial x^2} \ \in \Omega^{\mathrm{L}}(t), \tag{4.15}$$

by taking $u^{\mathrm{S}}(x,t) \equiv 0$ and $u^{\mathrm{L}}(x,t) \equiv u^{\mathrm{L}}(s^{+},t)$. Five boundary conditions are required to determine $T^{\mathrm{S}}(x,t)$, $T^{\mathrm{L}}(x,t)$ and $x^{*} = s(t)$ completely. These include two boundary conditions $T^{\mathrm{S}}(0,t) = T_o$ and $T^{\mathrm{L}}(l,t) = T_i$ and three interfacial conditions:

$$T^{\mathrm{S}}(x^{*},t) = T^{\mathrm{L}}(x^{*},t) = T_m, \tag{4.16}$$

$$\begin{aligned} &\rho^{\mathrm{S}} \left[(C^{\mathrm{L}} - C^{\mathrm{S}})(T_m - T_r) + L - \frac{1}{2}(1 - R_{\rho}^2) \left(\frac{\mathrm{d}s}{\mathrm{d}t} \right)^2 \right] \frac{\mathrm{d}s}{\mathrm{d}t} \\ &= \left(\kappa^{\mathrm{S}} \frac{\partial T^{\mathrm{S}}}{\partial x} - \kappa^{\mathrm{L}} \frac{\partial T^{\mathrm{L}}}{\partial x} \right)_{x^{*}}. \end{aligned} \tag{4.17}$$

Here, T_r is the bulk phase change temperature measured at a specified reference pressure p_r, and L is the latent heat of melting/solidification. Eq. (6.24), which is commonly referred to as the Stefan condition, contains terms related to latent heat and jumps in the specific heat, density, and kinetic energy of the two material phases; see Sec. 4.2.

As shown in the Appendix Sec. 6.6, Eqs. (6.20) and (6.21) admit similarity solution of the form

$$T^{\mathrm{S}}(x,t) = T_o + A(\lambda(t))\,\mathrm{erf}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{S}}t}}\right), \tag{4.18}$$

$$T^{\mathrm{L}}(x,t) = T_i + B(\lambda(t))\,\mathrm{erfc}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{L}}t}} - \frac{s(t)}{2\sqrt{\alpha^{\mathrm{L}}t}}(1-R_\rho)\right), \tag{4.19}$$

in which $\alpha^{\mathrm{S}} = \kappa^{\mathrm{S}}/(\rho^{\mathrm{S}}C^{\mathrm{S}})$ and $\alpha^{\mathrm{L}} = \kappa^{\mathrm{L}}/(\rho^{\mathrm{L}}C^{\mathrm{L}})$ are the solid and liquid thermal diffusivities, respectively, and $\lambda(t) = \frac{s(t)}{2\sqrt{\alpha^{\mathrm{L}}t}}$ is a yet to be determined function of time. These temperature profiles satisfy the boundary and initial conditions of their respective equations. The unknown functions A and B appearing in the temperature profiles $T^{\mathrm{S}}(x,t)$ and $T^{\mathrm{L}}(x,t)$ implicitly depend on time through $\lambda(t)$. If $\lambda(t)$ is time-(in)dependent, so are A and B. To our knowledge, all textbooks and papers first assume that A and B are constants, and then deduce that λ has to be a time-independent constant. In contrast, we posit that A and B should actually be determined the other way around. To wit, the interface temperature condition written in Eq. (6.23)

$$T_o + A(\lambda(t))\,\mathrm{erf}\left(\frac{s(t)}{2\sqrt{\alpha^{\mathrm{S}}t}}\right) = T_i + B(\lambda(t))\,\mathrm{erfc}\left(\frac{s(t)}{2\sqrt{\alpha^{\mathrm{L}}t}}R_\rho\right) = T_m,$$

allows us to express A and B as a function of $\lambda(t)$:

$$A(\lambda(t)) = \frac{T_m - T_0}{\mathrm{erf}\left(\lambda(t)\sqrt{\frac{\alpha^{\mathrm{L}}}{\alpha^{\mathrm{S}}}}\right)} \qquad \text{and} \qquad B(\lambda(t)) = \frac{T_m - T_i}{\mathrm{erfc}\left(\lambda(t)R_\rho\right)}. \tag{4.20}$$

The temperature distribution in the solid phase is therefore

$$T^{\mathrm{S}} = T_o + (T_m - T_o)\mathrm{erf}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{S}}t}}\right) \Big/ \mathrm{erf}\left(\lambda(t)\sqrt{\frac{\alpha^{\mathrm{L}}}{\alpha^{\mathrm{S}}}}\right) \tag{4.21}$$

and in the liquid phase is

$$T^{\mathrm{L}} = T_i + (T_m - T_i)\frac{\mathrm{erfc}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{L}}t}} - \lambda(t)\left(1-R_\rho\right)\right)}{\mathrm{erfc}\left(\lambda(t)R_\rho\right)}. \tag{4.22}$$

By substituting Eqs. (6.28), (6.29), and $s(t) = 2\lambda(t)\sqrt{\alpha^{\mathrm{L}} t}$ into the Stefan condition (Eq. (6.24)), we obtain a governing equation for $\lambda(t)$ [4]

$$\rho^{\mathrm{S}}\left[L^{\mathrm{eff}} - \frac{(1-R_\rho^2)}{2}\left(\frac{\lambda^2\alpha^{\mathrm{L}}}{t}\right)\right]\lambda\sqrt{\alpha^{\mathrm{L}}} =$$
$$\kappa^{\mathrm{S}}\frac{T_m - T_o}{\text{erf}\left(\lambda\sqrt{\frac{\alpha^{\mathrm{L}}}{\alpha^{\mathrm{S}}}}\right)}\frac{e^{-\lambda^2\alpha^{\mathrm{L}}/\alpha^{\mathrm{S}}}}{\sqrt{\pi\alpha^{\mathrm{S}}}} + \kappa^{\mathrm{L}}\frac{T_m - T_i}{\text{erfc}\left(\lambda R_\rho\right)}\frac{e^{-\lambda^2 R_\rho^2}}{\sqrt{\pi\alpha^{\mathrm{L}}}} + \mathcal{O}\left(\left(\frac{\mathrm{d}\lambda}{\mathrm{dt}}\right)^3, ..\right), \tag{4.23}$$

in which $L^{\mathrm{eff}} = L + (C^{\mathrm{L}} - C^{\mathrm{S}})(T_m - T_r)$ is the effective latent heat. In the equation above $\mathcal{O}(...)$ contains terms related to the time derivatives of $\lambda(t)$, which arise from differentiating $s(t)$

$$\frac{\mathrm{d}s}{\mathrm{d}t} = \lambda\sqrt{\frac{\alpha^{\mathrm{L}}}{t}} + 2\sqrt{\alpha^{\mathrm{L}} t}\,\frac{\mathrm{d}\lambda}{\mathrm{d}t}. \tag{4.24}$$

At the beginning of the solidification process, the interface speed $\mathrm{ds}/\mathrm{dt} \to \infty$ due to the leading-order $1/\sqrt{t}$ term. By retaining only this term for $\frac{\mathrm{d}s}{\mathrm{d}t}$ in the Stefan condition, we obtain a time-dependent transcendental equation for $\lambda(t)$ instead of a differential one

$$\rho^{\mathrm{S}}\left[L^{\mathrm{eff}} - \frac{(1-R_\rho^2)}{2}\left(\frac{\lambda^2\alpha^{\mathrm{L}}}{t}\right)\right]\lambda\sqrt{\alpha^{\mathrm{L}}} =$$
$$\kappa^{\mathrm{S}}\frac{T_m - T_o}{\text{erf}\left(\lambda\sqrt{\frac{\alpha^{\mathrm{L}}}{\alpha^{\mathrm{S}}}}\right)}\frac{e^{-\lambda^2\alpha^{\mathrm{L}}/\alpha^{\mathrm{S}}}}{\sqrt{\pi\alpha^{\mathrm{S}}}} + \kappa^{\mathrm{L}}\frac{T_m - T_i}{\text{erfc}\left(\lambda R_\rho\right)}\frac{e^{-\lambda^2 R_\rho^2}}{\sqrt{\pi\alpha^{\mathrm{L}}}}. \tag{4.25}$$

Furthermore, we demonstrate that the $\mathrm{d}\lambda/\mathrm{dt}$ terms can be omitted from Eq. (4.25) through numerical experiments in Sec. 4.6.1.

From Eq. (4.25) (and also Eq. (6.30)), it is clear that if we keep the density/kinetic energy jump term in the Stefan condition, λ is an explicit function of time, so $A(\lambda)$ and $B(\lambda)$ are implicit

---

[4] Due to the absence of the solid phase at the beginning, $T^{\mathrm{S}}(x, t = 0)$ is not defined. Furthermore, since λ is derived using the solutions to $T^{\mathrm{S}}(x, t)$ and $T^{\mathrm{L}}(x, t)$, it is defined for $t > 0$. This can also be argued from the definition of interface position $s(t) = 2\lambda(t)\sqrt{\alpha^{\mathrm{L}} t}$, which does not exist at $t = 0$. On the other hand $T^{\mathrm{L}}(x, t = 0)$ is defined and is independent of λ thanks to the erfc(.) function in the numerator of Eq. (6.29), which approaches zero as $t \to 0$.

functions of time. By dropping the density/kinetic energy jump term from the Stefan condition, which is done in the literature, λ becomes independent of time. In this situation, $A(\lambda)$ and $B(\lambda)$ are constants.

The closed form solution for $T^{\mathrm{S}}(x,t)$, $T^{\mathrm{L}}(x,t)$, and $u^{\mathrm{L}}(x,t)$ can be written once $\lambda(t)$ has been found from Eq. (4.25) (without considering $\mathrm{d}\lambda/\mathrm{dt}$ terms). When $R_\rho = 1$ and $L^{\mathrm{eff}} = L$, the present analytical solution reduces to the standard Stefan problem solution given in the Hahn and Özişik textbook [139].

Next we find the variation of pressure inside the fluid and solid phases. First consider the fluid momentum equation

$$\rho^{\mathrm{L}}\left(\frac{\partial u^{\mathrm{L}}}{\partial t} + u^{\mathrm{L}}\frac{\partial u^{\mathrm{L}}}{\partial x}\right) = -\frac{\partial p^{\mathrm{L}}}{\partial x} + \mu^{\mathrm{L}}\frac{\partial^2 u^{\mathrm{L}}}{\partial x^2}, \tag{4.26}$$

which simplifies to

$$\rho^{\mathrm{L}}(1 - R_\rho)\frac{\mathrm{d}^2\mathrm{s}}{\mathrm{dt}^2} = -\frac{\partial p^{\mathrm{L}}}{\partial x} \tag{4.27}$$

when $u^{\mathrm{L}} \equiv (1 - R_\rho)\frac{\mathrm{ds}}{\mathrm{dt}}$ is substituted in Eq. (4.26). Integrating the above equation yields a linear variation of pressure within $\Omega^{\mathrm{L}}(t)$: $s(t) < x \leq l$ as

$$p^{\mathrm{L}}(x,t) = p_i - \frac{\lambda(t)}{2\sqrt{t^3}}(l - x)\left(\rho^{\mathrm{L}} - \rho^{\mathrm{S}}\right)\sqrt{\alpha^{\mathrm{L}}}. \tag{4.28}$$

Here, p_i is the pressure of the liquid phase at the right end ($x = l$). Due to the assumption that the liquid phase is incompressible, the domain length l needs to be finite. l, however, is large enough to accommodate the plateau region of the error function used in Eq. (6.26). In the solid phase, we obtain a uniform pressure by considering the momentum equation with $u^{\mathrm{S}} \equiv 0$

$$\begin{aligned} p^{\mathrm{S}}(x,t) &= p^{\mathrm{L}}(s^+,t) - [\![p]\!] \\ &= p_i - \frac{\lambda(t)}{2\sqrt{t^3}}(l - s(t))\left(\rho^{\mathrm{L}} - \rho^{\mathrm{S}}\right)\sqrt{\alpha^{\mathrm{L}}} - [\![p]\!], \end{aligned} \tag{4.29}$$

in which $[\![p]\!] = p^{\mathrm{L}}(s^+, t) - p^{\mathrm{S}}(s^-, t) = \rho^{\mathrm{S}}(1 - R_\rho)\left(\frac{\mathrm{ds}}{\mathrm{dt}}\right)^2$ is the jump in pressure across the interface; see Eq. (4.7).

This completes the analytical derivation of the Stefan problem involving jumps in liquid and solid thermophysical properties and we now proceed to the new low Mach enthalpy method formulation.

4.4 A low Mach enthalpy method

The fixed-grid CFD techniques for modeling phase change phenomena can be divided into two main categories: (**1**) sharp and (**2**) diffuse interface methods. A sharp technique treats the phase boundary as an infinitesimally thin surface, whereas a diffuse technique smears it over a few grid cells. Therefore, the former class of methods can explicitly impose the jump conditions of the governing equations within the solution methodology. Using a diffuse interface formulation, the phase transition occurs across a finite "mushy" region, and thus there is no jump in the governing equations as all quantities vary continuously. In spite of this, diffuse interface methods are quite popular in the literature due to their simplicity of implementation, robustness, and ability to handle more than two phases simultaneously. We consider the diffuse interface method for simulating solidification/melting of a PCM in this study. In addition, the proposed method allows us to couple a solid-liquid PCM with a passive gas phase without posing major problems.

Specifically, we consider the enthalpy method (EM) pioneered by Voller and colleagues [124, 125] for modeling melting and solidification of PCMs. Phase field methods (PFM) are another popular diffuse domain approach for modeling phase change phenomena [140, 135]. A major advantage of EM over PFM is that, unlike PFM, the EM does not require additional material parameters (such as mobility, Gibbs-Thompson, linear kinetic coefficients, mixing energy density, double-well potential function, etc.) that are usually empirically selected during numerical simulation. The solid-liquid interface is implicitly tracked in the EM using the liquid fraction variable $\varphi(\mathbf{x}, t)$ that is defined over the entire domain ($\mathbf{x}$ denotes a spatial location in Ω). φ is defined as 1 in the liquid phase, 0 in the solid phase, and between 0 and 1 in the transition/mushy zone. A temperature interval of $\Delta T = T^{\text{liq}} - T^{\text{sol}}$ (where T^{liq} represents the liquidus temperature at which solidification begins, and T^{sol} represents the solidus temperature at which full solidification occurs) is chosen to represent the range over which the phase change occurs. This assumption is based on the fact that for metal alloys and glassy substances there is no single melting temperature T_m because the phase change occurs over an extended range of temperatures from T^{sol} to T^{liq}, and there is a mushy zone between the all solid and all liquid regions [139]. The energy/enthalpy equation implicitly models phase change—upon changing phase, the grid cell's enthalpy adjusts to account for latent heat release or absorption, which in turn changes the φ value. EMs are typically implemented as source-based methods [125], where the material enthalpy is divided into sensible and latent components. Newton's method is used to solve the nonlinear energy equation containing latent heat as a source term. Many CFD softwares, including ANSYS Fluent and OpenFOAM, support source-based EM (SB-EM). Studies using SB-EM have often ignored the phase change induced fluid motion resulting from a density jump between liquid and solid phases. Where applicable, the Boussinesq approximation is used in the momentum equation to account for density variations within the liquid phase (e.g., to model natural convection in a melting PCM [141]).

Some authors have only recently begun considering solid and liquid densities differently when utilizing the EM method [126, 127, 128, 129]. To solve variable-density mass, momentum, and energy equations, these works have employed classical finite volume algorithms such as SIMPLE and PISO [142, 88]. These works suffer from the following shortcomings:

1. While the density field is defined by the liquid-fraction variable φ, i.e., an equation of state (EOS) is defined for the system, it is not explicitly used to constrain the velocity field. In other words, the continuous formulation does not explicitly distinguish between the bulk of phases with no change in material volume (where velocity is divergence-free)

and the narrow mushy region that allows changes in material volume (where velocity is not divergence-free). The previous formulations also do not guarantee that when all of the liquid has solidified, or when all of the solid has melted, there are no further volume changes in the system.

2. A temperature equation is derived from the enthalpy equation. It is done by expressing enthalpy as a function of temperature, for example, by using $h = C(T - T_r)$ type of relations. Here, $h = e + p/\rho$ denotes the specific enthalpy. However, this conversion has a disadvantage in that the specification of specific heat C, that depends on φ, becomes ambiguous. This is because φ evolves with T and it must also be solved for along with temperature. When solving for T and φ, C is usually held constant in the numerical implementation. As a result, the $h = C(T - T_r)$ relation can be satisfied only weakly.

3. The advection of temperature or enthalpy involves a mass flux term $\mathbf{m}_{\boldsymbol{\rho}} = \rho\mathbf{u}$. The discrete versions of mass and energy fluxes must be strongly coupled to each other to ensure the numerical stability of high density ratio flows. Prior works had not ensured this coupling.

4. Lastly, the prior numerical algorithms are only qualitatively validated against complex experiments [143, 144] (in complicated geometries and configurations). This includes qualitatively comparing the interface evolution between simulations and experiments [143, 126]. As a consequence, it is unclear how well the prior continuous and discrete formulations capture density-induced flows or volume changes in PCMs.

To overcome the aforementioned shortcomings in the prior works, we re-formulate the original enthalpy method as a low Mach technique. A low Mach approach has been traditionally applied to gas dynamics applications, like combustion [145, 146] and astrophysical flows [147, 148], however it can also be applied to fluid flow problems. Bell, Donev, and colleagues used a low Mach formulation to simulate multispecies liquid flows at mesoscales [149, 150, 151]. Our new low Mach EM formulation keeps the flow velocity divergence-free (div-free for short) in the bulk of solid and liquid phases. In the narrow mushy region between solid and liquid, where the material volume changes, the velocity is not divergence-free (non-div-free for short). The conditions on the velocity field are made explicit in both the continuous and discrete versions of our low Mach EM. A minor change in the EOS allows us to include a (passive) gas phase in the original solid-liquid PCM system. We assume that gas is incompressible and the formulation ensures that the gas domain's velocity is div-free. We solve the nonlinear enthalpy equation as it is without dividing it into sensible and latent components, i.e., we do not follow the source-based approach. This makes the method more general and allows us to incorporate the gas phase more easily. Furthermore, this avoids the ambiguity of defining specific heat in the domain when solving for enthalpy and liquid fraction. We also solve an additional mass balance equation to strongly couple mass advection with energy and momentum advection. For high density ratio flows, this step ensures that the numerical scheme remains stable. The new low Mach EM is validated using the analytical solution to the two-phase Stefan problem for a PCM that undergoes a substantial volume change during solidification. To illustrate the practical utility of our formulation, we simulate a metal casting problem showing a pipe defect [152] caused by the volume shrinkage of solidifying metal. Pipe defects are captured only when the velocity field is non-div-free, i.e., they cannot be captured by relying solely on a variable density field in the momentum and energy equations.

4.4.1 Mathematical formulation

In our technique the gas-PCM interface $\Gamma(t)$ is tracked using the zero-contour of the signed distance function $\phi(\mathbf{x}, t)$. ϕ is defined to be positive in the PCM region $\Omega^{\mathrm{P}}(t) = \Omega^{\mathrm{S}}(t) \cup \Omega^{\mathrm{L}}(t)$ and negative in the gas region $\Omega^{\mathrm{G}}(t)$. It is advected with the non-div-free velocity $\mathbf{u}(\mathbf{x}, t)$

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} = \frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0. \tag{4.30}$$

A smoothed Heaviside function $H(\mathbf{x}, t)$ is used in conjunction with the SDF ϕ to distinguish the gas and PCM regions: H takes a value of 0 in the gas region, 1 in the solid-liquid PCM region, and smoothly transitions from 0 to 1 around $\Gamma = \Omega^{\rm G} \cap \Omega^{\rm P}$ with a prescribed width of 2 grid cells on either side of the gas-PCM interface; see Fig. 4.1(B). Using the Heaviside function H and the liquid fraction variable φ any thermophysical property β (e.g., ρ, C, κ) can be uniquely defined throughout the domain

$$\beta = \beta^{\rm G} + (\beta^{\rm S} - \beta^{\rm G})H + (\beta^{\rm L} - \beta^{\rm S})H\varphi. \tag{4.31a}$$

$$\rho = \rho^{\rm G} + (\rho^{\rm S} - \rho^{\rm G})H + (\rho^{\rm L} - \rho^{\rm S})H\varphi. \tag{4.31b}$$

When $\beta = \rho$, we get the EOS written in Eq. (4.31b).

The EOS and the mass balance equation provide a kinematic constraint on the velocity field

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \\
\hookrightarrow &\frac{\partial \rho}{\partial t} + \rho \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla \rho = 0, \\
\hookrightarrow &\nabla \cdot \mathbf{u} = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho\right) = -\frac{1}{\rho}\frac{{\rm D}\rho}{{\rm D}t}.
\end{aligned} \tag{4.32}$$

The velocity divergence constraint can be expressed in terms of liquid fraction φ and Heaviside function H using the EOS (Eq. (4.31b)) as

$$\begin{aligned}
\nabla \cdot \mathbf{u} &= -\frac{1}{\rho}\frac{{\rm D}\rho}{{\rm D}t} \\
&= -\frac{1}{\rho}\left((\rho^{\rm S} - \rho^{\rm G})\frac{{\rm D}H}{{\rm D}t} + (\rho^{\rm L} - \rho^{\rm S})\frac{{\rm D}\,(H\varphi)}{{\rm D}t}\right) \\
&= -\frac{1}{\rho}\left((\rho^{\rm S} - \rho^{\rm G})\frac{{\rm D}H}{{\rm D}t} + (\rho^{\rm L} - \rho^{\rm S})\left(H\frac{{\rm D}\varphi}{{\rm D}t} + \varphi\frac{{\rm D}H}{{\rm D}t}\right)\right) \\
&= \frac{(\rho^{\rm S} - \rho^{\rm L})}{\rho}H\frac{{\rm D}\varphi}{{\rm D}t}.
\end{aligned} \tag{4.33}$$

In Eq. (4.33) we have used $\frac{\mathrm{D}H}{\mathrm{D}t} = 0$ as H follows the same linear advection equation (Eq. (4.30)) as ϕ. Having derived the low Mach Eq. (4.33) above, we now provide a physical rationale for the low Mach formulation of the enthalpy method.

Since solid and liquid phases are assumed to be incompressible, the characteristic sound speed is infinite in both solid and liquid regions. This means that in the bulk of both phases, the Mach number of the flow is zero. The mushy region between all solid and liquid phases is a very narrow area that is of the order of a few atomic/molecular diameters. Consequently, the characteristic sound speed in the mushy region is expected not to deviate significantly from the bulk solid and liquid phases, and it remains close to infinity. This *ansatz* allows us to employ a low Mach model to express density as a function of liquid fraction, which in turn is a function of enthalpy. A derivation of φ-h relation will be provided in this section. Low Mach models also imply that variations in density don't affect thermodynamic pressure $\tilde{p}$. Additionally, the pressure variable p which appears in the momentum equation is mechanical in origin. It serves as a Lagrange multiplier that enforces the kinematic constraint on the velocity field as written in Eq. (4.33). We remark that although we call the new EM a "low Mach" method, it is actually a zero Mach method. This is the common name for the class of models described by equations such as (4.31a) and (4.33). It is similar to how "low Reynolds number" is most commonly used to mean "zero Reynolds number."

The material derivative of the liquid fraction $\frac{\mathrm{D}\varphi}{\mathrm{D}t}$ required on the right-hand side of the low Mach Eq. (4.33) is obtained from the energy equation, which is written in terms of specific enthalpy h

$$\frac{\partial (\rho h)}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) = \rho \frac{\mathrm{D}h}{\mathrm{D}t} = \nabla \cdot \, (\kappa \nabla T) + Q_{\mathrm{src}}. \tag{4.34}$$

Here, Q_{src} represents any heat source/sink term, such as a scanning laser beam. Note that the enthalpy equation is obtained by subtracting the kinetic energy equation from the conservation of energy equation by manipulating derivatives associated with velocity $\mathbf{u}$ and pressure p. For a diffuse interface formulation this is acceptable because velocity/kinetic energy and pressure are assumed to be continuous across the interface. For a sharp interface approach this leads to the loss of kinetic energy jump terms; see Sec. 4.2 for further discussion on jump conditions.

Specific enthalpy h of the PCM is defined in terms of its temperature T as

$$h = \begin{cases} C^{\mathrm{S}}(T - T_r), & T < T^{\mathrm{sol}}, \\ \bar{C}(T - T^{\mathrm{sol}}) + h^{\mathrm{sol}} + \varphi \dfrac{\rho^{\mathrm{L}}}{\rho} L, & T^{\mathrm{sol}} \leq T \leq T^{\mathrm{liq}}, \\ C^{\mathrm{L}}(T - T^{\mathrm{liq}}) + h^{\mathrm{liq}}, & T > T^{\mathrm{liq}}, \end{cases} \tag{4.35}$$

and of the gas as

$$h = C^{\mathrm{G}}(T - T_r). \tag{4.36}$$

In Eq. (4.35), $h^{\mathrm{sol}} = C^{\mathrm{S}}(T^{\mathrm{sol}} - T_r), h^{\mathrm{liq}} = \bar{C}(T^{\mathrm{liq}} - T^{\mathrm{sol}}) + h^{\mathrm{sol}} + L,$ and $\bar{C} = \frac{C^{\mathrm{S}}+C^{\mathrm{L}}}{2}$. $\bar{C}$ is the specific heat of the mushy region, which is taken as an average of liquid and solid specific heats. Eqs. (4.35) and (4.36) imply that PCM and gas enthalpies are zero at $T = T_r$. The numerical solution is not affected by this arbitrary choice of reference temperature T_r, and in the numerical simulations we set the melting/solidification temperature as the reference temperature $T_r = T_m$ [5]. We use the mixture model [153] to express density and specific enthalpy in terms of liquid fraction in the mushy region

$$\rho = \varphi \rho^{\mathrm{L}} + (1 - \varphi)\rho^{\mathrm{S}}, \tag{4.37}$$

$$\rho h = \varphi \rho^{\mathrm{L}} h^{\mathrm{liq}} + (1 - \varphi)\rho^{\mathrm{S}} h^{\mathrm{sol}}. \tag{4.38}$$

Eqs. (4.37) and (4.38) can also be derived from the general Eq. (4.31a) by substituting $H = 1$ (which holds true in the PCM region) and $\beta = \rho$ or ρh.

Substituting h from Eq. (4.35) and ρ from Eq. (4.37) into Eq. (4.38), we obtain a φ-T

---

[5] Another reasonable choice is to set $T_r = 0$.

relation for the mushy region

$$\varphi = \frac{\rho}{\rho^{\mathrm{L}}} \frac{T - T^{\mathrm{sol}}}{T^{\mathrm{liq}} - T^{\mathrm{sol}}}. \tag{4.39}$$

Knowing φ in terms of T (Eq. (4.39)) allows us to invert h-T relations. The temperature in the PCM region

$$T = \begin{cases} \dfrac{h}{C^{\mathrm{S}}} + T_r, & h < h^{\mathrm{sol}}, \\ T^{\mathrm{sol}} + \dfrac{h - h^{\mathrm{sol}}}{h^{\mathrm{liq}} - h^{\mathrm{sol}}}(T^{\mathrm{liq}} - T^{\mathrm{sol}}), & h^{\mathrm{sol}} \leq h \leq h^{\mathrm{liq}}, \\ T^{\mathrm{liq}} + \dfrac{h - h^{\mathrm{liq}}}{C^{\mathrm{L}}}, & h > h^{\mathrm{liq}}, \end{cases} \tag{4.40}$$

and in the gas region

$$T = \frac{h}{C^{\mathrm{G}}} + T_r \tag{4.41}$$

can be written in terms of h. These T-h relations are used in the Newton's iterations to solve the nonlinear Eq. (4.34). Similarly, substituting ρ from Eq. (4.37) into Eq. (4.38), we get a φ-h relation

$$\varphi = \begin{cases} 0, & h < h^{\mathrm{sol}}, \\ \dfrac{\rho^{\mathrm{S}}(h^{\mathrm{sol}} - h)}{h(\rho^{\mathrm{L}} - \rho^{\mathrm{S}}) - \rho^{\mathrm{L}} h^{\mathrm{liq}} + \rho^{\mathrm{S}} h^{\mathrm{sol}}}, & h^{\mathrm{sol}} \leq h \leq h^{\mathrm{liq}}, \\ 1, & h > h^{\mathrm{liq}}. \end{cases} \tag{4.42}$$

Although arbitrary, φ in the gas region is defined to be zero.

Finally, Eq. (4.42) allows us to define $\frac{\mathrm{D}\varphi}{\mathrm{D}t}$ for the low Mach Eq. (4.33) as

$$\frac{\mathrm{D}\varphi}{\mathrm{D}t} = \begin{cases} 0, & h < h^{\mathrm{sol}}, \\ \dfrac{-\rho^{\mathrm{S}}\rho^{\mathrm{L}}(h^{\mathrm{sol}} - h^{\mathrm{liq}})}{(h(\rho^{\mathrm{L}} - \rho^{\mathrm{S}}) - \rho^{\mathrm{L}} h^{\mathrm{liq}} + \rho^{\mathrm{S}} h^{\mathrm{sol}})^2} \dfrac{\mathrm{D}h}{\mathrm{D}t}, & h^{\mathrm{sol}} \leq h \leq h^{\mathrm{liq}}, \\ 0, & h > h^{\mathrm{liq}}. \end{cases} \tag{4.43}$$

The material derivative of h in Eq. (4.43) is obtained from the enthalpy Eq. (4.34) as

$$\frac{\mathrm{D}h}{\mathrm{D}t} = \frac{1}{\rho}\left(\nabla \cdot\, (\kappa \nabla T) + Q_{\mathrm{src}}\right).$$

It is clear from Eq. (4.43) that $\frac{\mathrm{D}\varphi}{\mathrm{D}t} \neq 0$ only in the mushy region where $h^{\mathrm{sol}} \leq h \leq h^{\mathrm{liq}}$ and $T^{\mathrm{sol}} \leq T \leq T^{\mathrm{liq}}$. This results in a non-div-free velocity field in the mushy region, but div-free elsewhere. Therefore, in the absence of mushy regions, velocity is div-free. This can happen when a liquid phase has solidified completely or when a solid phase has melted completely. Our formulation, therefore, guarantees that there will be no change in the volume of the system in the absence of phase change. It can also be seen from Eq. (4.33) that when the densities of the solid and liquid phases match, there is no induced flow and the velocity is div-free.

In most of the problems considered in this work, the phase change of PCM is induced by an imposed temperature condition at the domain boundary. There are also cases where phase change is induced by a heat source, which is denoted by Q_{src} in Eq. (4.34). Physically, Q_{src} models melting of metals by lasers, electron beams, or electric arcs, which is common in metal additive manufacturing processes [5]. The dimension of Q_{src} is power (Watts) per unit volume. In general, laser power is expressed in Watts (or in terms of heat flux). Through the use of the delta function, an imposed heat flux (q'') be incorporated into the enthalpy equation as

$$Q_{\mathrm{src}} = q''\delta = q''\,|\nabla \widetilde{B}|. \tag{4.44}$$

As in Eqs. (4.47), we use a smooth delta function $\delta = |\nabla \widetilde{B}|$ that is obtained from a mollified Heaviside function $\widetilde{B}$ in Eq. (4.44).

The low Mach Eq. (4.33) is solved in conjunction with the momentum equation

$$\frac{\partial\,(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot\, \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right] + \rho \mathbf{g} - A_d \mathbf{u} + \mathbf{f}_{\mathrm{st}}, \tag{4.45}$$

to obtain the Eulerian velocity $\mathbf{u}(\mathbf{x}, t)$ and pressure $p(\mathbf{x}, t)$ in all three phases. Here, $\mu(\mathbf{x}, t)$ is the spatiotemporally varying viscosity that is defined using Eq. (4.31a), g is the acceleration due to gravity, and $A_d = C_d \frac{{\varphi_{\text{S}}}^2}{(1-\varphi_{\text{S}})^3 + \epsilon}$ is the Carman-Kozeny drag coefficient that is used to retard any flow in the solid domain, $\varphi_{\text{S}} = H(1-\varphi)$ is the solid fraction of the grid cell, and $\epsilon = 10^{-3}$ is a small number to avoid a division by zero and to control the strength of penalty parameter (C_d/ϵ) in the solid region. To retard fluid motion within the solid domain, the model parameter C_d takes a large value. By comparing the magnitudes of the drag force and the first term on the left hand side of the momentum equation (i.e., equating intertial force to drag force), we obtain a sufficiently large value for $C_d = \rho^{\text{S}}/\Delta t$. Here, Δt denotes the time step size of the simulation. $\mathbf{f}_{\text{st}}$ is the surface tension force that acts on the liquid-gas interface. The next section details the numerical algorithm and the time stepping scheme. Due to the large density difference between the solid, liquid, and gas phases, special care is needed to avoid numerical instabilities. This is explained in the next section as well. Observe that the momentum equation is expressed using a diffuse interface formulation, where all quantities are assumed to vary continuously across the (three) phases. In addition, the Carman-Kozeny drag force strongly influences the pressure jump/gradient across the mushy region, which is similar to the Darcy-Brinkman equation for modeling flows in porous media [154]. When a diffuse interface formulation is used for the momentum equation, p and $[\![p]\!]$ will generally have numerical values that differ (perhaps by orders of magnitude) from those of a sharp interface formulation. This is discussed further in the context of the Stefan problem in the next section.

A continuum surface tension model, proposed by Brackbill et al. [119] is used to account for surface tension $\mathbf{f}_{\text{st}}$ along the liquid-gas interface. The surface tension forces in flows with large gradients of temperature and concentration are nonuniform. Temperature-dependent surface tension causes additional thermocapillary flows. We assume that the surface tension coefficients σ are temperature-dependent in this work. This assumption leads to a continuum surface tension force of the form

$$\mathbf{f}_{\text{st}} = \sigma(T)\mathcal{C}(\phi)\delta\mathbf{n} + \nabla_{||}\sigma(T)\,\delta, \tag{4.46a}$$

$$\hookrightarrow \mathbf{f}_{\text{st}} = \sigma(T)\mathcal{C}(\phi)\delta\mathbf{n} + \left[\nabla\sigma(T) - \left(\nabla\sigma(T)\cdot\mathbf{n}\right)\mathbf{n}\right]\,\delta. \tag{4.46b}$$

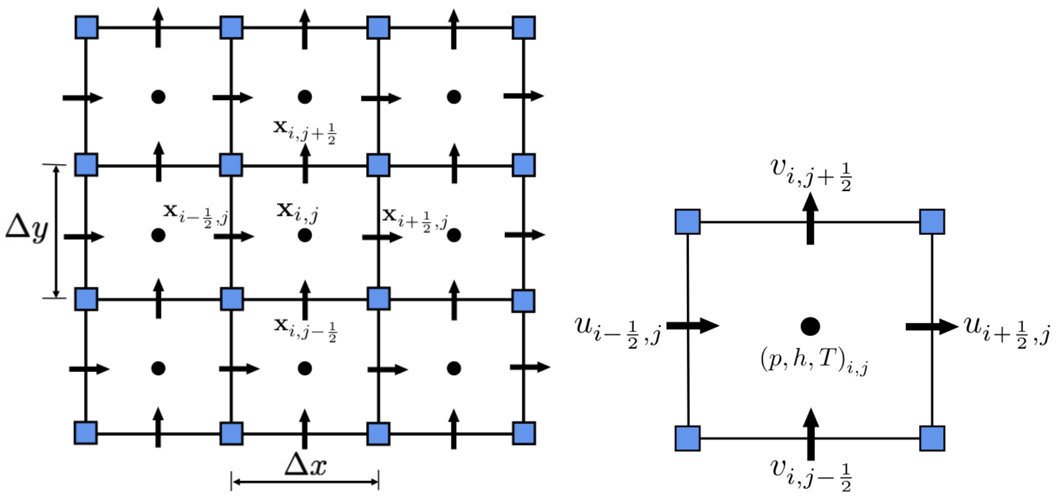
Here, $\mathcal{C}(\phi) = -\nabla\cdot\mathbf{n} = -\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ is the curvature of the interface computed from the level set function ϕ, and δ is the smoothed/mollified delta function. Implementing Eq. (4.46b) as is leads to spurious/parasitic velocity currents near the interface. Francois et al. [] proposed a well-balanced scheme wherein δ and $\delta\,\mathbf{n}$ are obtained from the discrete gradient of a mollified Heaviside function $\widetilde{B}$ as $\delta = |\nabla\widetilde{B}|$ and $\delta\,\mathbf{n} = \nabla\widetilde{B}$. Furthermore, the spatial gradient of the surface tension coefficient can be expressed as $\nabla\sigma = \frac{\mathrm{d}\sigma}{\mathrm{dT}}\nabla T$. With these substitutions, Eq. (4.46b) becomes

$$\mathbf{f}_{\text{st}} = \sigma(T)\mathcal{C}(\phi)\nabla\widetilde{B} + \nabla\sigma(T)|\nabla\widetilde{B}| - \left(\nabla\sigma(T)\cdot\frac{\nabla\phi}{|\nabla\phi|}\right)\nabla\widetilde{B}, \tag{4.47a}$$

$$\hookrightarrow \mathbf{f}_{\text{st}} = \sigma(T)\mathcal{C}(\phi)\nabla\widetilde{B} + \frac{\mathrm{d}\sigma}{\mathrm{dT}}\nabla T|\nabla\widetilde{B}| - \left(\nabla T\cdot\frac{\nabla\phi}{|\nabla\phi|}\right)\frac{\mathrm{d}\sigma}{\mathrm{dT}}\nabla\widetilde{B} \tag{4.47b}$$

$\frac{\mathrm{d}\sigma}{\mathrm{dT}}$ is commonly referred to as the Marangoni surface tension coefficient.

In our numerical simulations we use enthapy Eq. (4.34) in conjunction with surface tension Eq. (4.47b) to simulate thermocapillary flows. Most thermocapillary flow models in the literature use temperature equation rather than enthalpy. Using enthalpy equation instead of temperature does not make any difference at a continuous level, but discretely the results might differ. Our results for simulating thermocapillary flows match benchmark solutions reported in the literature, thanks to consistent mass-momentum-enthalpy integrators employed in this work.



(A) A 2D staggered Cartesian grid

(B) A single Cartesian grid cell

Figure 4.2. An illustration of a 2D staggered Cartesian grid. (A) shows the coordinate system for the staggered grid. (B) shows a single grid cell with velocity components u and v approximated at the cell faces ($\rightarrow$) and scalar variables such as pressure p, specific enthalpy h and temperature T approximated at the cell center ($\bullet$).

4.4.2 Discretization

Spatial discretization

The continuous equations of motion are discretized on a locally refined staggered Cartesian grid. In this section, we explain the discretization in two spatial dimensions at the coarsest/single level without sacrificing generality. Locally refined grids are addressed in Sec. 4.5.

The coarsest grid level covers the entire domain Ω with $N_x \times N_y$ rectangular cells. The cell size in x and y directions is Δx and Δy, respectively, as illustrated in Fig. 4.2(A). Unless stated otherwise, a uniform grid spacing $\Delta x = \Delta y$ is used for all simulations in this work. We assume that the bottom-left corner of the computational domain Ω aligns with the origin $(0, 0)$. The position of each grid cell center is given by $\mathbf{x}_{i,j} = \big((i + \frac{1}{2})\Delta x, (j + \frac{1}{2})\Delta y\big)$, where $i = 0, \ldots, N_x - 1$ and $j = 0, \ldots, N_y - 1$. The face center in the $x-$direction, which is located half a grid space away from the cell center $\mathbf{x}_{i,j}$ in the negative $x-$direction, is represented by $\mathbf{x}_{i-\frac{1}{2},j} = \big(i\Delta x, (j + \frac{1}{2})\Delta y\big)$, where $i = 0, \ldots, N_x$ and $j = 0, \ldots, N_y - 1$. Similar conventions apply to other face center locations. Scalar variables such as pressure (p), specific enthalpy (h), and temperature (T) are stored at the cell centers. The $x-$component of velocity (u) is stored at the $x-$direction cell faces, while the $y-$component of velocity v is stored at the $y-$direction cell faces, as depicted in Fig. 4.2(B). Material properties, including density (ρ), are stored at the cell centers along with interface tracking variables ϕ and φ. Second-order interpolation is used to interpolate the cell-centered quantities to the faces. Momentum forcing terms, such as surface tension, gravity, and Carman-Kozeny drag force, are stored at the cell faces as velocity components. Standard second-order finite differences are employed to approximate the spatial differential operators. The spatial discretizations of the key continuous operators are as follows:

- The divergence of the velocity field $\mathbf{u} = (u, v)$ is approximated at cell centers by

$$\mathbf{D} \cdot \mathbf{u} = D^x u + D^y v, \tag{4.48a}$$

$$(D^x u)_{i,j} = \frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x}, \tag{4.48b}$$

$$(D^y v)_{i,j} = \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y}. \tag{4.48c}$$

- The diffusion term in the energy equation is approximated as

$$\begin{aligned}\mathbf{D} \cdot (\kappa \nabla T) =& \frac{1}{\Delta x}\left(\kappa_{i+\frac{1}{2},j}\frac{T_{i+1,j} - T_{i,j}}{\Delta x} - \kappa_{i-\frac{1}{2},j}\frac{T_{i,j} - T_{i-1,j}}{\Delta x}\right) \\ &+ \frac{1}{\Delta y}\left(\kappa_{i,j+\frac{1}{2}}\frac{T_{i,j+1} - T_{i,j}}{\Delta y} - \kappa_{i,j-\frac{1}{2}}\frac{T_{i,j} - T_{i,j-1}}{\Delta y}\right)\end{aligned} \tag{4.49}$$

- The gradient of cell-centered quantities (i.e., p) is approximated at cell faces by

$$\mathbf{G}p = (G^x p, G^y p), \tag{4.50a}$$

$$(G^x p)_{i-\frac{1}{2},j} = \frac{p_{i,j} - p_{i-1,j}}{\Delta x}, \tag{4.50b}$$

$$(G^y v)_{i,j-\frac{1}{2}} = \frac{p_{i,j} - p_{i,j-1}}{\Delta y}. \tag{4.50c}$$

- The continuous strain rate tensor form of the viscous term is

$$\nabla\cdot\left[\mu\left(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathsf{T}}\right)\right]=\begin{bmatrix} 2\frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}+\mu\frac{\partial v}{\partial x}\right) \\ 2\frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right)+\frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}+\mu\frac{\partial u}{\partial y}\right)\end{bmatrix}, \tag{4.51}$$

which couples the velocity components through spatially variable viscosity

$$\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u}=\begin{bmatrix}(\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u})^{x}_{i-\frac{1}{2},j} \\ (\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u})^{y}_{i,j-\frac{1}{2}}\end{bmatrix}. \tag{4.52}$$

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

$$\begin{aligned}
(\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u})^{x}_{i-\frac{1}{2},j} &= \frac{2}{\Delta x}\left[\mu_{i,j}\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x}-\mu_{i-1,j}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{3}{2},j}}{\Delta x}\right] \\
&+\frac{1}{\Delta y}\left[\mu_{i-\frac{1}{2},j+\frac{1}{2}}\frac{u_{i-\frac{1}{2},j+1}-u_{i-\frac{1}{2},j}}{\Delta y}-\mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{1}{2},j-1}}{\Delta y}\right] \\
&+\frac{1}{\Delta y}\left[\mu_{i-\frac{1}{2},j+\frac{1}{2}}\frac{v_{i,j+\frac{1}{2}}-v_{i-1,j+\frac{1}{2}}}{\Delta x}-\mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{v_{i,j-\frac{1}{2}}-v_{i-1,j-\frac{1}{2}}}{\Delta x}\right] && \text{(4.53a)} \\
(\mathbf{L}_{\boldsymbol{\mu}}\mathbf{u})^{y}_{i,j-\frac{1}{2}} &= \frac{2}{\Delta y}\left[\mu_{i,j}\frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta y}-\mu_{i,j-1}\frac{v_{i,j-\frac{1}{2}}-v_{i,j-\frac{3}{2}}}{\Delta y}\right] \\
&+\frac{1}{\Delta x}\left[\mu_{i+\frac{1}{2},j-\frac{1}{2}}\frac{v_{i+1,j-\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta x}-\mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{v_{i,j-\frac{1}{2}}-v_{i-1,j-\frac{1}{2}}}{\Delta x}\right] \\
&+\frac{1}{\Delta x}\left[\mu_{i+\frac{1}{2},j-\frac{1}{2}}\frac{u_{i+\frac{1}{2},j}-u_{i+\frac{1}{2},j-1}}{\Delta y}-\mu_{i-\frac{1}{2},j-\frac{1}{2}}\frac{u_{i-\frac{1}{2},j}-u_{i-\frac{1}{2},j-1}}{\Delta y}\right], && \text{(4.53b)}
\end{aligned}$$

in which viscosity is required at both cell centers and nodes of the staggered grid (i.e., $\mu_{i\pm\frac{1}{2},j\pm\frac{1}{2}}$). Node-centered quantities are obtained via interpolation by either arithmetically or harmonically averaging the neighboring cell-centered quantities. In this work, we utilize harmonic averaging.

In terms of convective discretization, we employ the third-order accurate cubic upwind interpolation (CUI) scheme. The CUI method satisfies both the convection-boundedness criterion

(CBC) and the total variation diminishing (TVD) property. Specifically, the CUI scheme demonstrates third-order spatial accuracy in monotonic regions, where the gradient of the advected quantity remains monotone, and transitions to first-order spatial accuracy in non-monotonic regions due to upwinding. We do not present the spatial discretization of the advection equation using CUI here for brevity, but detailed information can be found in our prior publication [113]. Extending these discretizations to three-dimensional Cartesian grids is straightforward. For further details on the spatial discretization and boundary conditions on adaptively refined meshes, refer our prior works [113, 4, 42].

4.4.3 Temporal discretization

In this section, we detail the temporal discretization method utilized in our study for the continuous equation of motion. Within each time step $[t^n, t^{n+1}]$, we employ n_{cycles} cycles of fixed-point iteration to approximate the solution to the coupled fluid-thermal system. At the start of the simulation ($t = 0$), all variables are initialized with the initial conditions of the problem. At the beginning of the each time step, with cycle number $k = 0$, variables are initialized to their values from the previous time step, denoted as $\mathbf{u}^{n+1,0} = \mathbf{u}^n$, $p^{n+\frac{1}{2},0} = p^{n-\frac{1}{2}}$, $h^{n+1,0} = h^n$, $\phi^{n+1,0} = \phi^n$, and $\varphi^{n+1,0} = \varphi^n$, and iterated till $k = n_{\text{cycles}} - 1$. Convective and surface tensions terms are treated explicitly, while other terms are treated implicitly. Unless otherwise specified, $n_{\text{cycles}} = 2$ is used in this work.

Consistent mass-momentum integrators

For incompressible multiphase flows, mass is advected indirectly via an interface tracking variable. In the present work this corresponds to the advection of the level set variable ϕ. The momentum equation also contains a mass flux term $\mathbf{m}_{\boldsymbol{\rho}} \equiv \rho\mathbf{u}$ in the the convective operator $\nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = \nabla \cdot (\mathbf{m}_{\boldsymbol{\rho}} \otimes \mathbf{u})$. For high density ratio flows $\rho^{\text{L}}/\rho^{\text{G}} > 100$ this weak coupling of mass and momentum leads to numerical instabilities. One way to ensure strong coupling between mass and momentum advection is to solve the redundant mass balance Eq. (4.32) and use the same mass flux $\mathbf{m}_{\boldsymbol{\rho}}$ in the two advective operators: $\nabla \cdot (\rho \mathbf{u})$ and $\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u})$. This idea was proposed in our prior works [113, 4].

Although the use of the same discrete mass flux is essential for ensuring stability of high density ratio flows, there is still some freedom in the choice of time integrators for mass and momentum equations. In our prior works we employed the strong stability preserving Runga-Kutta (SSP-RK3) scheme for solving the mass balance equation and a midpoint RK-2 scheme for the momentum equation. Specifically the mass balance equation is integrated in three stages:

$$\breve{\boldsymbol{\rho}}^{(1)} = \breve{\boldsymbol{\rho}}^{n} - \Delta t \mathbf{R}\left(\mathbf{u}_{\text{adv}}^{n}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{n}\right), \tag{4.54a}$$

$$\breve{\boldsymbol{\rho}}^{(2)} = \frac{3}{4}\breve{\boldsymbol{\rho}}^{n} + \frac{1}{4}\breve{\boldsymbol{\rho}}^{1} - \frac{1}{4}\Delta t \mathbf{R}\left(\mathbf{u}_{\text{adv}}^{(1)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)}\right), \tag{4.54b}$$

$$\breve{\boldsymbol{\rho}}^{n+1,k+1} = \frac{1}{3}\breve{\boldsymbol{\rho}}^{n} + \frac{2}{3}\breve{\boldsymbol{\rho}}^{(2)} - \frac{2}{3}\Delta t \mathbf{R}\left(\mathbf{u}_{\text{adv}}^{(2)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}\right), \tag{4.54c}$$

in which $\breve{\boldsymbol{\rho}}$ denotes the face-centered density and $\mathbf{u}_{\text{adv}}$ represents the advection velocity that is centered at the faces of the staggered (velocity) control volume. The right-hand side term in Eqs. (4.54) $\mathbf{R}\left(\mathbf{u}_{\text{adv}}, \breve{\boldsymbol{\rho}}_{\text{lim}}\right) \approx \left[\left(\nabla \cdot \left(\mathbf{u}_{\text{adv}}\breve{\boldsymbol{\rho}}_{\text{lim}}\right)\right)_{i-\frac{1}{2},j}, \left(\nabla \cdot \left(\mathbf{u}_{\text{adv}}\breve{\boldsymbol{\rho}}_{\text{lim}}\right)\right)_{i,j-\frac{1}{2}}\right]$ is an explicit CUI limited approximation to the linear advection term $\nabla \cdot (\rho \mathbf{u})$. Subscripts "lim" and "adv" indicate a limited and advected quantity, respectively. To maintain the accuracy of the SSP-RK3 integrator, it is essential to use velocities that are appropriately extrapolated or interpolated in time. To wit, for the first cycle ($k = 0$), the velocities are given by

$$\mathbf{u}^{(1)} = 2\mathbf{u}^{n} - 2\mathbf{u}^{n-1}, \tag{4.55a}$$

$$\mathbf{u}^{(2)} = \frac{3}{2}\mathbf{u}^{n} - \frac{1}{2}\mathbf{u}^{1}, \tag{4.55b}$$

and for the remaining cycles ($k > 0$), the velocities are

$$\mathbf{u}^{(1)} = \mathbf{u}^{n+1,k}, \tag{4.56a}$$

$$\mathbf{u}^{(2)} = \frac{3}{8}\mathbf{u}^{n+1,k} + \frac{3}{4}\mathbf{u}^{n} - \frac{1}{8}\mathbf{u}^{n-1}. \tag{4.56b}$$

The consistency between mass and momentum advection is achieved by using the same discrete mass flux $\mathbf{m}_{\boldsymbol{\rho}} = \breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}\mathbf{u}_{\text{adv}}^{(2)}$ from the last stage of the SSP-RK3 integrator (Eq. $4.54c$) in the discrete momentum equation

$$\begin{aligned}\frac{\breve{\boldsymbol{\rho}}^{n+1,k+1}\mathbf{u}^{n+1,k+1} - \breve{\boldsymbol{\rho}}^{n}\mathbf{u}^{n}}{\Delta t} + \mathbf{C}\left(\mathbf{u}_{\text{adv}}^{(2)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}\mathbf{u}_{\text{lim}}^{(2)}\right) &= -\mathbf{G}p^{n+\frac{1}{2},k+1} + \left(\mathbf{L}_{\mu}\mathbf{u}\right)^{n+\frac{1}{2},k+1} \\ &\quad + \breve{\boldsymbol{\rho}}^{n+1,k+1}\mathbf{g} - \left(A_d\mathbf{u}\right)^{n+1,k+1} + \mathbf{f}_{\text{st}}^{n+\frac{1}{2},k+1}.\end{aligned} \tag{4.57}$$

in which the convective term is approximated as

$$\mathbf{C}\left(\mathbf{u}_{\text{adv}}^{(2)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}\mathbf{u}_{\text{lim}}^{(2)}\right) \approx \left[\left(\nabla \cdot \left(\mathbf{u}_{\text{adv}}^{(2)}\breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}u_{\text{lim}}^{(2)}\right)\right)_{i-\frac{1}{2},j}, \left(\nabla \cdot \left(\mathbf{u}_{\text{adv}}^{(2)}\breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}v_{\text{lim}}^{(2)}\right)\right)_{i,j-\frac{1}{2}}\right], \tag{4.58}$$

and $\breve{\boldsymbol{\rho}}^{n+1,k+1}$ is the updated value of the density obtained from the last stage of the SSP-RK3 integrator. The term $A_d = C_d\dfrac{\varphi_{\text{S}}^2}{(1-\varphi_{\text{S}})^2 + \epsilon}$ in the momentum equation represents the Carman-Kozeny drag coefficient that is used to retard any flow in the solid domain. Here, $\varphi_{\text{S}} = H(1-\varphi)$ is the solid fraction of the grid cell, and $\epsilon = 10^{-3}$ is a small number to avoid a division by zero and to control the strength of penalty parameter (C_d/ϵ) in the solid region. In the solid domain, C_d is also taken to be large in magnitude to retard fluid motion. When equating drag force with the first term on the left hand side of the momentum equation, a sufficiently large value is obtained for $C_d = \rho^{\text{S}}/\Delta t$.

In our prior works, for the test problems considered, it was demonstrated that although the mass and momentum equations employed different time integrators, the use of the same discrete mass flux was sufficient to ensure numerical stability. In this work, we propose an additional stabilizing term in the momentum equation to account for different integration schemes used for mass and momentum equations. Specifically, after integrating the mass balance equation (using SSP-RK3), the (face-centered) residual of the equation is computed

$$\boldsymbol{\mathcal{R}} = \frac{\breve{\boldsymbol{\rho}}^{n+1,k+1} - \breve{\boldsymbol{\rho}}^{n}}{\Delta t} + \mathbf{R}\left(\mathbf{u}_{\text{adv}}^{(2)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}\right). \tag{4.59}$$

The residual $\boldsymbol{\mathcal{R}}$ is expected to be very small (close to zero). The residual of the mass equation is used to define a body force term in the momentum equation. The modified momentum equation reads as

$$\begin{aligned}\frac{\breve{\boldsymbol{\rho}}^{n+1,k+1}\mathbf{u}^{n+1,k+1} - \breve{\boldsymbol{\rho}}^{n}\mathbf{u}^{n}}{\Delta t} + \mathbf{C}\left(\mathbf{u}_{\text{adv}}^{(2)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(2)}\mathbf{u}_{\text{lim}}^{(2)}\right) &= \boldsymbol{\mathcal{R}}\mathbf{u}^{n+1,k} \\ &+ \text{viscous + pressure + other forces.}\end{aligned} \tag{4.60}$$

The rationale for adding $\boldsymbol{\mathcal{R}}\mathbf{u}^{n+1,k}$ to the right hand side of the momentum equation is as follows. Consider a uniform and constant velocity ($\mathbf{u} \equiv \mathbf{u}_c$) flow which is not subject to any viscous, pressure or body force. In such a case the momentum equation reverts to the mass balance equation:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = 0 \tag{4.61a}$$

$$\hookrightarrow \mathbf{u}_c\left(\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}_c)\right) = 0, \tag{4.61b}$$

$$\hookrightarrow \mathbf{u}_c\left(\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u})\right) = \boldsymbol{\mathcal{R}}\mathbf{u}_c = 0. \tag{4.61c}$$

Thus the addition of the close-to-zero term $\boldsymbol{\mathcal{R}}\mathbf{u}^{n+1,k}$ ensures that the left and right hand sides of the modified momentum Eq. (4.60) discretely balance each other.

Another way to ensure consistency between mass and momentum transport is to employ the same integration scheme for the two equations. Since our existing fluid solvers employ a midpoint/RK-2 scheme for integrating the momentum equation, we choose to employ the same integration scheme for the mass equation. In this case, the density update reads as:

$$\breve{\boldsymbol{\rho}}^{n+1,k+1} = \breve{\boldsymbol{\rho}}^{n} - \Delta t \mathbf{R}\left(\mathbf{u}_{\text{adv}}^{(1)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)}\right), \tag{4.62}$$

in which $\mathbf{u}_{\text{adv}}^{(1)} = \frac{1}{2}\left(\mathbf{u}_{\text{adv}}^{n} + \mathbf{u}_{\text{adv}}^{n+1,k}\right)$ and $\breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)} = \frac{1}{2}\left(\breve{\boldsymbol{\rho}}_{\text{lim}}^{n} + \breve{\boldsymbol{\rho}}_{\text{lim}}^{n+1,k}\right)$. It can be noted that the above scheme reduces to forward Euler (RK1) scheme with $n_{\text{cycles}} = 1$. With the same integration scheme for mass and momentum equations (RK-2) it is not necessary to add $\boldsymbol{\mathcal{R}}\mathbf{u}^{n+1,k}$ to the right hand side of the momentum equation. Nevertheless, in our codes we retain this term for consistency purposes.

A test problem to check consistency of mass-momentum transport: To illustrate the importance of consistent mass and momentum transport for high density ratio flows, consider an advection of a dense, isothermal bubble moving in a uniform velocity $\mathbf{u}_c = [u_c, v_c] = [1, 1]$ in a square periodic domain $\Omega \in [0, 1]^2$. The radius of the bubble is $R = 0.2$ and it is initially centered at the middle of the domain $[x_i = 0.5, y_i = 0.5]$. The density ratio between the bubble and the outer liquid is $\rho_i/\rho_o = 10,000$. The domain is discretized with 2 levels of mesh, with the coarsest mesh size of $N_x \times N_y = 128^2$. A refinement ratio of $n_{\text{ref}} = 2$ is used between the two levels. The bubble's interface is always placed on the finest level. Further details on the adaptive mesh refinement (AMR) are provided in the next section. Viscosity μ is set to zero in the entire domain, and no other body force (e.g., gravity or surface tension) is present. Under these conditions, the governing equations for this problem reduce to

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{4.63a}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{4.63b}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = 0. \tag{4.63c}$$

For this problem it is expected that the bubble advects with a constant velocity $\mathbf{u}_c$ and returns to its original position (with minimal distortions) after $t = d/|\mathbf{u}_c|$, in which d represents the distance travelled to reach its original position. We simulate the advection of the bubble considering four scenarios

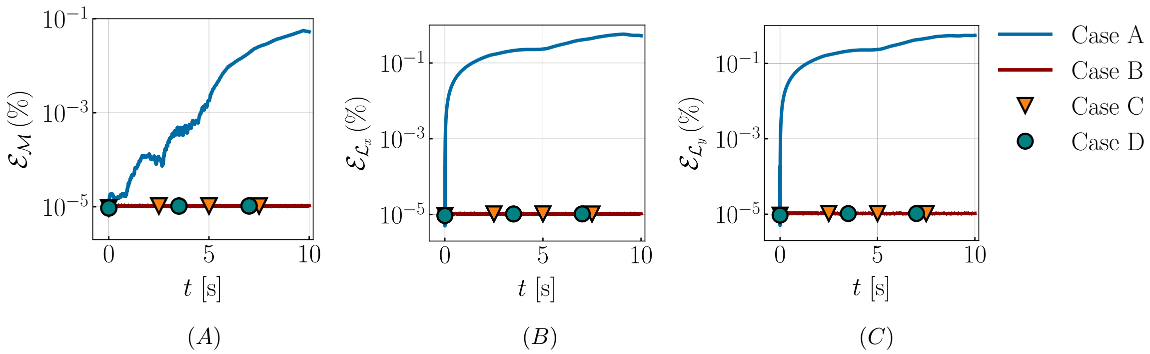
- Case A: RK-2 integrator for momentum equation and SSP-RK3 integrator for mass balance equation.
- Case B: RK-2 integrator for momentum equation, SSP-RK3 integrator for mass balance equation, and residual force $\mathcal{R}\mathbf{u}$ in the momentum equation.
- Case C: RK-2 integrator for both mass and momentum equations.
- Case D: RK-2 integrator for both mass and momentum equations, and residual force $\mathcal{R}\mathbf{u}$ in the momentum equation.

In all four cases we consider two cycles of fixed-point iteration, i.e., $n_{\text{cycles}} = 2$ is used. To assess the performance of the integrator choice, we compute the percentage change in mass and momentum of the system. The relative change is defined as

$$\mathcal{E}_{\mathcal{B}} = \frac{|\beta(t) - \beta_0|}{\beta_0} \times 100, \tag{4.64}$$

in which $\beta = \int_\Omega \rho \, \mathrm{dV}$ and $\mathcal{B} = \mathcal{M}$ denote the relative change in the mass of the system, and $\beta = \int_\Omega \rho\mathbf{u} \, \mathrm{dV}$ and $\mathcal{B} = \mathcal{L}_{\mathbf{x}}$ denote the relative change in the linear momentum of the system. For this problem we expect that the mass and momentum of the system should remain invariant over time. In order to check whether any catastrophic instability occurs in the four cases, we run the simulations till $t = 10$, which corresponds to 10 complete passes of the dense bubble across the domain.

Fig. 4.3 shows the results for the four cases. It can be observed that Case A produces spurious changes in mass and momentum of the system that increase slowly over time. However, adding the residual force in the momentum equation (Case B) reduces the spurious changes in



(A) (B) (C)

Figure 4.3. Percentage change in (A) mass (B) x-momentum and (C) the y-momentum of the system for various choices of time integrators: Case A - RK-2 integrator for momentum equation and SSP-RK3 integrator for mass balance equation; Case B - RK-2 integrator for momentum equation, SSP-RK3 integrator for mass balance equation, and residual force $\mathcal{R}\mathbf{u}$ in the momentum equation; Case C - RK-2 integrator for both mass and momentum equations; Case D - RK-2 integrator for both mass and momentum equations, and residual force $\mathcal{R}\mathbf{u}$ in the momentum equation. $n_{\text{cycles}} = 2$ is used for all four cases.

mass and momentum, and keeps them relatively low and constant over time. Similarly, using the same time integrator (Case C and Case D) also yields very small percentage changes in both mass and momentum of the system that remain steady over time. In all four cases, the simulation remained stable till $t = 10$. The dynamics of the problem are illustrated in Fig. 4.17 of the results Sec. 4.6. Based on the results of this test, we choose the integrator combination of Case D (i.e., RK-2 integrator for integrating both mass and momentum equations and considering an additional residual force in the momentum equation) in this work.

Consistent mass-momentum-enthalpy integrators

The concept of consistent mass-momentum transport is extended to consistent mass-momentum-enthalpy transport in this section. This involves solving the mass balance equation on both cell faces and cell centers. The former is done to obtain the discrete mass flux $\mathbf{m}_{\boldsymbol{\rho}}{}^{\text{fc}}$ for the convective operator $\nabla \cdot (\mathbf{m}_{\boldsymbol{\rho}}{}^{\text{fc}} \otimes \mathbf{u})$ of the momentum equation, while the latter is done to obtain the discrete mass flux $\mathbf{m}_{\boldsymbol{\rho}}{}^{\text{cc}}$ for the advective operator $\nabla \cdot (\mathbf{m}_{\boldsymbol{\rho}}{}^{\text{cc}} h)$ of the enthalpy equation. We use an RK-2 scheme for time integrating both cell-centered mass and enthalpy equations, as explained in the previous section. An additional residual term $\mathcal{R}h^{n+1,k}$ is added to the right hand side of the enthalpy equation for reasons explained previously. Overall the discretized cell-centered enthalpy equation reads as

$$\frac{\breve{\boldsymbol{\rho}}^{n+1,k+1} h^{n+1,k+1} - \breve{\boldsymbol{\rho}}^{n} h^{n}}{\Delta t} + \mathcal{A}\left(h_{\text{lim}}^{(1)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)} \mathbf{u}_{\text{adv}}^{(1)}\right) = \mathcal{R} h^{n+1,k} + (\nabla \cdot \kappa \nabla T)^{n+1,k+1} + Q_{\text{src}}^{n+1,k}, \tag{4.65}$$

in which the advective term is defined as

$$\mathcal{A}\left(h_{\text{lim}}^{(1)}, \breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)} \mathbf{u}_{\text{adv}}^{(1)}\right) \approx \left[\left(\nabla \cdot \left(h_{\text{lim}}^{(1)} \breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)} u_{\text{adv}}^{(1)}\right)\right)_{i-\frac{1}{2},j}, \left(\nabla \cdot \left(h_{\text{lim}}^{(1)} \breve{\boldsymbol{\rho}}_{\text{lim}}^{(1)} v_{\text{adv}}^{(1)}\right)\right)_{i,j-\frac{1}{2}}\right]. \tag{4.66}$$

In the equation above $\mathcal{R}$ is the (cell-centered) residual of the discrete mass balance equation (see also Eq. (4.59)). We explore the combined consistency of mass, momentum, and enthalpy integrators in Sec. 4.6.3.

4.4.4 Complete solution algorithm

In this section we describe the time stepping algorithm used to solve the coupled mass, momentum and enthalpy equations described above. We assume that all quantities of interest, denoted θ, have been computed or are known at time $t = t^n$. To advance the solution to the next time level $n+1$, we employ n_{cycles} number of fixed point iterations (with $k = 0, 1, \ldots, n_{\text{cycles}} - 1$ denoting the iteration number) within a single time step to approximate $\theta^{n+1} = \theta^{n+1, n_{\text{cycles}}-1}$. Within each fixed point iteration, we employ q_{max} number of Newton's iterations (with $m = 0, 1, \ldots, q_{\text{max}} - 1$ denoting the Newton's iteration number) to solve the nonlinear enthalpy equation. At the beginning of the time step, we initialize $\theta^{n+1,k=0} = \theta^n$ for variables $\mathbf{u}, \rho, \phi$, and H. The temperature variable is initialized similarly $T^{n+1,k=0,m=0} = T^n$. Hence, within a single time step (of size $\Delta t = t^{n+1} - t^n$), the Navier-Stokes system and the level-set and Heaviside advection equations are solved for n_{cycles} times and the enthalpy equation is solved (possibly) for $n_{\text{cycles}} \times q_{\text{max}}$ times. For all cases presented in this work, we use $n_{\text{cycles}} = 2$ fixed-point iterations and set $q_{\text{max}} = 5$ for the Newton solver, unless otherwise stated. The governing equations are solved in the order described next.

1. The level set function ϕ^n is first advected with the non-div-free velocity $\mathbf{u}$ to obtain $\phi^{n+1,k+1}$

   $$\frac{\phi^{n+1,k+1} - \phi^n}{\Delta t} + \left(\nabla \cdot [\phi \mathbf{u}]\right)^{n+1,k} = \left(\phi \nabla \cdot \mathbf{u}\right)^{n+1,k}. \tag{4.67}$$

   Under (linear) advection, ϕ does not maintain its signed distance property. A reinitialization procedure suggested by Sussman et al. [84] is used to restore the signed distance property of ϕ. Implementation details about the level set method, and its reinitialization can be found in our prior work [113].

2. A smooth Heaviside function H is used to track the gas-PCM interface. H takes a value of 0 in the gas region, 1 in the solid-liquid PCM region, and transitions smoothly from 0 to 1 around the interface with a prescribed width of $n_{\text{cells}} = 2$ grid cells (of size Δ) on either side of the gas-PCM interface

   $$H = \begin{cases} 0, & \phi(\mathbf{x}) < -n_{\text{cells}}\,\Delta, \\ \frac{1}{2}\left[1 + \frac{1}{n_{\text{cells}}\,\Delta}\,\phi(\mathbf{x}) + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{cells}}\,\Delta}\,\phi(\mathbf{x})\right)\right], & |\phi(\mathbf{x})| \leq n_{\text{cells}}\,\Delta, \\ 1, & \text{otherwise.} \end{cases} \tag{4.68}$$

   Although $H^{n+1,k+1}$ can be defined directly in terms of $\phi^{n+1,k+1}$, we instead advect H^n (defined in terms of ϕ^n using Eq. (4.68)) to approximate $H^{n+1,k+1}$

   $$\frac{H^{n+1,k+1} - H^n}{\Delta t} + \left(\nabla \cdot [H \mathbf{u}]\right)^{n+1,k} = \left(H \nabla \cdot \mathbf{u}\right)^{n+1,k}. \tag{4.69}$$

   This is done to obtain the advective flux of Heaviside $H\mathbf{u}$, which could be used to advect additional scalar variables of a more involved problem. At the end of the time step H^{n+1} is synchronized with ϕ^{n+1} using Eq. (4.68). A third-order accurate cubic upwind interpolation (CUI) scheme [113] is used for advecting ϕ and H in Eqs. (4.67) and (4.69).

CUI satisfies both the convection-boundedness criterion (CBC) (see chapter 12 of [142] for a discussion on high-resolution schemes) as well as the total variation diminishing (TVD) property. Both properties are essential to bound H ($0 \leq H \leq 1$) during its advection.

3. In practical applications, the density contrast between PCM and gas is usually very large. The metal to gas density ratio, for example, is $\rho^{\mathrm{S}}/\rho^{\mathrm{G}} \sim 10^4$. It is important to ensure numerical stability of the scheme when advecting energy/enthalpy and momentum in the domain with very high density ratios. Our recent work proposed an efficient approach for maintaining the stability of isothermal flows (no phase change) with a high density ratio. It involves solving an additional mass balance equation and computing the mass flux $\mathbf{m}_{\boldsymbol{\rho}} = \rho\mathbf{u}$. The same mass flux $\mathbf{m}_{\boldsymbol{\rho}}$ is used in the convective operator of the momentum equation, i.e., $\nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u})$ is discretized as $\nabla \cdot (\mathbf{m}_{\boldsymbol{\rho}} \otimes \mathbf{u})$ in the momentum equation. The same idea is applied to advect enthalpy h as well, i.e., $\nabla \cdot (\rho\mathbf{u}h) = \nabla \cdot (\mathbf{m}_{\boldsymbol{\rho}}h)$.

   The discrete mass balance equation reads as

$$\frac{\breve{\rho}^{n+1,k+1} - \rho^n}{\Delta t} + (\nabla \cdot \mathbf{m}_{\boldsymbol{\rho}})^{n+1,k} = 0, \tag{4.70}$$

   which is solved to obtain $\breve{\rho}^{n+1,k+1}$ and the discrete mass flux $\mathbf{m}_{\boldsymbol{\rho}}$. We use a second-order accurate explicit Runge-Kutta scheme (mid-point rule) for time integrating Eq. (4.70). CUI is used as a limiter to ensure that ρ remains bounded during advection. In Eq. (4.70) ρ^n is defined through EOS. In other words, approximation to the new density $\breve{\rho}^{n+1,k+1}$ is only temporarily used within a time step, after which it is synchronized with the EOS. The synchronization step ensures that (**i**) density does not deviate from the EOS; and (**ii**) gas-PCM interface remains sharp. The latter is due to the use of reinitialized level set function in the EOS.

4. Using the discrete approximation for the new density $\breve{\rho}$ and mass flux $\mathbf{m}_{\boldsymbol{\rho}}$, the nonlinear enthalpy equation is solved to update enthalpy h, temperature T, and liquid fraction φ.

κ and C are defined as functions of φ, which is in turn a function of h. As a result, the enthalpy equation is highly nonlinear.

The discrete enthalpy equation reads as

$$\frac{\breve{\rho}^{n+1,k+1}h^{n+1,k+1} - \rho^n h^n}{\Delta t} + (\nabla \cdot \mathbf{m}_{\boldsymbol{\rho}} h)^{n+1,k} = (\nabla \cdot \kappa \nabla T)^{n+1,k+1} + Q_{\text{src}}. \tag{4.71}$$

In the examples considered in this work, there are no additional heat source/sink terms, i.e., $Q_{\text{src}} = 0$ in Eq. (4.71). This term could be treated implicitly or explicitly depending upon the numerical stiffness and/or its complexity. We omit the Q_{src} term in the remainder of the algorithm.

(a) The nonlinear enthalpy equation is solved using Newton's iteration to obtain $h^{n+1,k+1}$. Specifically, h is linearized using Taylor's expansion as

$$h^{n+1,k+1,m+1} = h^{n+1,k+1,m} + \left.\frac{\partial h}{\partial T}\right|^{n+1,k+1,m} (T^{n+1,k+1,m+1} - T^{n+1,k+1,m}), \tag{4.72}$$

in which m is the inner (Newton) iteration level. Substituting the above equation into Eq. (4.71), we get

$$\frac{\tilde{\rho}^{n+1,k+1}\left(h^{n+1,k+1,m} + \left.\frac{\partial h}{\partial T}\right|^{n+1,k+1,m}\left(T^{n+1,k+1,m+1} - T^{n+1,k+1,m}\right)\right)}{\Delta t}$$
$$\frac{-\rho^n h^n}{\Delta t} + \nabla \cdot (\mathbf{m}_{\boldsymbol{\rho}} h)^{n+1,k} = (\nabla \cdot \kappa \nabla T)^{n+1,k+1,m+1} \tag{4.73}$$

Eq. (4.73) is solved to obtain $T^{n+1,k+1,m+1}$. The h-T relations written in Eq. (4.35) allows an analytical evaluation of $\frac{\partial h}{\partial T}$. Specifically, in the PCM domain, the derivative

$\frac{\partial h}{\partial T}$ is given by

$$\left.\frac{\partial h}{\partial T}\right|_{\text{PCM}} = \begin{cases} C^{\text{S}}, & T < T^{\text{sol}}, \\ \bar{C} + L/(T^{\text{liq}} - T^{\text{sol}}), & T^{\text{sol}} \leq T \leq T^{\text{liq}}, \\ C^{\text{L}}, & T > T^{\text{liq}}, \end{cases} \tag{4.74}$$

and in the gas $\left.\frac{\partial h}{\partial T}\right|_{\text{gas}} = C^{\text{G}}$. Note that the specific enthalpy of the PCM is defined to be a $\mathcal{C}^0$ piecewise-continuous function[6] of T, and its derivative (with respect to T) jumps at T^{sol} and T^{liq}. The PCM and gas regions are distinguished by the Heaviside contour $H = 0.5$ (or alternatively by the $\phi = 0$ contour). Therefore, $\frac{\partial h}{\partial T}$ in the entire domain is defined as:

$$\frac{\partial h}{\partial T} = \begin{cases} \left.\frac{\partial h}{\partial T}\right|_{\text{PCM}}, & H \geq 0.5, \\ \left.\frac{\partial h}{\partial T}\right|_{\text{gas}}, & \text{otherwise.} \end{cases} \tag{4.75}$$

$\frac{\partial h}{\partial T}$ defined in Eq. (4.75) can be made "more smooth" by defining it as $\frac{\partial h}{\partial t} = H\left.\frac{\partial h}{\partial T}\right|_{\text{PCM}} + (1 - H)\left.\frac{\partial h}{\partial T}\right|_{\text{gas}}$. In our numerical experiments we did not observe any major improvement in the Newton solver (in term of its convergence rate) using the "smoother" version of $\frac{\partial h}{\partial t}$. Hence, we make use of Eq. (4.75) in the code. The linear system of Eq. (4.73) is solved using a geometric multigrid preconditioned FGMRES solver with a relative tolerance of 10^{-9}.

(b) Update enthalpy $h^{n+1,k+1,m+1}$ using the Taylor series expansion (Eq. (4.72)) and $T^{n+1,k+1,m+1}$.

---

[6] Strictly speaking, the specific enthalpy h of a pure PCM cannot be a $\mathcal{C}^0$ continuous function. This is because a large amount of latent heat is released/absorbed at its solidification/melting temperature T_m and h jumps at T_m. In enthalpy methods, this condition is relaxed and the latent heat is assumed to be released over a temperature interval $\Delta T = T^{\text{liq}} - T^{\text{sol}}$.

(c) Update $T^{n+1,k+1,m+1}$ and $\varphi^{n+1,k+1.m+1}$ using $h^{n+1,k+1,m+1}$ and analytical T-h and φ-h relations written in the Sec. 4.4.1, respectively.

(d) Update thermophysical properties (κ, C, and μ) using $\varphi^{n+1,k+1,m+1}$. In spite of the fact that the model does not need specific heat C directly, we update it for consistency reasons. The updated C values can be used for post-processing or to model additional physics, for example.

(e) Compute the relative change in liquid fraction

$$\epsilon = \frac{||\varphi^{n+1,k+1,m+1} - \varphi^{n+1,k+1,m}||_2}{||1 + \varphi^{n+1,k+1,m}||_2} \tag{4.76}$$

The Newton solver is deemed to be converged if $\epsilon \leq 10^{-8}$ or if $m + 1 = q_{\text{max}} = 5$ iterations have completed.

5. Finally we solve the momentum and low Mach equations together

$$\begin{aligned}
&\frac{\breve{\rho}^{n+1,k+1}\mathbf{u}^{n+1,k+1} - \rho^n\mathbf{u}^n}{\Delta t} + (\nabla \cdot [\mathbf{m}_{\boldsymbol{\rho}} \otimes \mathbf{u}])^{n+1,k} = -\nabla p^{n+\frac{1}{2},k+1} \\
&+ (\nabla \cdot \left[\mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right])^{n+\frac{1}{2},k+1} - A_d^{n+1,k+1}\mathbf{u}^{n+1,k+1} + \mathbf{f}_{\text{st}}{}^{n+\frac{1}{2},k+1},
\end{aligned} \tag{4.77}$$

$$\nabla \cdot \mathbf{u} = \begin{cases} 0, & H < 0.5 \text{ (i.e., in the gas phase)}, \\ 0, & h < h^{\text{sol}}, \\ -\frac{\rho^{\text{S}}\rho^{\text{L}}}{\rho^2}(\rho^{\text{L}} - \rho^{\text{S}})H\left(\nabla \cdot \kappa \nabla T\right) \\ \quad \dfrac{(h^{\text{liq}} - h^{\text{sol}})}{\left(h(\rho^{\text{L}} - \rho^{\text{S}}) - \rho^{\text{L}}h^{\text{liq}} + \rho^{\text{S}}h^{\text{sol}}\right)^2}, & h^{\text{sol}} \leq h \leq h^{\text{liq}}, \\ 0, & h > h^{\text{liq}}. \end{cases} \tag{4.78}$$

to update velocity $\mathbf{u}^{n+1,k+1}$ and pressure $p^{n+\frac{1}{2},k+1}$. In Eq. (4.77) we use the same discrete density $\breve{\rho}^{n+1,k+1}$ and mass flux $\mathbf{m}_{\boldsymbol{\rho}}$ that we obtained from solving Eq. (4.70). This maintains consistency between mass and momentum transport for high-density ratio flows. The

Carman-Kozeny drag coefficient A_d employs an updated value of $\varphi^{n+1,k+1}$ obtained from solving the enthalpy Eq. (4.71). The surface tension force $\mathbf{f}_{\text{st}}$ acting on the liquid-gas interface is modeled using the continuous surface tension formulation [119, 155, 156]. The continuous surface tension force reads as

$$\mathbf{f}_{\text{st}} = \varphi \frac{2\breve{\rho}}{\rho^{\text{L}} + \rho^{\text{G}}} \left(\sigma \mathcal{C} \nabla \widetilde{B} \right), \tag{4.79}$$

in which σ is the uniform liquid-gas surface tension coefficient and $\mathcal{C}(\phi)$ is the curvature of the interface computed from the level set function $\mathcal{C}^{n+\frac{1}{2},k+1} = -\nabla \cdot \left(\frac{\nabla \phi}{||\nabla \phi||} \right)$. In Eq. (4.79), $\widetilde{B}(\phi)$ represents a mollified Heaviside function that ensures the surface tension force only acts near the PCM-gas region. In addition, the multiplier φ limits the influence of surface tension to liquid and gas. We use the mid-point value of $\phi^{n+\frac{1}{2},k+1} = \frac{1}{2}\left(\phi^{n+1,k+1} + \phi^{n}\right)$ in computing $\mathcal{C}$ and $\widetilde{B}$. The right-hand side of the discrete low Mach Eq. (4.78) is evaluated by using the most updated values of $H^{n+1,k+1}, h^{n+1,k+1}, T^{n+1,k+1}$, and $\rho^{n+1,k+1}$. The linear system of Eqs. (4.77) and (4.78) is solved monolithically for the coupled $\mathbf{u}$-p system using an FGMRES solver with a relative tolerance of 10^{-9}. The FGMRES solver employs a projection method-based preconditioner as explained in Thirumalaisamy et al. [157].

Each case presented in this work uses a uniform time step size Δt, and the CFL number does not exceed 0.5.

4.5 Adaptive mesh refinement (AMR)

We use a structured adaptive mesh refinement framework to discretize the governing equations. The three-dimensional computational domain $\Omega \in [0, L] \times [0, H] \times [0, D]$ is discretized into multiple levels of structured grids. The ratio between each successive grid level is denoted n_{ref}. The cell dimensions on the coarsest level are $\Delta x_0 = \frac{L}{N_{x0}}$, $\Delta y_0 = \frac{H}{N_{y0}}$, $\Delta z_0 = \frac{D}{N_{z0}}$, in which N_{x0}, N_{y0}, and N_{z0} are the number of cells in the x, y, and z directions. The cell dimensions on the finer levels are then $\Delta \mathbf{x}_{\min} = \Delta \mathbf{x}_0 / n_{\text{ref}}^{l-1}$, in which l is the grid level number.

We support both static and adaptive meshes in our framework. During static refinement, coarse mesh levels are refined over a fixed area of space, and mesh configuration remains constant over time. It is applicable when the region of interest exhibits minimal movement. With dynamic meshes (or adaptive meshes), coarse level cells are tagged/untagged throughout the simulation based on user-specified criteria. In our framework, we employ two tagging criteria:

1. Tagging cells based on the signed distance function: To resolve the gas-PCM interface with sufficient mesh resolution, a *value-based tagging* criterion is used. A cell is tagged for grid refinement when its signed distance function value is within zero threshold, e.g., $|\phi| \leq 2\Delta x_0$. Using numerical simulations, we find that this amount of tagging is sufficient to resolve the sharp variation in material properties (ρ, κ, μ) across the gas-PCM interface.
2. Tagging cells based on the liquid fraction variable: In order to capture the mushy region and the associated volume change of PCM, it is necessary to use sufficiently refined grids to resolve the $\Omega^{\mathrm{M}}(t)$ region. We can identify the mushy region by either: (i) probing the value of φ, e.g., $0.3 \leq \varphi \leq 0.8$; or by (ii) probing the gradient of φ, e.g., $|\nabla\varphi| > 0$. We term the former tagging criterion as *φ-based tagging* or *value-based tagging* and the latter as *$\nabla\varphi$-based tagging* or *gradient-based tagging.*

4.6 Results and discussion

4.6.1 Validation of the low Mach enthalpy method with analytical solutions

We validate the novel low Mach EM by examining the Stefan problem (solidification of a liquid PCM) discussed in Sec. 4.3, with three different density ratios $R_\rho = \rho^{\mathrm{S}}/\rho^{\mathrm{L}}$ that lead to: (**1**) no flow ($R_\rho = 1$); (**2**) flow with volume expansion ($R_\rho < 1$); and (**3**) flow with volume shrinkage ($R_\rho > 1$). The numerical model consists of a quasi one-dimensional computational domain $\Omega \in [0, 1] \times [0, 0.05]$ with $N_x \times N_y = 1280 \times 64$ grid cells. The domain is periodic in the y-direction. Initially, liquid occupies the entire domain at $T_i = 973.6$ K. The left boundary

Table 4.1. Thermophysical properties used to simulate the Stefan problem

Property	Value
Thermal conductivity of solid κ^{S}	211 W/m.K
Thermal conductivity of liquid κ^{L}	91 W/m.K
Specific heat of solid C^{S}	910 J/kg.K
Specific heat of liquid C^{L}	1042.4 J/kg.K
Solidification temperature T_m	933.6 K
Bulk phase change temperature T_r	933.6 K
Liquidus temperature T^{liq}	938.6 K
Solidus temperature T^{sol}	928.6 K
Latent heat L	383840 J/kg

($x = 0$) is set to $T_o = 298.6$ K and the right boundary ($x = 1$) is adiabatic (homogeneous Neumann). The flow solver uses zero-velocity and zero-pressure/outflow boundary conditions at the left and right ends, respectively. PCM's thermophysical properties are largely aluminum-based, and are listed in Table 4.1. In this case, both fluid and solid viscosities are set to zero. For simplicity, we take the bulk phase change temperature T_r equal to the solidification temperature T_m, so $L^{\text{eff}} = L$. The temperature interval ΔT between solidus and liquidus and the grid size are selected based on convergence studies presented in Secs. 4.6.1 and 4.6.1, respectively.

The no volume change case

Fig. 4.4(A) compares CFD results[7] for the interface position $x^* = s(t)$ and temperature distribution in the solid and liquid phases against the analytical solutions for $R_\rho = 1$ case. We take solid and liquid densities to be the same $\rho^{\text{S}} = \rho^{\text{L}} = 2475$ kg/m^3. Table 4.1 lists the rest of the thermophysical properties. Simulation is run until $t = 10$ s with a uniform time step size of $\Delta t = 10^{-3}$ s. The numerical solid-liquid interface is defined by an iso-contour value of 0.5 of the liquid fraction φ. It is evident from the figure that the interface position and temperature profiles match the analytical solution very well at different times. The analytical solution derived in this work reduces to the solution of the standard Stefan problem when R_ρ equals 1. The top row of Fig. 4.4 shows both the new and standard Stefan problem solutions for x^*; the latter solution is

---

[7] This benchmark test is provided in IBAMR GitHub within the directory `examples/phase_change/ex0`.

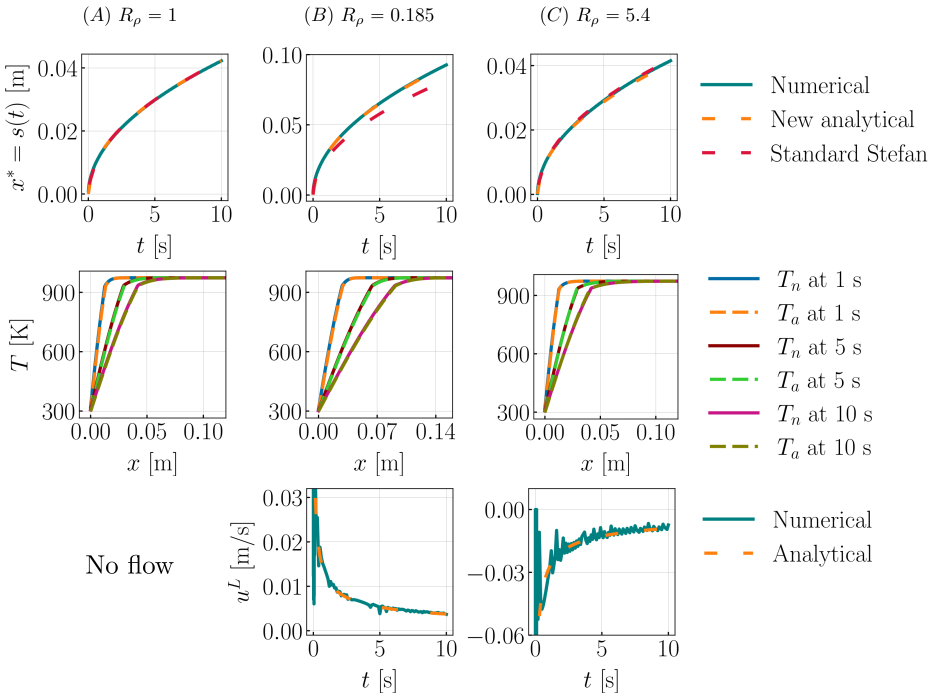


Figure 4.4. Comparison of CFD and analytical solutions to the Stefan problem at various density ratios $R_\rho = \rho^{\mathrm{S}}/\rho^{\mathrm{L}}$. (A) Comparison of the solid-liquid interface position and temperature distribution in the domain when the liquid and solid densities are the same. In this case, there is no fluid flow. (B) and (C), respectively, compare CFD and analytical solutions (interface position and temperature and [uniform] velocity distributions) for the expansion ($R_\rho = 0.185$) and shrinkage ($R_\rho = 5.4$) cases.

from the Hahn and Özişik textbook [139].

The expansion case

For this case, the liquid and solid densities are assumed to be $\rho^{\mathrm{L}} = 2700$ and $\rho^{\mathrm{S}} = 500$ kg/m^3, respectively. Other thermophysical properties of aluminum-based PCM can be found in Table 4.1. A uniform time step size of $\Delta t = 10^{-4}$ s is used throughout the simulation to maintain the CFL number below 0.5. Analytical and CFD solutions[8] are compared in Fig. 4.4(B). There is excellent agreement between the two. As can be seen, the standard Stefan solution

---

[8] This benchmark test is provided in IBAMR GitHub within the directory `examples/phase_change/ex1`.

underpredicts the interface position. This is because it does not take into account the additional flow that is generated in the direction of interface propagation. In addition, the temperature and liquid velocity profiles agree well with the new analytical model. At $t = 0^+$ the interface velocity $u^* = \mathrm{ds}/\mathrm{dt} \ \propto \ 1/\sqrt{\mathrm{t}} \rightarrow \infty$. As fluid velocity is proportional to interface speed (see Eq. (6.22)), the CFD simulation produces large u^{L} values at the beginning[9]. The pressure profiles from CFD and analytical methods for this case are compared in Sec. 4.6.1.

The shrinkage case

In order to simulate shrinkage, liquid and solid densities are assumed to be $\rho^{\mathrm{L}} = 500$ and $\rho^{\mathrm{S}} = 2700$ kg/m^3, respectively. All other simulation and thermophysical parameters are kept the same as in the expansion case. Results[10] are shown in Fig. 4.4(C). Liquid-solid interface location matches the analytical solution very well. Temperature and velocity profiles are also in good agreement with the analytical solution. $R_\rho > 1$ results in fluid flow opposite to the interface propagation, since fluid shrinks as it solidifies. Solidification rate is (slightly) reduced as additional hot fluid is pulled towards the solidifying front. Both the new analytical solution and standard Stefan solution (without a density jump) predict an interface position that is qualitatively similar. Volume shrinkage during solidification may seem insignificant based on this analysis. This argument is refuted in Sec. 4.6.6, in which we present a modeling study that highlights the importance of volume shrinkage in causing pipe defects during metal casting.

ΔT convergence study for the Stefan problem with volume change

The thickness of the mushy zone for the enthalpy method depends on the temperature interval $\Delta T = T^{\mathrm{liq}} - T^{\mathrm{sol}}$ around the phase change temperature T_m. The numerical solution is expected to approach the analytical one as $\Delta T \rightarrow 0$. In practice ΔT is kept finite so that the latent heat can be absorbed or released within the grid-resolved mushy region. While simulating,

---

[9] At $t = 0$, the liquid is taken to be quiescent. u^{L} starts with a zero value but jumps to a large value immediately for CFD velocity profiles.

[10] This benchmark test is provided in IBAMR GitHub within the directory `examples/phase_change/ex1`.

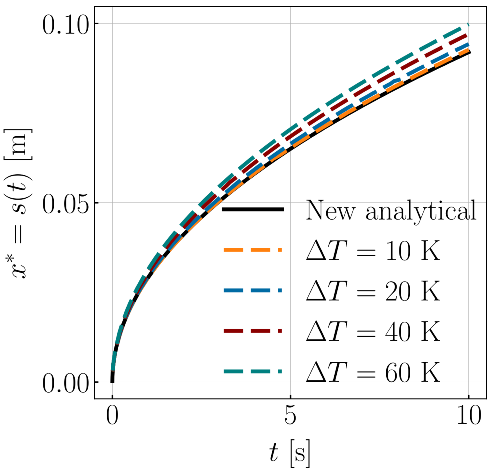


Figure 4.5. Effect of temperature interval $\Delta T = T^{\text{liq}} - T^{\text{sol}}$ on the numerical solution of the Stefan problem with volume expansion ($R_\rho = 0.185$). The grid size considered is $N_x \times N_y = 1280 \times 64$. A uniform time step size of $\Delta t = 10^{-4}$ is used for all the temperature intervals.

if ΔT is set too small (but finite), the mushy region becomes very narrow and falls in the sub-grid region. Numerical oscillations are produced by intermittent appearances and disappearances of the mushy region during simulation. In order to select grid size and temperature interval for the EM, a convergence study is necessary.

For the Stefan problem simulated in this section, we consider $\Delta T = \{10, 20, 40, 60\}$ K. A convergence study is performed for the expansion problem $R_\rho = 0.185$. A fixed grid size of $Nx \times Ny = 1280 \times 64$ is chosen for the study. Results for the interface position $x^* = s(t)$ at various temperature intervals are shown in Fig. 4.5. As expected, a smaller temperature interval leads to a more accurate solution for the EM. Additionally, we also considered ΔT = 2.5 and 5 K; the numerical solutions either did not change appreciably or exhibited minor oscillations at coarse grid resolutions for these values of ΔT (data not presented). Therefore, we use $\Delta T = 10$ K in the numerical simulations for a PCM that is largely aluminum-based, unless otherwise stated. We also perform a grid convergence study using three grids: coarse, medium and fine of size $N_x \times N_y = 640 \times 32, 1280 \times 64, 2560 \times 128$, respectively. $\Delta T = 10$ K is used for three grids. As observed in Fig. 4.6, the analytical and numerical solutions agree reasonably well. Consequently, we use medium grid to simulate the Stefan problems of this section.

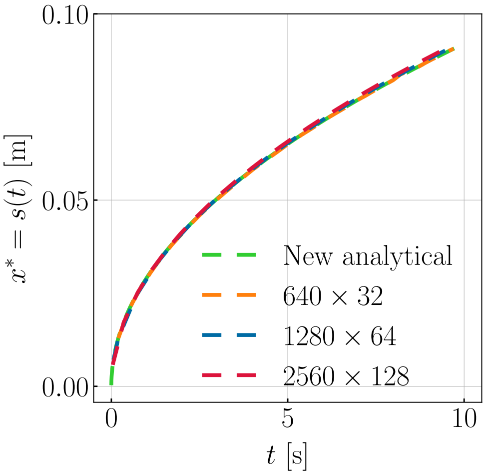


Figure 4.6. Grid convergence study for the Stefan problem with volume expansion ($R_\rho = 0.185$). $\Delta T = 10$ K is used for all grids. Uniform time step sizes used for the coarse, medium, and fine grids are $\Delta t = 10^{-3}, 10^{-4}$, and 5×10^{-5} s, respectively.

Spatio-temporal convergence rate of the low Mach enthalpy method

This section investigates the spatio-temporal convergence rate of the proposed low Mach enthalpy method. A grid convergence study is conducted for the three Stefan problems of Secs. 4.6.1- 4.6.1 using four grid sizes: $N_x \times N_y = \{320 \times 16, 640 \times 32, 1280 \times 64, 2560 \times 128\}$. A uniform time step size of $\Delta t = 4 \times 10^{-4}$ s is employed for the coarse grid $N_x \times N_y = 320 \times 16$ and for each successive grid, the time step size is halved. This ensures the CFL number stays below 0.5 for all four grids. The temperature interval between liquidus and solidus is taken to be $\Delta T = 10$ K, as determined by the results of the previous section. The $\mathcal{L}^2$ error for a quantity ψ is defined to be the root mean squared error (RMSE) of the vector $||\mathcal{E}_\psi||_{\text{RMSE}} = ||\psi_{\text{reference}} - \psi_{\text{numerical}}||_2/\sqrt{\mathcal{N}}$. Here, $\mathcal{N}$ denotes the size of the vector $\mathcal{E}_\psi$. Two different reference solutions are considered here: (i) analytical and (ii) numerical solutions obtained using the finest grid resolution (2560×128). Errors based on analytical solutions indicate the convergence of the present diffuse interface approach to its sharp interface counterpart. This is when the thickness of the mushy zone decreases. In contrast, errors based on the finest grid solutions indicate the spatio-temporal convergence rate of the diffuse interface model itself. We present errors as a function of mesh resolution for the interface position $x^* = s(t)$ for the entire simulation period ($0 \leq t \leq 10$ s) and for temperature $T(x, t)$ in the entire domain ($0 \leq x \leq l$) at $t = 5$ s. The uniform velocity in the liquid domain is a scalar multiple of the interface position (see Eq. (4.5)). Therefore, error in the interface position is also a measure of error in the flow field.

Fig. 4.7 illustrates the spatio-temporal convergence rate of the numerical solution to the Stefan problem for the matched density case ($\rho^{\text{L}} = \rho^{\text{S}}$). Convergence rates based on the analytical solution are shown in Fig. 4.7 (A). As can be observed in the figure the present method exhibits close to first order with respect to the temperature solution. In the case of interface position, the present method exhibits a convergence rate between first and second order. We note that the interface location in diffuse interface enthalpy methods is implicitly defined as the 0.5 contour of the liquid fraction variable φ. Extracting the interface location x^* itself introduces an unavoidable interpolation error that also contributes to the non-uniform convergence rate of $\mathcal{E}_{x^*}$. In this work we rely on VisIt software's [158] excellent post-processing capabilities to extract the interface location from the (distributed memory parallel) φ data. Fig. 4.7 (B) illustrates the convergence rates based on the finest grid numerical solutions of x^* and T. Here, the convergence rate between first and second order is observed for both quantities.

Figs. 4.8 (A) and (B) present the convergence rates for the Stefan problem exhibiting material expansion upon solidification ($R_\rho < 1$) using analytical and finest grid numerical solutions, respectively. The temperature convergence rate trend is the same as in the constant density case: close to first order convergence rate with respect to the analytical solution and between first and second order with respect to the finest grid numerical solution. Errors for interface position exhibit a non-monotonic convergence rate, however error magnitudes are low (on the order of 10^{-3}). For larger temperature intervals ($\Delta T > 10$ K) the convergence rate of the interface position error is slightly better, but the error magnitude is higher (data not presented for brevity).

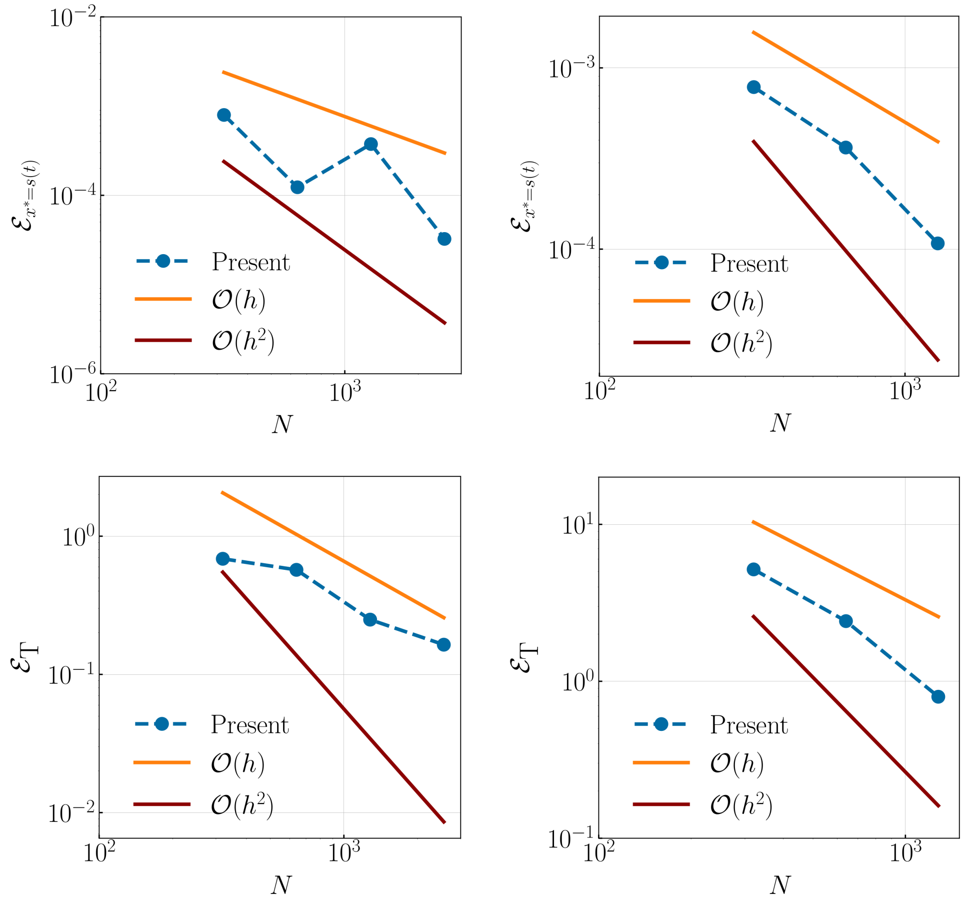
Finally, the method's accuracy is tested for the Stefan problem exhibiting material shrinkage upon solidification ($R_\rho > 1$) and the results are presented in Fig. 4.9. Similar to the previous two cases, we observe close to first order convergence rate of temperature errors with respect to the analytical solution. We also observe close to second order convergence rate with respect to finest grid numerical solution. Interface position errors, though small saturate at fine grids.

Solution to the transcendental equation

In Fig. 4.10, we plot λ versus t that is obtained by solving the transcendental Eq. (4.25) for the expansion and shrinkage cases considered in this section. It is evident from the plot that λ varies during the early stages of solidification (when kinetic energy dominates in the Stefan condition) before reaching a steady state. The relative magnitude of two terms comprising the interface velocity ds/dt (see Eq. (4.24)) are also compared for the expansion and shrinkage case in Fig. 4.10. We can observe that the second term involving $\mathrm{d}\lambda/\mathrm{dt}$ is much smaller (at least six orders of magnitude) than the first term $\lambda/\sqrt{t}$, so it is justified to solve the simpler Eq. (4.25) rather than the original, more complex Eq. (6.30) for $\lambda(t)$. For the thermophysical properties considered in this work, λ variation is quite small and can arguably be ignored. Nevertheless, it is possible to include the kinetic energy jump term in the analytical solution to the two-phase Stefan problem.

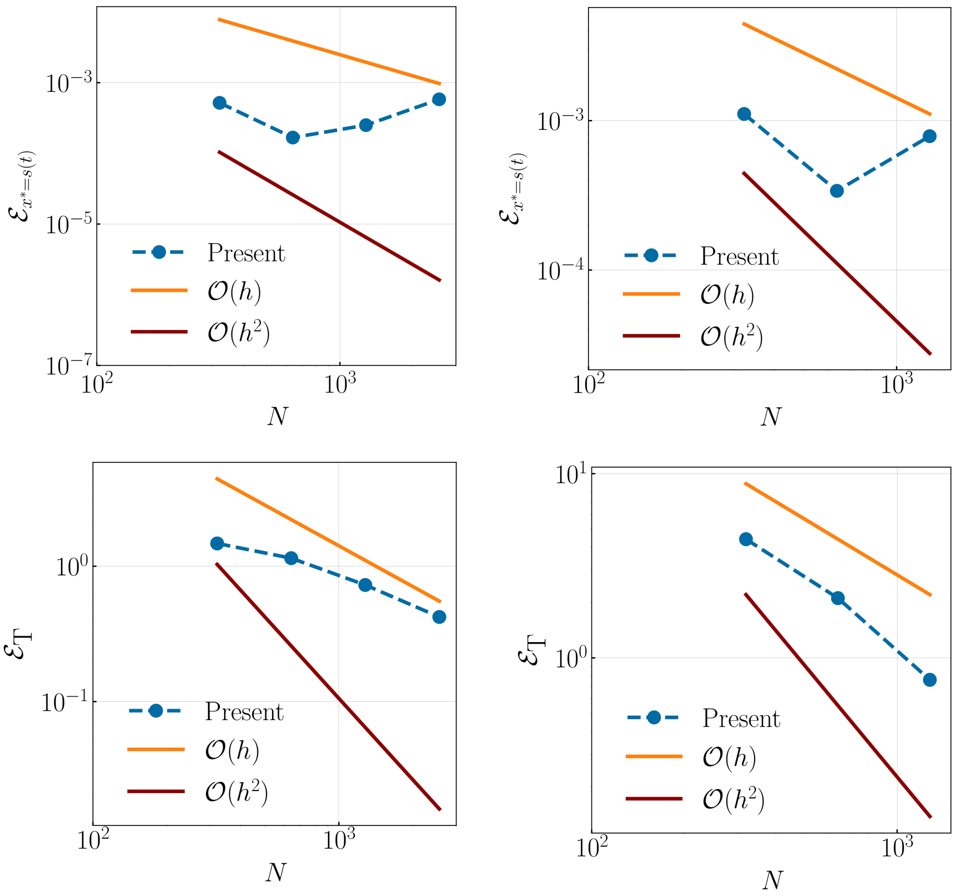
Pressure jump across the interface for the Stefan problem exhibiting volume change

Based on the analytical solution of the one-dimensional Stefan problem considering fluid flow, pressure varies linearly in the liquid phase and remains uniform in the solid phase. The numerical solution also exhibits this behavior of pressure variation. Figs. 4.11(A) and 4.11(B) show pressure in the entire domain at $t = 5$ and 10 s, respectively for the expansion case ($R_\rho = 0.185$). Zoomed-in plots are required to discern variation in liquid pressure since solid pressure is much greater. Although the numerical and analytical models predict the same trend in pressure variation, the numerical values differ substantially; numerical pressure values are much larger than the analytical ones (data not shown for brevity). This is due to the diffuse-interface



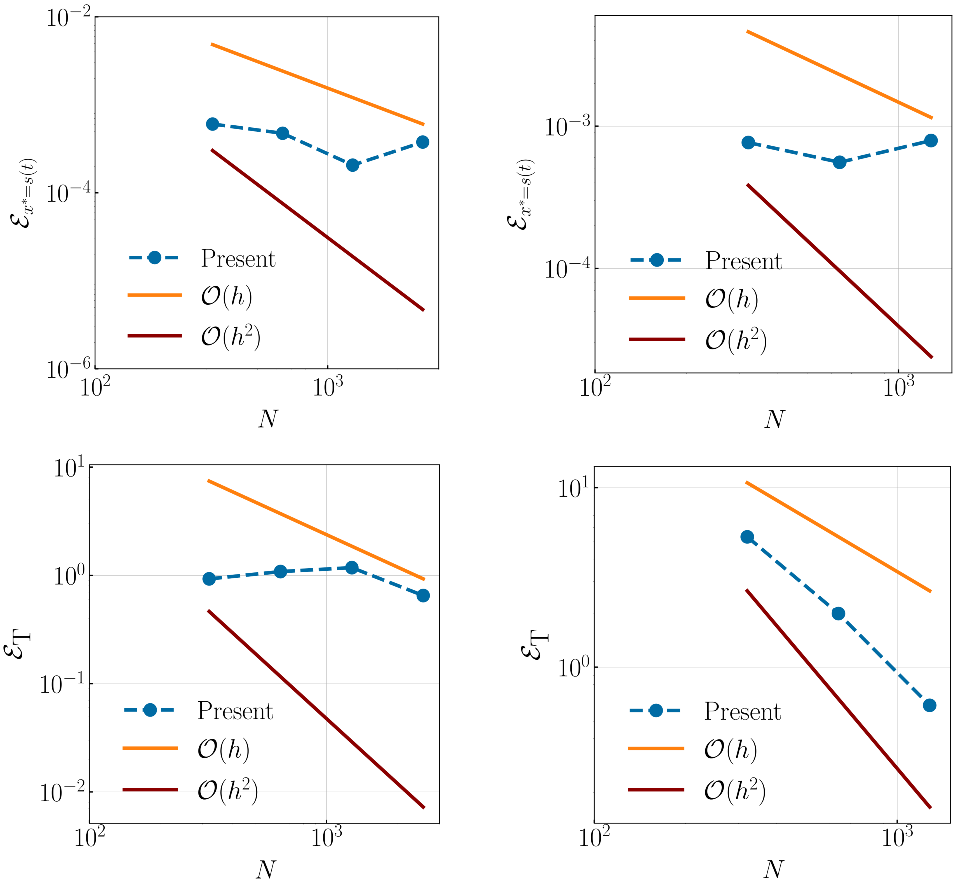
(A) (B)

Figure 4.7. Convergence rates of the low Mach enthalpy method considering the Stefan problem with matched densities of solid and liquid phases ($\rho^{\mathrm{L}} = \rho^{\mathrm{S}}$). The $\mathcal{L}^2$ error for a quantity ψ is defined to be the root mean squared error (RMSE) of the vector $||\mathcal{E}_\psi||_{\mathrm{RMSE}} = ||\psi_{\text{reference}} - \psi_{\text{numerical}}||_2/\sqrt{\mathcal{N}}$, in which $\mathcal{N}$ denotes the size of the vector $\mathcal{E}_\psi$. Here, Ψ represents the interface position $x^* = s(t)$ and temperature $T(x,t)$ in the domain. The reference solutions are (A) the analytical solutions and (B) the finest grid ($N_x \times N_y = 2560 \times 128$) numerical solutions. Errors are presented as a function of mesh resolution for the interface position $x^* = s(t)$ for the entire simulation period ($0 \leq t \leq 10$ s) and for temperature $T(x,t)$ in the entire domain ($0 \leq x \leq l$) at $t = 5$ s. The temperature interval between liquidus and solidus is $\Delta T = 10$ K.



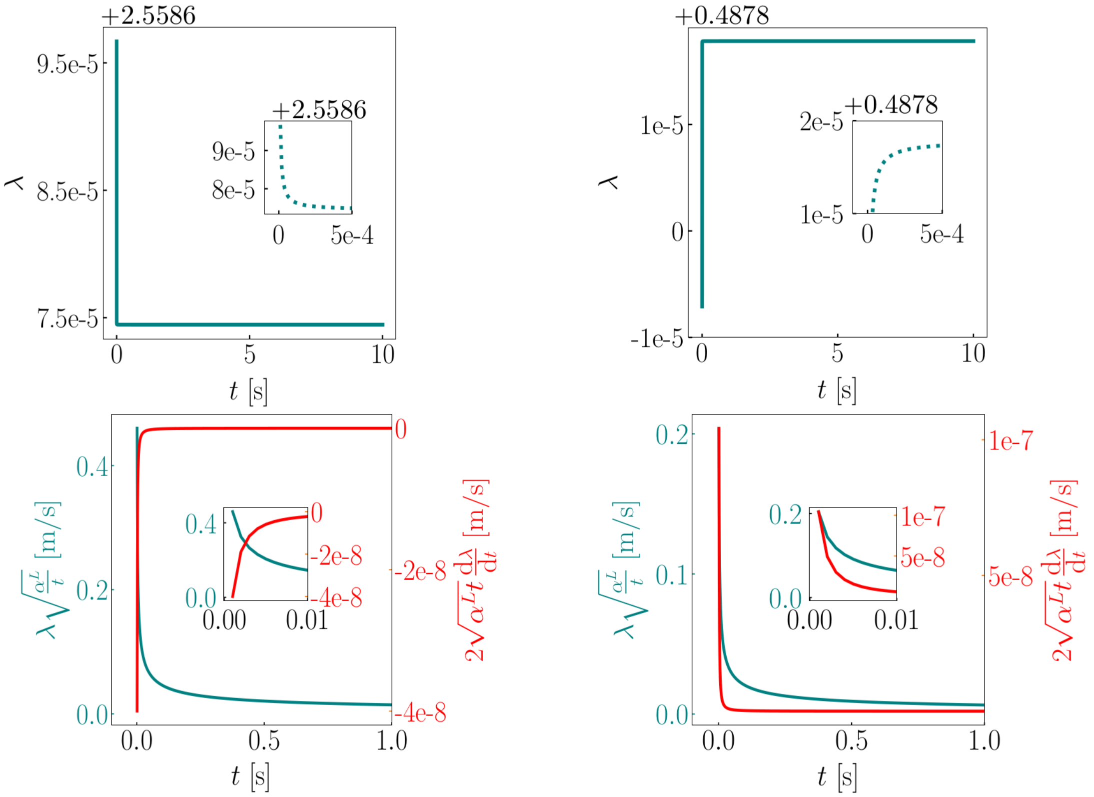
(A) (B)

Figure 4.8. Convergence rates of the low Mach enthalpy method considering the Stefan problem with the solid phase density lower than the liquid phase ($R_\rho < 1$). The $\mathcal{L}^2$ error for a quantity ψ is defined to be the root mean squared error (RMSE) of the vector $||\mathcal{E}_\psi||_{\mathrm{RMSE}} = ||\psi_{\text{reference}} - \psi_{\text{numerical}}||_2/\sqrt{\mathcal{N}}$, in which $\mathcal{N}$ denotes the size of the vector $\mathcal{E}_\psi$. Here, Ψ represents the interface position $x^* = s(t)$ and temperature $T(x,t)$ in the domain. The reference solutions are (A) the analytical and (B) the finest grid ($N_x \times N_y = 2560 \times 128$) numerical solutions. Errors are presented as a function of mesh resolution for the interface position $x^* = s(t)$ for the entire simulation period ($0 \leq t \leq 10$ s) and for temperature $T(x,t)$ in the entire domain ($0 \leq x \leq l$) at $t = 5$ s. The temperature interval between liquidus and solidus is $\Delta T = 10$ K.



(A) (B)

Figure 4.9. Convergence rates of the low Mach enthalpy method considering the Stefan problem with the solid phase density higher than the liquid phase ($R_\rho > 1$). The $\mathcal{L}^2$ error for a quantity ψ is defined to be the root mean squared error (RMSE) of the vector $||\mathcal{E}_\psi||_{\mathrm{RMSE}} = ||\psi_{\text{reference}} - \psi_{\text{numerical}}||_2/\sqrt{\mathcal{N}}$, in which $\mathcal{N}$ denotes the size of the vector $\mathcal{E}_\psi$. Here, Ψ represents the interface position $x^* = s(t)$ and temperature $T(x,t)$ in the domain. The reference solutions are (A) the analytical and (B) the finest grid ($N_x \times N_y = 2560 \times 128$) numerical solutions. Errors are presented as a function of mesh resolution for the interface position $x^* = s(t)$ for the entire simulation period ($0 \leq t \leq 10$ s) and for temperature $T(x,t)$ in the entire domain ($0 \leq x \leq l$) at $t = 5$ s. The temperature interval between liquidus and solidus is $\Delta T = 10$ K.



(A) (B)

Figure 4.10. Variation of λ and interface speed's two components $\lambda\sqrt{\frac{\alpha^{\mathrm{L}}}{t}}$ and $2\sqrt{\alpha^{\mathrm{L}}t}\frac{\mathrm{d}\lambda}{\mathrm{d}t}$ (see Eq. (4.24)) as a function of time for the Stefan problem with (A) volume expansion and (B) volume shrinkage. $\lambda(t)$ is obtained by solving the transcendental equation (Eq. (4.25)) using MATLAB's `fzero` function. The transcendental equation is solved at $t = \Delta t$ and onwards. $\frac{\mathrm{d}\lambda}{\mathrm{d}t}$ is computed from λ in a post-processing step.

formulation of the EM. Specifically, a Carman-Kozeny drag model is used in the EM to enforce no flow in the solid phase. In a diffuse formulation, velocity changes continuously from zero to a finite value within the mushy region. The pressure jump across the mushy region helps the fluid to "squeeze" through. This is similar to the Darcy-Brinkman model of flow through porous regions

$$\mathbf{u} \propto -\nabla p.$$

The numerical pressure jump $[\![p]\!] = p^{\mathrm{L}} - p^{\mathrm{S}}$ ($\sim \nabla p$) across the mushy region is plotted as a function of liquid velocity u^{L}. When the flow has subsided and the Darcy-Brinkman model becomes applicable, the curve is shown for $t > 2$ s. There is a linear relationship between $[\![p]\!]$ and u^{L}, confirming our hypothesis that the numerical pressure jump occurs to push fluid through the mushy region. Additionally, the diffuse-domain momentum equation provides a magnitude scale of $[\![p]\!]$

$$\begin{aligned} -\frac{\partial p}{\partial x} &\sim A_d\, u^{\mathrm{L}} \\ \hookrightarrow p^{\mathrm{S}} - p^{\mathrm{L}} = [\![-p]\!] &\sim A_d\, \Delta\, u^{\mathrm{L}}, \end{aligned} \tag{4.80}$$

in which Δ is the cell size and $A_d(\varphi_{\mathrm{S}})$ is the drag coefficient. Here, we have ignored the convective scale ($\rho\frac{\partial u^{\mathrm{L}}}{\partial t}$) in the mushy zone as it is several orders lower than the pressure gradient and drag force terms (data not shown). Eq. (4.80) suggests that the slope of $[\![-p]\!]$ versus u^{L} curve is $A_d\Delta$. This is confirmed from Fig. 4.11 where a line of slope $A_d\Delta$ evaluated at a solid fraction $\varphi_{\mathrm{S}} = 0.833$ (this value of φ_{S} is obtained by equating the slope of the best-fit linear curve to $A_d\Delta$, and then solving for φ_{S}) captures the $[\![-p]\!]$ versus u^{L} trend reasonably well.

Capturing the mushy region with AMR

Prior works [113, 159, 160] have confirmed our AMR framework's ability to adaptively refine/coarsen the grid near gas-liquid/gas-solid interfaces in isothermal flows without any phase change. Here we test its capability for the more complex scenario: capturing the evolving mushy

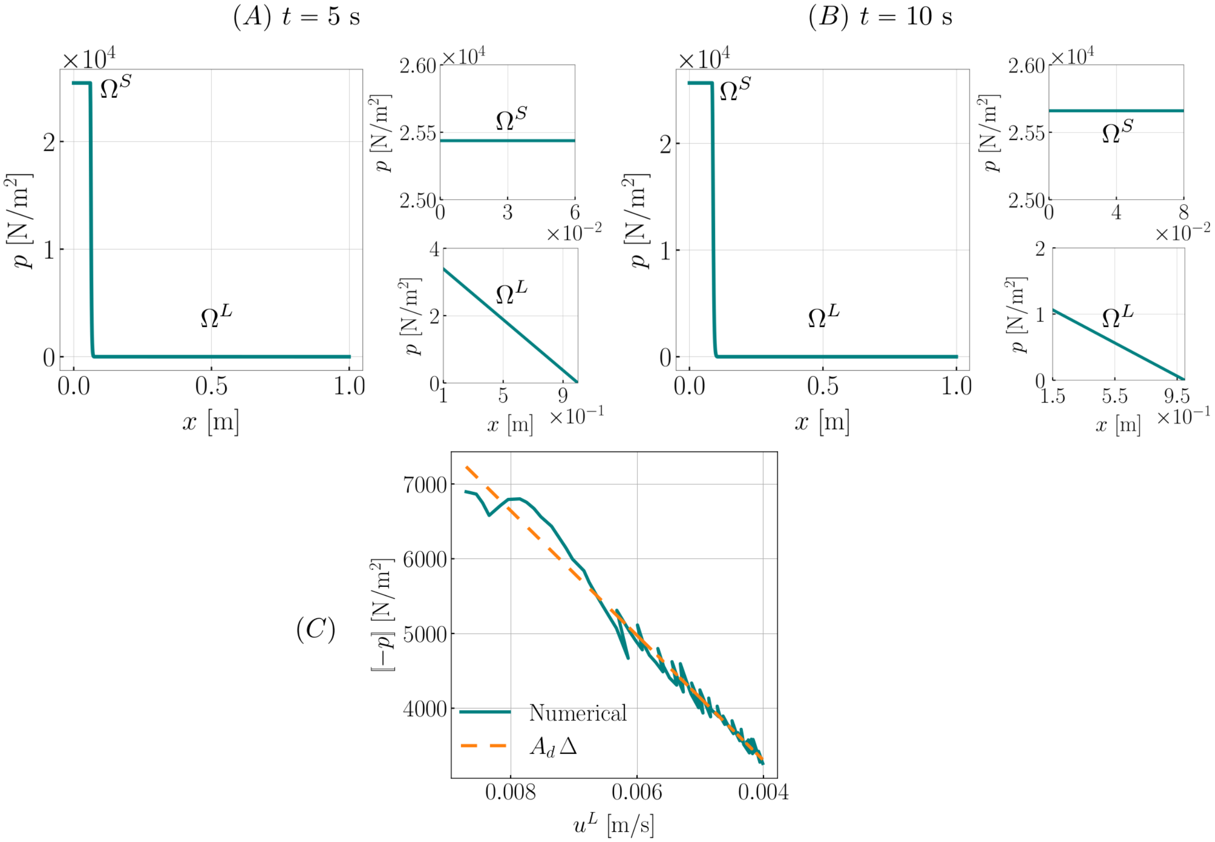


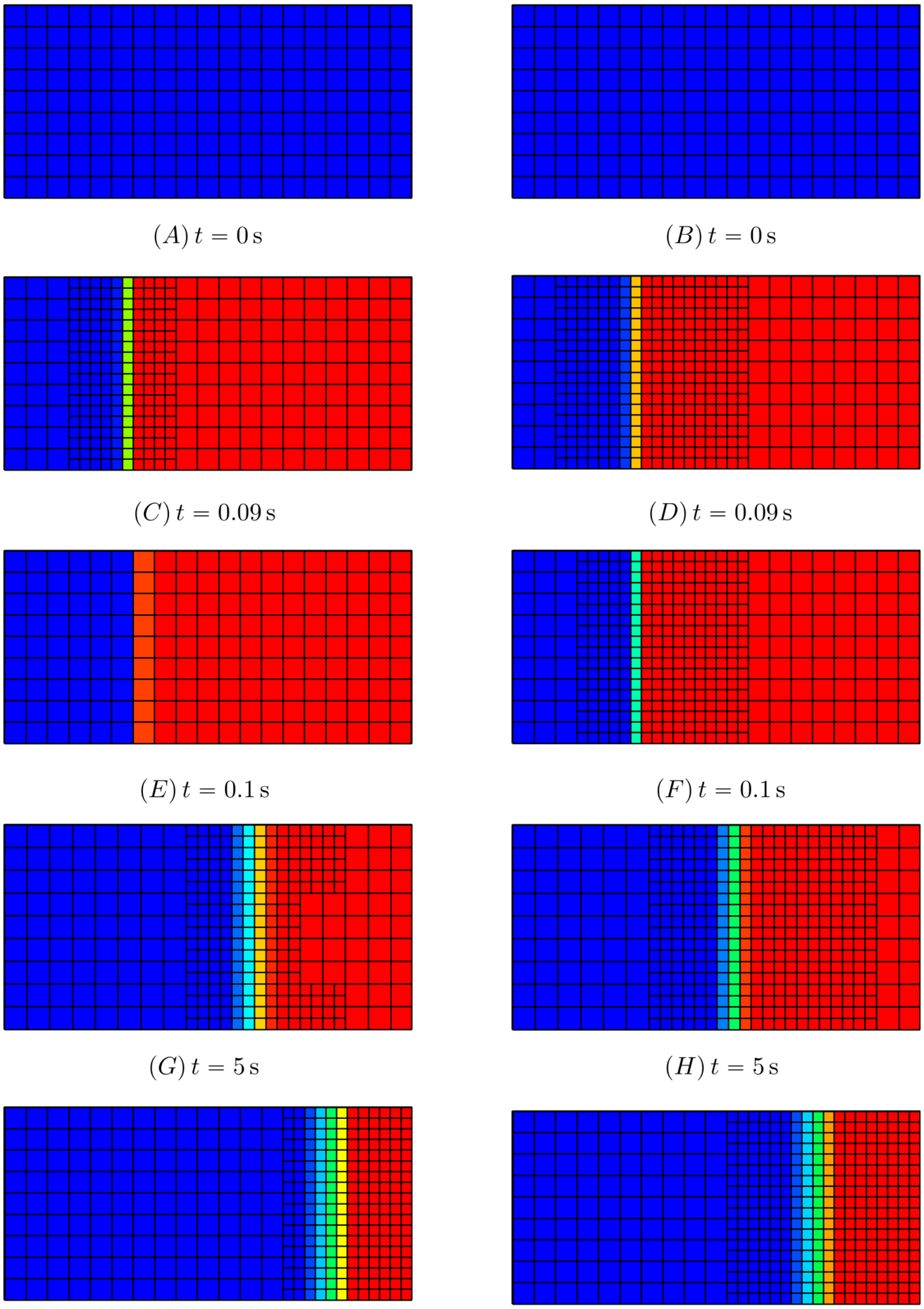
Figure 4.11. Stefan problem with volume expansion ($R_\rho = 0.185$): Pressure distribution along the length of the channel and zoomed-in views for the liquid and solid domains at (A) $t = 5$ s and (B) $t = 10$ s. (C) Plot of the numerical pressure jump ($[\![-p]\!] = p^{\rm S} - p^{\rm L}$) across the interface as a function of liquid velocity u^L. The dashed line has a slope of value $A_d\Delta$, in which the Carman-Kozeny drag coefficient A_d is computed using a solid fraction value of $\varphi_{\rm S} = 0.833$. We remark that for the purposes of this plot only the temperature interval is taken to be $\Delta T = 60$ K. This is done to obtain a relatively smoother $[\![-p]\!]$ versus $u^{\rm L}$ curve. Using $\Delta T = 10$ K, pressure jump across the mushy region exhibited larger oscillations as a function of fluid velocity. The grid size and time step size used for this case are $N_x \times N_y = 1280 \times 64$ and $\Delta t = 10^{-4}$ s, respectively.

region during phase change. We use the Stefan problem with expansion from above section using adaptively refined AMR grids to check the robustness of capturing the mushy region with AMR.

The Stefan problem is simulated by considering a quasi one-dimensional computational domain $\Omega \in [0, 1] \times [0, 0.05]$ with $l = 2$ mesh levels and a refinement ratio of $n_{\text{ref}} = 2$. The coarsest level ($l = 1$) is covered by $N_{x0} \times N_{y0} = 640 \times 32$ number of cells. The simulation runs until $t = 10$ s with a constant time step size of $\Delta t = 10^{-4}$ s. The other problem statement and the thermo-physical properties are same as Sec. 4.6.1. We use this problem to analyze the performance of both tagging methods (φ-based vs. $\nabla\varphi$-based) to identify the mushy region effectively within the AMR framework.

The position of the moving (from left to right) solidification front is denoted $x^* = s(t)$. The numerical solid–liquid interface fraction is defined by the iso-contour value 0.5 of the liquid fraction φ. Fig. 4.12 shows the evolution of φ over time t using two different tagging methods: φ-based (left) and $\nabla\varphi$-based (right). It can be seen that the former tagging method produces intermittent fine mesh patterns. This can be explained as follows. There are instances during the simulation in which the mushy region narrows and falls into the subgrid level. As a result, no cells are tagged using the φ-based tagging method. This results in the simulation removing the entire fine grid level $l = 2$. Fig. 4.12 (E) illustrates this effect, with φ either 0 or 1 and not meeting the value-based refinement criteria. Gradient-based tagging, on the other hand, produces finer meshes near the liquid-solid interface. Despite the narrowness of the mushy region, the gradient of liquid fraction always exists.

Fig. 4.13 shows a comparison between the time evolution of the solidification front and the analytical solution. As a result of uneven appearance and disappearance of the fine mesh level, φ-based tagging overpredicts the front position. Alternatively, $\nabla\varphi$-based tagging produces more accurate results since the solidifying front is always resolved by finer meshes. Accordingly, we select the gradient-based tagging criteria to capture the moving phase change front in our simulations.



$(A)\, t = 0\,\mathrm{s}$

$(B)\, t = 0\,\mathrm{s}$

$(C)\, t = 0.09\,\mathrm{s}$

$(D)\, t = 0.09\,\mathrm{s}$

$(E)\, t = 0.1\,\mathrm{s}$

$(F)\, t = 0.1\,\mathrm{s}$

$(G)\, t = 5\,\mathrm{s}$

$(H)\, t = 5\,\mathrm{s}$

$(I)\, t = 10\,\mathrm{s}$

$(J)\, t = 10\,\mathrm{s}$

Figure 4.12. Stefan problem with solidification: The evolution of the liquid fraction φ over a time t. The solidification happens due to the imposed Dirichlet boundary condition on the left. The evolution of grids is shown for φ based AMR (left) and $\nabla\varphi$ based AMR (right). The solid and liquid regions are depicted by yellow and blue colors, respectively. The region between the solid and liquid regions is the mushy region.

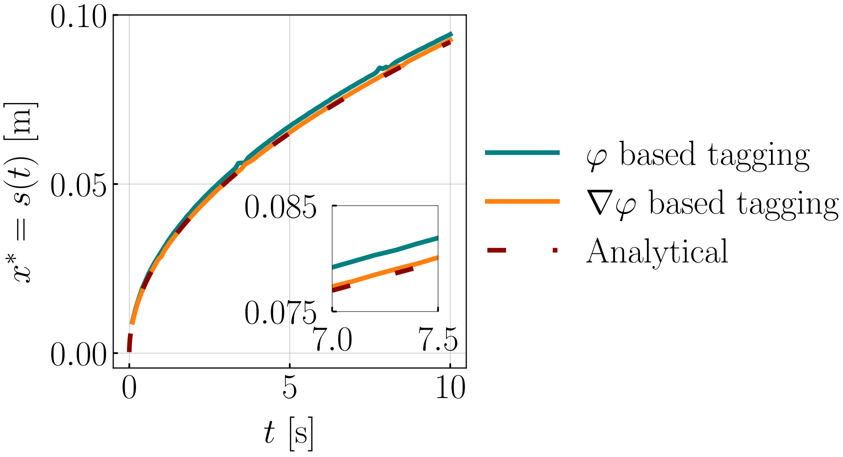


Figure 4.13. Comparison of numerical interface position obtained using "φ based AMR" and "$\nabla\varphi$ based AMR" with analytical solution for density ratio $R_\rho = 0.185$.

4.6.2 Stefan problem with melting

Previous studies have primarily relied on experimental results to validate the effects of laser beams. However, numerical results were in good agreement with experimental data after enhancing thermophysical properties using an enhancement factor. In this section, we utilize the analytical solution of the two-phase Stefan melting problem to validate the influence of the laser force. In Sec. 4.3, we provided the analytical solutions of two-phase Stefan solidification problem with volume change. Building upon that, we derive analytical solutions for the two-phase Stefan melting problem with volume change, presented in Appendix 6.8.

We explore two cases in this section:

- Two-phase Stefan melting problem with a flux boundary condition.
- Three-phase Stefan melting problem with flux boundary condition is supplied to Eq. (4.44) as heat flux.

The thermophysical properties of the iron-based PCM are given in Table 4.2, with the density of the liquid phase being 8100 kg/m^3 and the solid phase being 7165.384 kg/m^3.

Two-phase Stefan melting problem

Consider a quasi one-dimensional computational domain $\Omega \in [0, 0.1] \times [0, 0.005]$, as shown in Fig. 4.14(A). The simulation employs a two-level mesh with a coarsest grid size of $\Delta x = \Delta y = 0.005/32$. The simulation runs untill $t = 10s$ with a constant time step size of $\Delta t = 10^{-4}$. Initially, the domain comprises only the solid phase. On the right boundary ($x = 0.1$), a zero pressure/outflow condition for the velocity $\mathbf{u}$ and an adiabatic (homogeneous Neumann) condition for the temperature T are applied, while on the left wall ($x = 0.1$), a no-slip condition and inhomogeneous Neumann boundary condition $-k\frac{\mathrm{d}T^{\mathrm{L}}_{\mathrm{exact}}}{\mathrm{d}x}\Big| x = 0$ are utilized, where T^{L}exact represents the analytical temperature distribution derived in Appendix 6.8. The top and bottom boundaries are periodic.

The schematic of this problem is depicted in Fig. 4.14(A). Since $\mathbf{R}_\rho = \rho^{\mathrm{L}}/\rho^{\mathrm{S}} < 1$, there is a volume expansion, inducing additional flow. As shown in Appendix 6.8, the solid moves with the rigid body velocity $u^{\mathrm{S}}(t)$, while the liquid velocity u^{S} is zero. To simulate this behavior, the solid fraction φ_{rmS} in the Carman-Kozeny force is replaced with the liquid fraction φ. Consequently, the Carman-Kozeny force retards any flow in the liquid region. The "$\nabla\varphi$ based tagging" is used for the grid refinement. Melting initiates due to the flux boundary condition and progresses towards the right over time until the solid completely liquefies. The evolution of the liquid fraction and the adaptive mesh refinement (AMR) mesh is shown in the first column of Fig.4.14. It is observable that the interface (defined as the 0.5 contour of φ) moves towards the right as the solid phase converts into the liquid phase.

The simulation is performed on 2 levels mesh with coarsest grid size of $\Delta x = \Delta y = 0.005/32$. The simulation is performed till $t = 10s$ with constant time step size $\Delta t = 10^{-4}$. Initially, the domain consists of only solid phase. On the right boundary is ($x = 0.1$), we use zero pressure/outflow and adiabatic (homogeneous Neumann) boundary condition while on the left wall ($x = 0.1$), no-slip and inhomogeneous Neumann $-\kappa^{\mathrm{L}}\frac{\mathrm{d}T^{\mathrm{L}}_{\mathrm{exact}}}{\mathrm{d}x}\Big|_{x=0}$ is used where $T^{\mathrm{L}}_{\mathrm{exact}}$ is the analytical temperature distribution. The analytical solution for this problem is derived in Appendix A. The top and bottom boundaries are periodic. The schematic of this problem is shown in Fig. 4.14(A). Since $\mathbf{R}_\rho = \rho^{\mathrm{S}}/\rho^{\mathrm{L}} > 1$, there is a volume expansion and this will induce additional flow. As shown in Appendix A, the solid moves with the rigid body velocity $u^{\mathrm{S}}(t)$ and the liquid velocity u^{S} is zero. To simulate this behavior, the solid fraction φ_{rmS} in the Carman-Kozeny force is replaced with the liquid fraction φ. Now, the Carman-Kozeny will retard any flow in the liquid region. The $\nabla\varphi$ based tagging is used for the grid refinement. The melting starts due to the flux boundary condition and it progresses towards the right over time. This will continue till the solid completely liquidifies. We have shown the the evolution of the liquid fraction and the AMR mesh in the first column of the Fig. 4.14. We can see that the interface (defined as 0.5 contour of φ) moves towards right while the solid phase converts into the liquid phase.

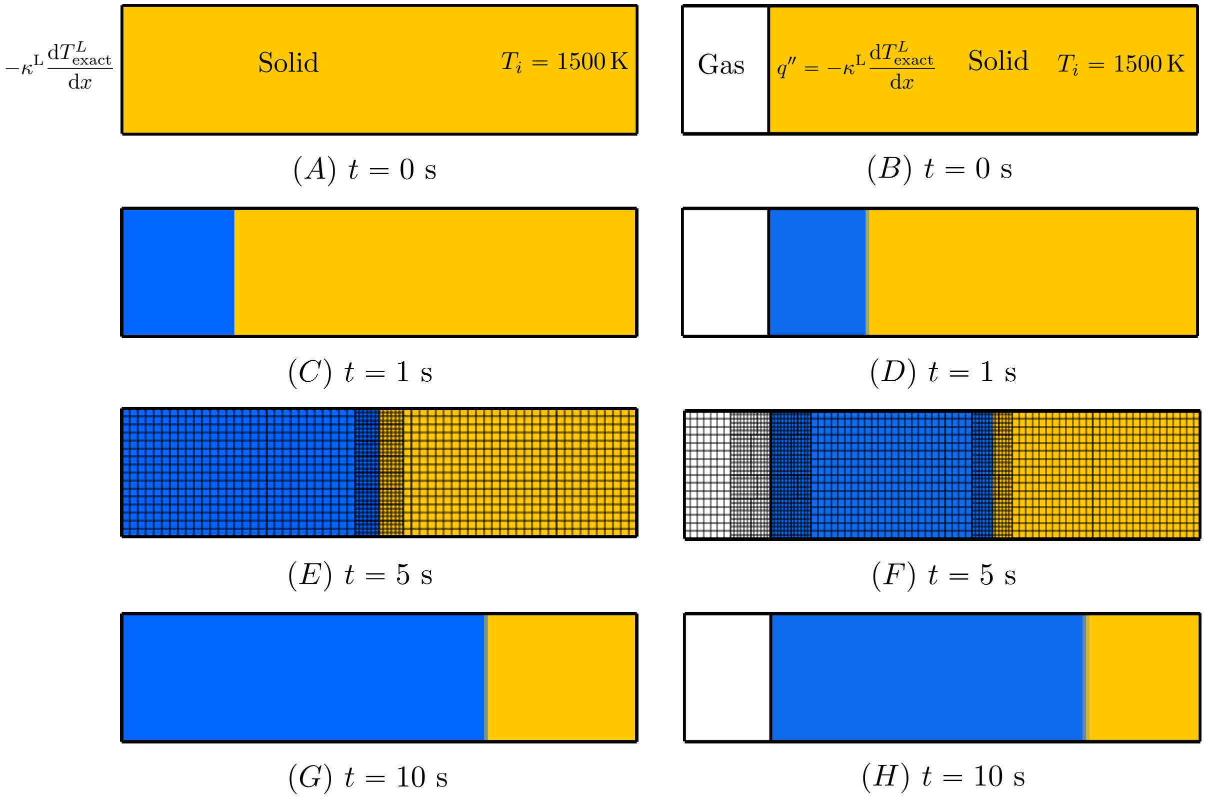
We also compare the numerical solutions with the analytical solutions provided in Appendix 6.8. Comparisons for the interface position, solid velocity, and temperature distribution are shown in Fig. 4.15. A very good match is observed between the numerical and the analytical results.

Table 4.2. Thermophysical properties used to simulate the Stefan melting problem

Property	Value
Thermal conductivity of solid κ^{S}	22.9 W/m.K
Thermal conductivity of liquid κ^{L}	22.9 W/m.K
Specific heat of solid C^{S}	627 J/kg.K
Specific heat of liquid C^{L}	723.14 J/kg.K
Solidification temperature T_m	1620 K
Bulk phase change temperature T_r	1620 K
Liquidus temperature T^{liq}	1622.5 K
Solidus temperature T^{sol}	1617.5 K
Latent heat L	250800 J/kg

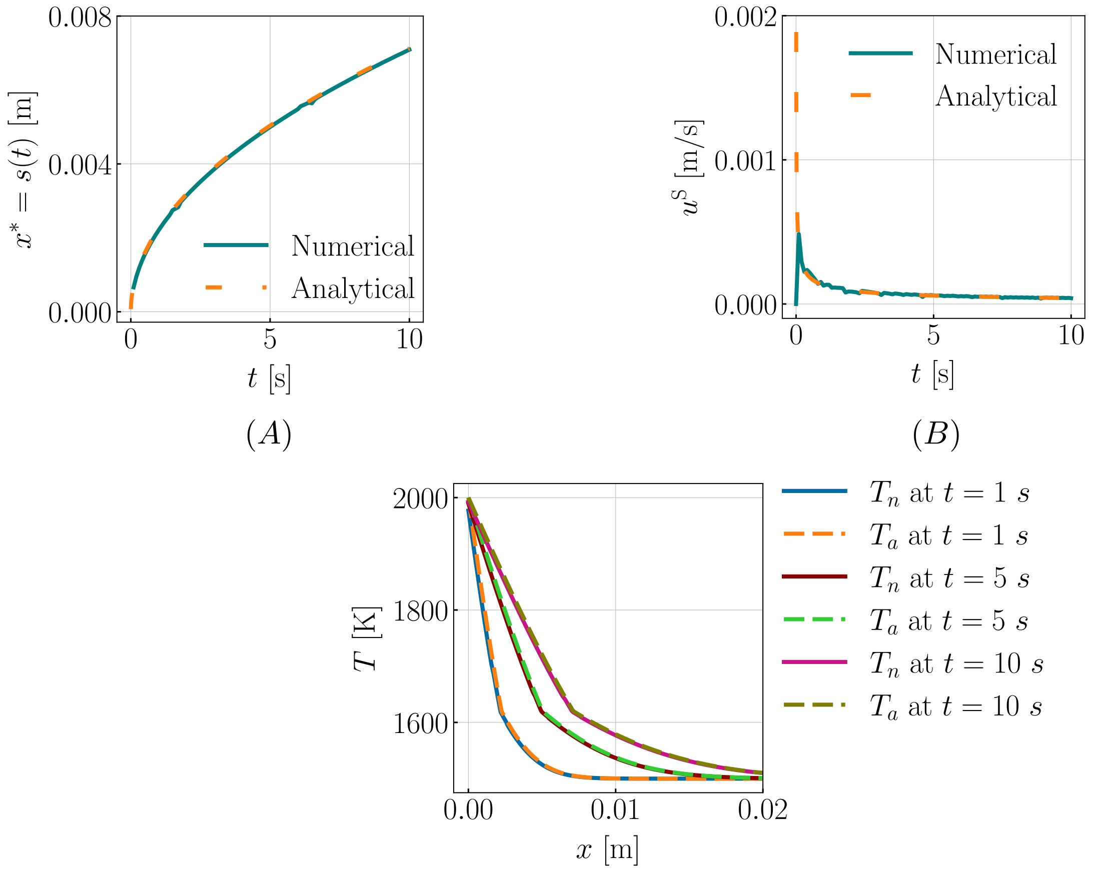
Three-phase Stefan melting problem

We now consider a three-phase melting problem with flux boundary conditions supplied to the energy equation as heat flux. This problem is employed to validate the implementation



$(A)\ t = 0$ s
$(B)\ t = 0$ s
$(C)\ t = 1$ s
$(D)\ t = 1$ s
$(E)\ t = 5$ s
$(F)\ t = 5$ s
$(G)\ t = 10$ s
$(H)\ t = 10$ s

Figure 4.14. Stefan problem with melting: The evolution of the liquid fraction φ over a time t. The melting happens due to the imposed flux boundary condition on the left boundary for the first column plots. On the right plot, the flux boundary condition is imposed through the source term in the energy equation. The evolution of grids is also shown for using $\nabla\varphi$ based AMR. The solid, liquid and gas regions are depicted by yellow, blue and white colors, respectively. The region between the solid and liquid regions is the mushy region.



(A) (B)

(C)

Figure 4.15. Comparison of CFD and anlalytical solutions for Two phase Stefan melting problem with flux boundary conditions: (A) Interface position (B) Solid velocity and (C) Temperature.

of the heat source (Eq. (4.44)). While many studies have validated the laser beam flux using experimental results with an enhancement factor, it's important to note that the heat source that melts the material acts as a surface force, typically included in the energy equation using the CSF method. This force is often represented as a surface heat flux, commonly modeled with a Gaussian profile. Thus, we solve the three-phase Stefan problem with a heat flux boundary condition.

The thermophysical properties remain the same as in the previous case. The problem setup involves the gas and solid phases initially, as illustrated in Fig. 4.14(B). The PCM region is identical to the two-phase problem: $\Omega_{\text{PCM}} \in [0, 0.1] \times [0, 0.005]$. The gas region $\Omega_{\text{g}} \in [-0.02, 0] \times [0, 0.005]$ is attached to the left side of the PCM computational domain.

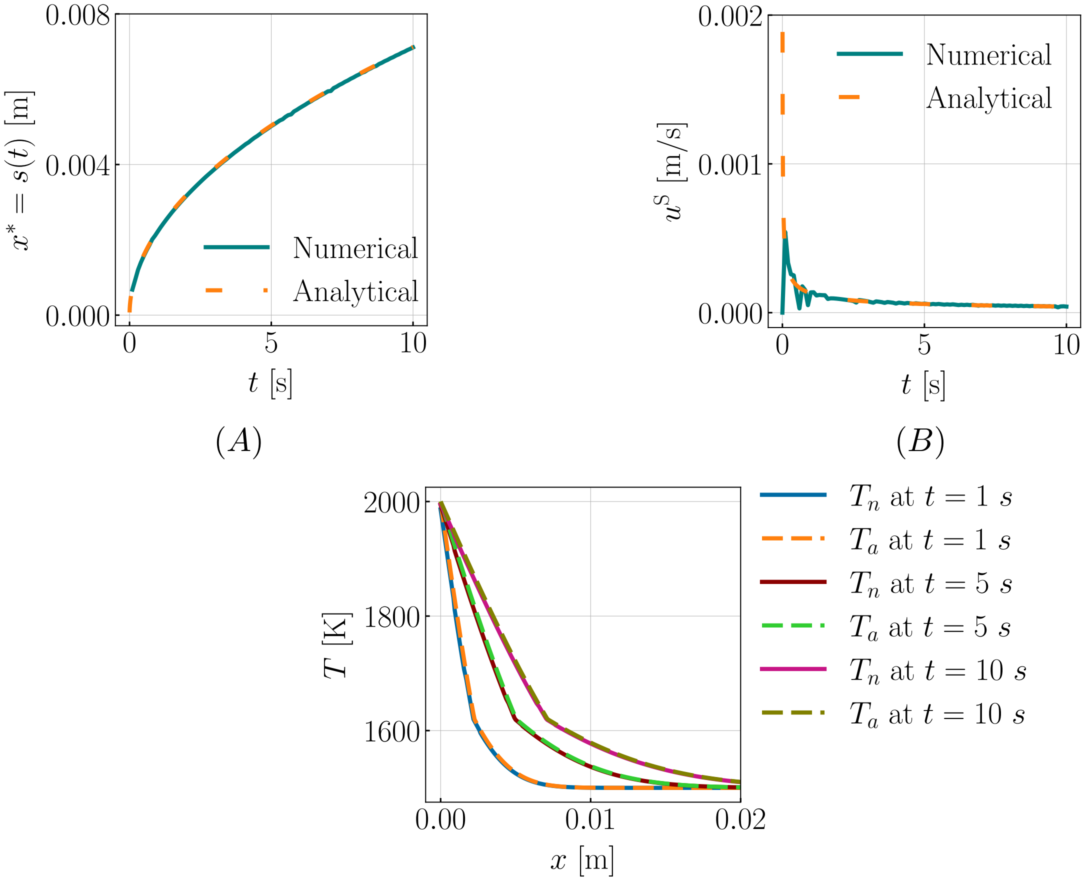
Boundary conditions are set as follows: periodic at the top and bottom, zero pressure/outflow for velocity, and adiabatic (homogeneous Neumann) for temperature on the left and right boundaries. The flux boundary condition $q'' = -k \frac{\mathrm{d}T^{\text{L}}_{\text{exact}}}{\mathrm{d}x}\Big|_{x=0}$ is applied to the energy equation forcing term (Eq. (4.44)).

Gas properties are specified as $\rho_{\text{g}} = 0.4$ kg/m^3, $k_{\text{g}} = 6.1 \times 10^{-2}$ W/m·K, and $c_{\text{pg}} = 1100$ J/kg·K. Viscosity is assumed to be zero in all three phases. A two-level AMR mesh with a coarse grid size of $0.005/32$ cells is utilized for simulation. Tagging is based on ϕ for the PCM-gas interface ($x = 0$) and on $\nabla\varphi$ for refining the liquid-solid interface. A smooth Heaviside function H with $n_{\text{smear}} = 2$ is employed.

Due to the flux boundary condition, melting initiates and progresses toward the right, as illustrated in the right plots of Fig.4.14. Comparison between numerical and analytical solutions is presented in Fig.4.16. It is evident that the numerical and analytical solutions exhibit excellent agreement.

4.6.3 Isothermal advection of a dense bubble

To demonstrate the combined consistency of mass, momentum, and enthalpy integrators, we modify the dense bubble advection problem discussed in Sec. 4.4.3: in addition to mass and



(A)

(B)

(C)

Figure 4.16. Comparison of CFD and anlalytical solutions for Three phase Stefan melting problem with heat source to the energy equation: (A) Interface position (B) Solid velocity and (C) Temperature.

momentum transport, we also consider enthalpy transported by the bubble. Thermal conductivity κ and Q_{src} are set to zero. All other problem parameters (e.g., bubble size, boundary conditions, etc.) remain the same as in Sec. 4.4.3. In the absence of thermal diffusion and body forces, the system of equations reduces to

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{4.81a}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{4.81b}$$

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) = 0, \tag{4.81c}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = 0. \tag{4.81d}$$

The initial temperature of the liquid bubble and the surrounding gas is set to $T_i = 3$ K [11] and the fusion/phase change temperature of the liquid bubble is set to $T_m = 2$ K. The liquidus and solidus temperatures are $T^{\text{liq}} = 2.1$ K and $T^{\text{sol}} = 1.9$ K, and a latent heat value of $L = 100$ J/kg is used. For this problem we expect that:

- In the absence of body forces (pressure and viscous) the velocity should remain constant throughout the simulation.
- The bubble gets advected with constant velocity, retains its shape, and returns to its original position after each time period.
- With no energy sources and non-conducting fluids, the enthalpy and temperature should remain constant.
- There should be no phase change or spurious phase generation. This also means that specifying latent heat value is arbitrary for this problem.
- The total mass, momentum, and enthalpy in the domain should remain constant.

---

[11] For this test problem, units of physical quantities are not relevant, but are included to maintain consistency.

Fig. 4.17 illustrates the advection of the dense bubble at various time instances. The simulation employs $l = 2$ levels of mesh with the coarse grid size set to $N_x \times N_y = 128^2$, and a refinement ratio is $n_{\text{ref}} = 2$. The simulation runs until $t = 10$ s. As shown in Fig. 4.17, the bubble moves with a constant velocity and returns to its original position after one time period $\tau = \frac{d}{|\mathbf{u}|} = 1$ s, in which $d = \sqrt{2}$ and $|\mathbf{u}| = \sqrt{2}$. Additionally, the bubble undergoes minimal distortion while returning to its original position. Throughout the simulation (data not shown here for brevity), velocity remains constant at its initial value $\mathbf{u} = (u, v) = (1, 1)$. In addition, the enthalpy equation solution does not result in spurious phases (for example, liquid becoming solid).

To quantify the accuracy of consistent mass-momentum-enthalpy integrators, a grid convergence study is conducted. We consider uniform grids of size $N_x \times N_y = 64^2, 128^2, 256^2$ for the convergence study. This is to avoid discretization errors near the coarse-fine boundary of adaptively refined grids. Errors in total mass $\mathcal{M} = \int_\Omega \rho \, \mathrm{dV}$, x-momentum $\mathcal{L}_x = \int_\Omega \rho u \, \mathrm{dV}$, y-momentum $\mathcal{L}_y = \int_\Omega \rho v \, \mathrm{dV}$ and enthalpy $\mathcal{H} = \int_\Omega \rho h \, \mathrm{dV}$ are computed using Eq. (4.64), and the results are shown in Fig. 4.18. The system's mass, momentum, and enthalpy remain constant over time, with the exception of the initial transient phase where the initially sharp interface diffuses. Additionally, the temperature remains mostly constant throughout the simulation ($T \approx 2$ K) except near the interface (data not shown for brevity). This is because the primary variable h diffuses over time and T is reconstructed based on T-h relations in our solution algorithm. The temperature change near the interface does not induce phase change, however. Mass and momentum are at least third-order accurate ($\mathcal{O}(\Delta x^3)$) using this method. As velocity remains constant (numerically), mass and momentum errors are the same. However, enthalpy solution is only first-order ($\mathcal{O}(\Delta x)$) accurate. The behavior of error as a function of grid resolution can be explained as follows. At the end of each time step, the density field is synchronized with the level set function. This limits density (and momentum, in this case) diffusion due to advection. The level set function is reinitialized to a signed distance function using a third-order accurate ENO scheme; see our prior works on level set reinitialization [113]. Enthalpy transport, however,

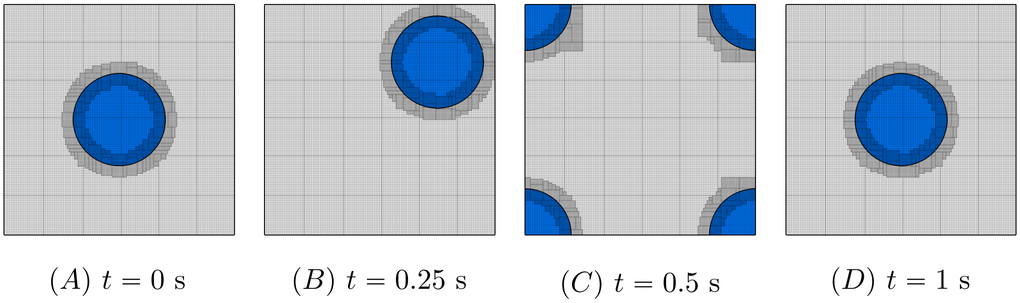

$(A)\ t = 0$ s
$(B)\ t = 0.25$ s
$(C)\ t = 0.5$ s
$(D)\ t = 1$ s

Figure 4.17. Advection of a dense liquid bubble with density ratio $\rho_i/\rho_o = 10,000$ in an initially uniform velocity field $\mathbf{u} = (u, v) = (1, 1)$. The simulation is performed with two levels of mesh refinement. The coarse grid size is $N_x \times N_y = 128^2$. A unit periodic domain is considered. κ, μ and Q_{src} are set to zero.

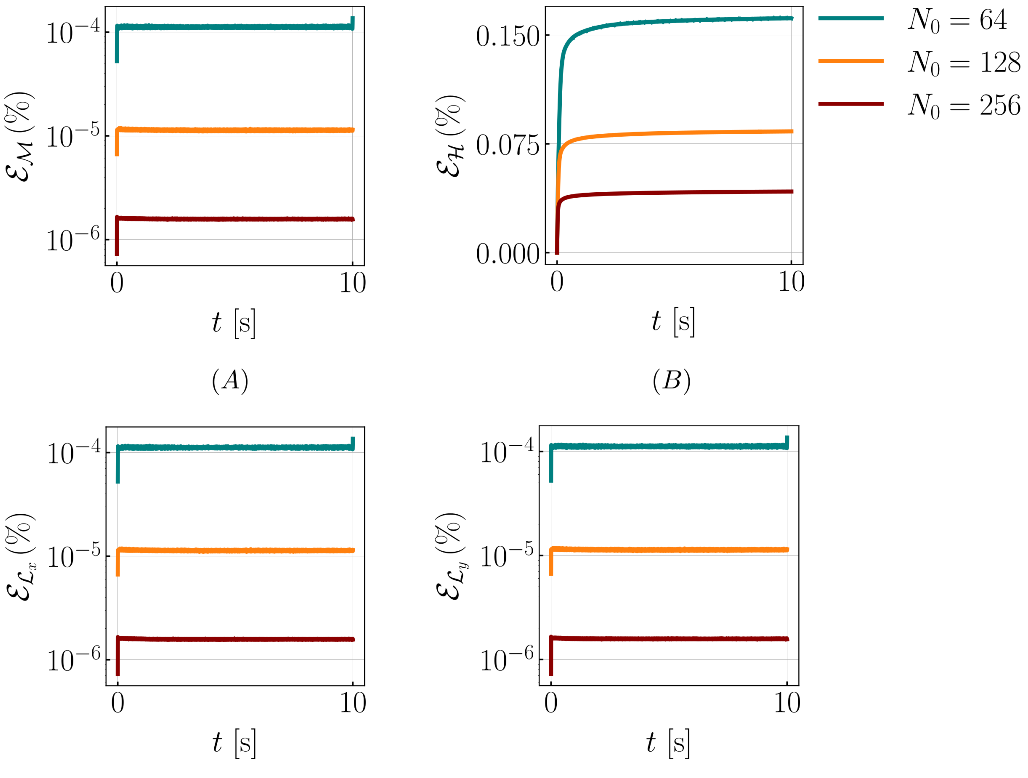
is subject to the upwinding errors of the CUI limiter, which displays first-order accuracy near non-monotone regions. These are regions close to the liquid-gas interface. The CUI scheme's first-order accuracy corresponds to Gudonov's order barrier theorem.

These results indicate that our mass, momentum, and enthalpy integrators are capable of stably simulating high density ratio flows without generating spurious momentum or phases.

4.6.4 Thermocapillary flows

Here we solve enthapy Eq. (4.34) in conjunction with surface tension Eq. (4.47b) to simulate surface tension-driven flows. Specifically, the thermocapillary migration of gas bubbles is considered. In most thermocapillary flow models in the literature, the temperature equation is used instead of the enthalpy equation [120, 117, 121, 122, 123]. Using an enthalpy equation instead of temperature does not matter on a continuous level, but discretely, it may. The purpose of this test is to test whether the low Mach enthalpy integrator can accurately simulate thermocapillary flows.

Consider a computational domain $\Omega \in [0, L]^d$, $d = 2$ and 3, within which a bubble of radius R is immersed in the ambient fluid, as illustrated in Fig. 4.19. The bubble's initial centroid is in the middle of the domain. The ratio of gas bubble thermophysical properties (μ, ρ, C, κ) to the surrounding fluid is denoted γ. The top and bottom boundaries are considered walls



(A) (B)

(C) (D)

Figure 4.18. Grid convergence study of mass $\mathcal{M} = \int_\Omega \rho \,\mathrm{dV}$, x-momentum $\mathcal{L}_{\mathbf{x}} = \int_\Omega \rho u \,\mathrm{dV}$, y-momentum $\mathcal{L}_{\mathbf{x}} = \int_\Omega \rho v \,\mathrm{dV}$ and enthalpy $\mathcal{H} = \int_\Omega \rho h \,\mathrm{dV}$ for the isothermal advection of a bubble.

with no-slip (zero-velocity) boundary conditions. Temperature varies linearly in the domain $T_{\text{low}} \leq T \leq T_{\text{high}}$. Here, T_{low} and T_{high} represent temperatures at the bottom and top walls, respectively. All other boundaries are periodic.

The surface tension coefficient varies with temperature as

$$\sigma = \sigma_0 + \left.\frac{\mathrm{d}\sigma}{\mathrm{dT}}\right|_{T_0} (T - T_0), \tag{4.82}$$

in which σ_0 and $\left.\frac{\mathrm{d}\sigma}{\mathrm{dT}}\right|_{T_0}$ are the surface tension and Marangoni coefficients (respectively) computed at the reference temperature T_0. The gas bubble moves upward due to the variation in surface tension caused by the imposed temperature gradient. Key non-dimensional parameters of the problem are the Reynolds number $Re = \frac{\rho^{\mathrm{L}} V R}{\mu^{\mathrm{L}}}$, Marangoni number $Ma = \frac{VR}{\alpha^{\mathrm{L}}}$, Capillary number $Ca = \frac{V \mu^{\mathrm{L}}}{\sigma_0}$, and Prandtl number $Pr = \frac{\mu^{\mathrm{L}} C^{\mathrm{L}}}{\kappa^{\mathrm{L}}}$. Here, V is the velocity scale defined as $V = \left| \frac{\left.\frac{\mathrm{d}\sigma}{\mathrm{dT}}\right|_{T_0} \nabla T R}{\mu^{\mathrm{L}}} \right|$. Two different cases are considered: zero Marangoni and finite Marangoni. The zero Marangoni case checks the accuracy of the surface tension force model. The finite Marangoni case examines the low Mach enthalpy integrator's ability to simulate thermocapillary flows.

Zero Marangoni number

A time-invariant linear temperature field ($\kappa \to \infty$) is considered, which is referred to as the "zero Marangoni number" problem in the literature. Young et al. [161] investigated the zero Marangoni number thermocapillary migration of a bubble in ambient fluid. The authors provide an analytical expression for the terminal velocity of the bubble which reads as

$$V_t = -\frac{2 \left.\frac{\mathrm{d}\sigma}{\mathrm{dT}}\right|_{T_0} \nabla T R}{6\mu^{\mathrm{L}} + 9\mu^{\mathrm{G}}} \tag{4.83}$$

We compare the numerically computed bubble rise velocity with the analytical solution to assess the surface tension force model accuracy. The normalized bubble rise velocity is numerically

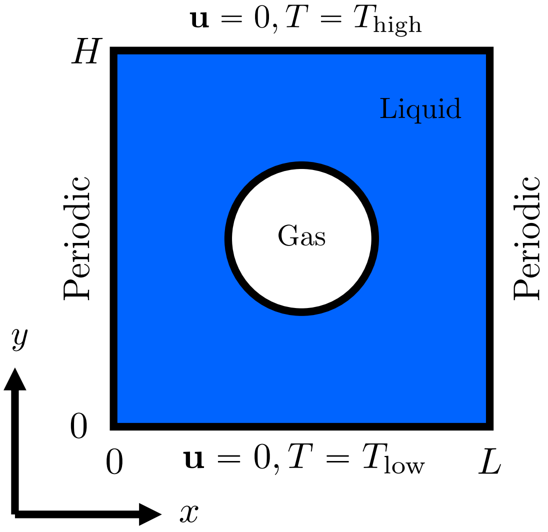
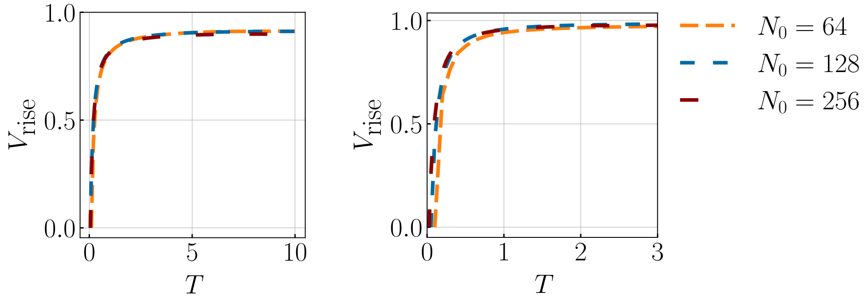


Figure 4.19. Schematic of the thermocapillary migration of a gas bubble in 2D. A gas bubble of radius R rises in an ambient fluid due to gradient in the surface tension coefficient. The temperature varies linearly $T_{\text{low}} \leq T \leq T_{\text{high}}$ with bottom and top wall temperatures of T_{low} and T_{high}, respectively. No-slip boundary conditions are used for velocity at the top and bottom walls. Periodic boundary conditions are considered for the vertical walls.

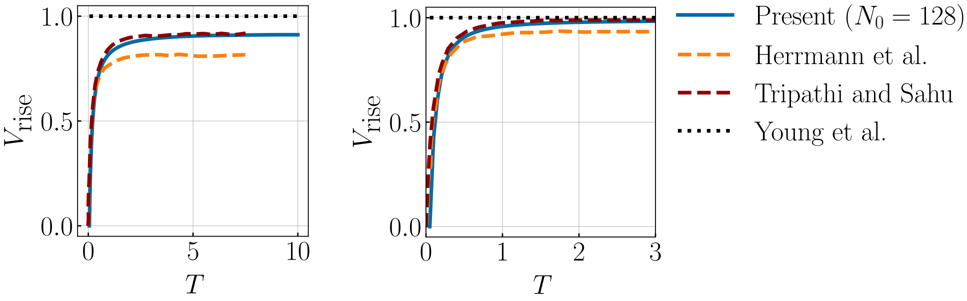
calculated as

$$V_{\text{rise}} = \frac{\int_\Omega v(1-H)\text{d}V}{V_t \int_\Omega (1-H)\text{dV}}. \tag{4.84}$$

Due to the time-invariant nature of the linear temperature field, we do not solve the enthalpy equation, since it would change the temperature field numerically, even slightly. The radius of the bubble is taken to be $R = 0.5$ m. The ratio of thermophysical properties of the gas and ambient fluid is set to $\gamma = 1$. In this case, the density and viscosity of the bubble (and also the liquid) are 0.2 kg/m^3 and 0.1 kg/m·s. The square and cubic computational domains are of length $L = 15R$ in 2D and 3D, respectively. The surface tension parameters are $\sigma_0 = 0.1$ N/m and $\frac{\text{d}\sigma}{\text{dT}}\big|_{T_0} = -0.1$ N/m·K. Temperatures at the bottom and top walls are $T_{\text{low}} = 0$ K and $T_{\text{high}} = 1$ K. This corresponds to a temperature gradient of $\dfrac{\partial T}{\partial y} = 0.133$ K/m. These dimensional values give us $V = Re = Ca = 6.67 \times 10^{-2}$ and $Ma = Pr = 0$. Grid convergence studies are performed using three grid resolutions: coarse, medium and fine. These grids have a coarse level ($l = 1$) resolution of N_0 = 64, 128 and 256, respectively. Two levels of grid refinement



(A) 2D grid convergence study (B) 3D grid convergence study



(C) 2D comparison (D) 3D comparison

Figure 4.20. Normalized rise velocity of a drop for Ma = 0 case: Grid independence study for (A) 2D and (B) 3D. Comparison of present numerical result with other numerical and theoretical results for (C) 2D and (D) 3D.

with a refinement ratio of $n_{\text{ref}} = 2$ are employed for all three grids. A uniform time step size of $\Delta t = 0.1$ s is used for the coarse grid. Δt is halved for each successively refined mesh. The top row of Fig. 4.20 shows the time evolution of normalized rise velocity for the three grids in both two and three spatial dimensions. Convergent behavior for rise velocity can be observed in both 2D and 3D simulations. Consequently, we choose the medium grid (with $N_0 = 128$) for comparing our results with prior studies [117, 161, 123]. From Figs. 4.20 (C) and (D) it can be observed that our results for the zero Marangoni case are in reasonable agreement with the literature.

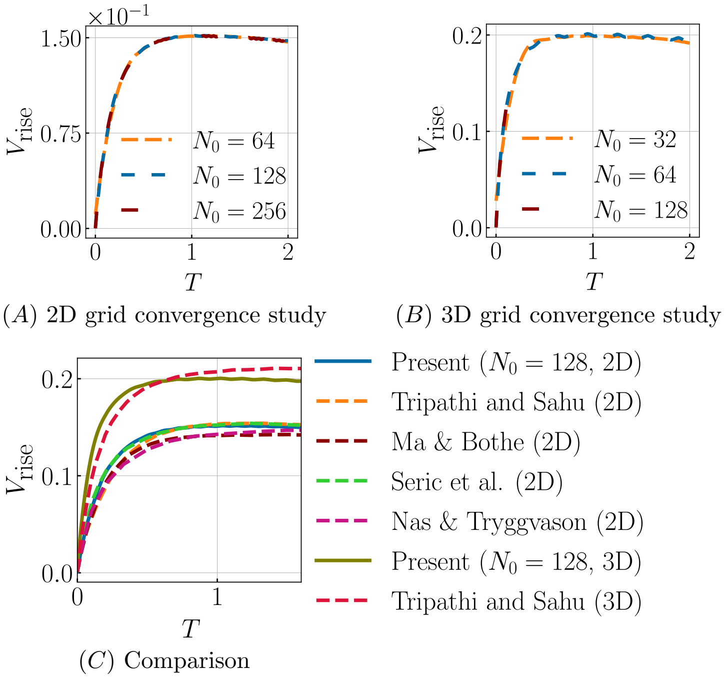
Results from the zero Marangoni case validate the accuracy of the surface tension force model as presented in Eq. (4.47b). However, it does not confirm the accuracy of the low Mach enthalpy integrator for simulating thermocapillary flows. Next, a finite Marangoni case is considered to verify this.

Finite Marangoni number

We investigate the thermocapillary migration of the gas bubble at a finite Marangoni number of $Ma = 0.72$. Here, the temperature varies with time, and the Navier-Stokes and enthalpy equations are coupled to each other through the advection term in the enthalpy equation and the surface tension force in the momentum equation. This finite Marangoni case ($Ma = 0.72$) has been considered in several prior works as well [120, 121, 122, 123] for validation purposes.

The drop in our simulations has a radius of $R = 1.44 \times 10^{-3}$ m and is initially positioned at the center of the computational domain. Square and cubic computational domains with length $L = 4R$ are used for 2D and 3D simulations, respectively. The thermophysical properties of the ambient liquid are $\rho^{\mathrm{L}} = 500$ kg/m^3, $\mu^{\mathrm{L}} = 0.024$ kg/m·s, $\kappa^{\mathrm{L}} = 2.4 \times 10^{-6}$ W/m·K, and $C^{\mathrm{L}} = 10^{-4}$ J/kg·K. The ratio of gas bubble thermophysical properties to the surrounding fluid is set to $\gamma = 0.5$. The surface tension parameters are $\sigma_0 = 10^{-2}$ N/m, $\frac{\mathrm{d}\sigma}{\mathrm{dT}}\big|_{T_0} = -2 \times 10^{-3}$ N/m·K. Temperatures at the bottom and top walls are $T_{\text{low}} = 289.424$ K and $T_{\text{high}} = 290.576$ K. This corresponds to a temperature gradient of $\dfrac{\partial T}{\partial y} = 200$ K/m. The phase change/fusion temperature of the ambient liquid is set to T_m = 265 K, with $T^{\text{liq}} = 270$ K, $T^{\text{sol}} = 260$ K, $T_{\text{ref}} = 290$ K. For a consistent enthalpy solver, these temperature values should not induce solidification of the liquid as the imposed temperature values are higher than the fusion temperature. The dimensional values of the problem parameters yield $Re = Ma = 0.72$, $Ca = 0.0576$, $V = 0.024$ and $Pr = 1$.

We solve the coupled Navier-Stokes and enthalpy equations to find the approximations for velocity ($\mathbf{u}$) and specific enthalpy (h) until $t = 0.12$ s. First, we perform grid convergence studies for both 2D and 3D simulations. The grids considered are $N_0 = 64, 128$, and 256 with two levels of mesh refinement and $n_{\text{ref}} = 2$. A uniform time step size of $\Delta t = 2.5 \times 10^{-5}$ is employed for the coarsest grid ($N_0 = 64$), with the time step size quadrupled for successive grids. For 3D simulations, grids of $N_0 = 32, 64$, and 128 are considered, with the coarsest grid time step size set to $\Delta t = 2 \times 10^{-5}$. The time step size is halved for each finer grid.



(A) 2D grid convergence study (B) 3D grid convergence study

(C) Comparison

Figure 4.21. Normalized rise velocity of a bubble for $\mathrm{Re} = \mathrm{Ma} = 0.72$ case: Grid independence study for (A) 2D simulation and (B) 3D simulation. (C) Comparison of present 2D and 3D numerical result with other numerical results.

The results are presented in Fig. 4.21 (A) for the 2D simulation and Fig. 4.21 (B) for the 3D simulation. It is evident that the numerical rise velocity is consistent across all grid resolutions for both 2D and 3D simulations. For the sake of comparison with existing literature, we select the results obtained from medium grids.

Fig. 4.21 (C) illustrates the comparison of the numerical rise velocity with other studies for both 2D and 3D simulations. Remarkably, our results closely align with those of Tripathi and Sahu [123]. It is worth noting that the choice of time step size in this case is determined by the capillary time constraint, given by $\Delta t_\sigma = \sqrt{\frac{\rho^{\mathrm{L}}+\rho^{\mathrm{G}}}{4\pi\sigma}\Delta x^3}$ [119, 162].

4.6.5 Metal melting

As our next example[12], we simulate melting of aluminum metal with a free surface to highlight two salient features of the new low Mach enthalpy method: (**1**) the ability to

---

[12] This benchmark test is provided in IBAMR GitHub within the directory `examples/phase_change/ex3`.

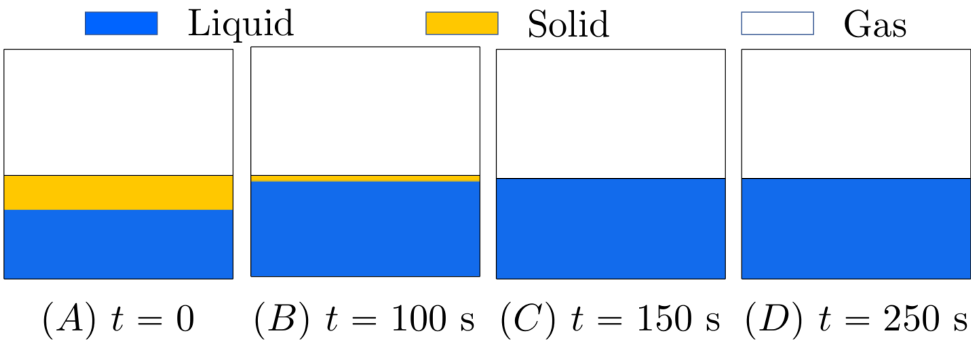


Figure 4.22. Time evolution of the solid, liquid, and gas domains during metal melting.

capture volume change effects of the PCM in the presence of gas phase; and (**2**) the ability to handle phase appearance or disappearance from the domain. This problem is inspired by Huang et al. [135] who solved a similar problem using a phase-field method (PFM), but using hypothetical[13] thermophysical properties of the PCM. The computational domain is considered to be a unit square $\Omega \in [0,1]^2$ that is discretized by $N \times N = 256 \times 256$ grid cells. At $t = 0$ the heavier liquid phase of density $\rho^{\rm L} = 2700$ kg/m^3 occupies the region below $y = 0.3$ m, and the lighter solid phase of density $\rho^{\rm S} = 2475$ kg/m^3 rests above the liquid phase and fills the domain until $y = 0.45$ m; see Fig. 4.22 (A). The gas occupies the remaining domain $(0.45 < y <= 1)$. Both liquid and solid phases are assumed to have the same viscosity $\mu^{\rm L} = \mu^{\rm S} = 1.4 \times 10^{-3}$ kg/m·s. Viscosity in the solid phase is fictitious and does not affect numerical results. The rest of the thermophysical properties for the liquid and solid phases are taken from Table 4.1. The thermophysical properties of the gas are based on air and are taken to be $\rho^{\rm G} = 0.4$ kg/m^3, $\kappa^{\rm G} = 6.1 \times 10^{-2}$ W/m·K, $C^{\rm G} = 1100$ J/kg·K and $\mu^{\rm G} = 4 \times 10^{-5}$ kg/m·s. In the x-direction, periodic boundary conditions are applied. On the top boundary $(y = 0)$, we use zero-pressure/outflow and no heat flux boundary conditions, while on the bottom wall $(y = 0)$, we use zero-velocity and fixed temperature boundary conditions $(T = 6T_m)$. $T_m = 933.6$ K is the melting temperature of aluminum. For the liquid region, the initial temperature is $5T_m$, whereas for the solid and gas regions, it is $0.9T_m$. The simulation is performed till $t = 250$ s with a uniform time step size of $\Delta t = 10^{-3}$ s.

---

[13] As mentioned earlier it is difficult to estimate/measure various material properties that is required in a PFM.

The solid melts when heat is transferred from the bottom wall to the liquid. As the melting process continues, the solid phase disappears after $t = 150$ s. To ensure that no spurious phase changes or interfacial dynamics exist, the simulation is continued until $t = 250$ s. As shown in panels (C) and (D) of Fig. 4.22, this is indeed the case. Since liquid density is larger than solid density, the gas-liquid interface position ($y = 0.4375$ m) at $t = 150$ s is lower than the initial gas-solid interface position ($y = 0.45$ m). It provides a simple "sanity check" for a CFD method that seeks to capture the volume change effect of the PCM; the final position of the gas-liquid interface indicates the success of the method. Additionally, the initial and final gas-metal interface locations can be used to quantify the percentage change in metal mass. It should ideally be zero. For the present simulation (using $256\times$ 256 grid), it is approximately $\mathcal{E} \approx 0.027\%$. Even though the percentage mass change is quite small, it is not near machine precision. This is attributed to two factors: (**1**) the non-conservative nature of the level set method, which is used to track the gas-PCM interface in our formulation; and (**2**) the errors incurred in computing the right-hand side of the low Mach Eq. (4.33) numerically. We show in Appendix Sec. 6.7 that $\mathcal{E}$ decreases with increasing grid resolution. It also decreases with decreasing time step size Δt (data not presented). We expect mass change errors would be lower if the level set method were replaced by a more conservative interface capturing technique like the volume of fluid technique. However, this needs to be verified.

4.6.6 Metal solidification

We consider the opposite scenario of the previous section in our final example: molten aluminum solidifying in a cast[14]. A common casting defect is solidification shrinkage. Metals that are denser in their solid form than in their liquid form (which is almost always the case) suffer from this defect. As liquid metal solidifies, its volume contracts due to the density contrast between phases. Macroscopically, the free surface of the metal is recessed down into the solidified metal, giving it the appearance of a pipe. The goal of this example is to demonstrate

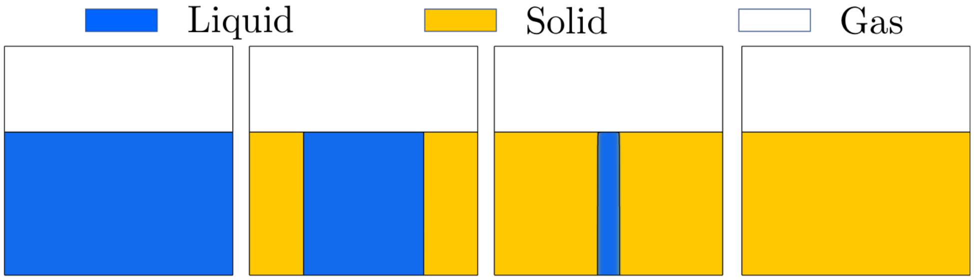
---

[14] This benchmark test is provided in IBAMR GitHub within the directory `examples/phase_change/ex4`.

that pipe defects are captured only when velocity is taken to be non-div-free, and the prior inconsistent enthalpy methods [130, 131, 31] would not be able to capture this feature of phase change process. In those works, a divergence-free (div-free) velocity condition is used, which means that no additional flow is generated when the solid phase changes to liquid or vice versa. In the momentum and energy equations, the two phases (liquid and solid) are allowed to have different densities, which is at odds with the div-free velocity assumption.

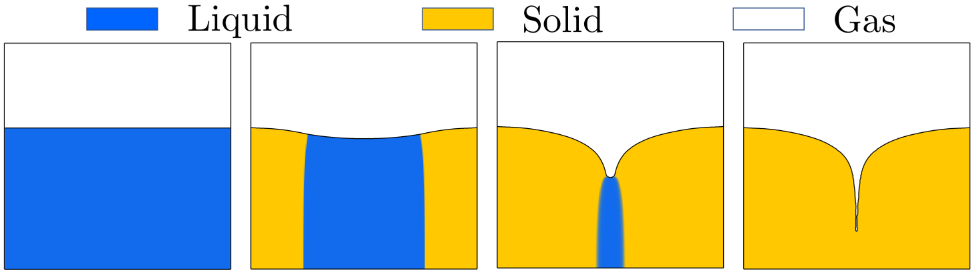
Solidification of aluminum occurs within a square computational domain of extents $\Omega \in [0, 8 \times 10^{-3}]^2$, which is discretized into 256×256 uniform cells. Initially, liquid metal is filled below the region $y = 5 \times 10^{-3}$ m, and the rest of the domain is occupied by gas (air); see Fig. 4.23(A). In contrast to the previous example, there is no solid phase at the beginning of the simulation. The densities of liquid and solid are assumed to be $\rho^{\mathrm{L}} = 2475$ kg/m^3 and $\rho^{\mathrm{S}} = 2700$ kg/m^3, respectively. The other thermophysical properties for liquid, solid, and gas are taken from the previous section. The surface tension coefficient between liquid aluminum and gas is taken to be $\sigma = 0.87$ N/m. On the top surface, zero-pressure/outflow boundary conditions are applied, while zero-velocity boundary conditions are applied elsewhere. On all boundaries, the temperature is fixed at $T = 0.5T_m$ ($T_m = 933.6$ K is the solidification temperature of aluminum), except at the bottom wall, where a zero-flux (homogenous Neumann) condition is imposed. The initial temperature in the liquid domain is set to $2T_m$, whereas in the gas domain it is $0.5T_m$. As a result of the imposed boundary conditions, solidification begins on the right and left sides of the domain. Solidification is not affected by the top boundary since air has a low thermal conductivity. The simulation runs until $t = 1$ s with a uniform time step size of $\Delta t = 10^{-5}$ s.

First, we present the simulation results assuming that velocity in the domain is div-free. Results are shown in Fig. 4.23. In this case, the liquid metal solidifies completely without deforming the free surface or changing its volume. This solution is unphysical because PCM volume must change to accommodate a change in density during phase change. It is important to note that variable density and gravitational body forces are taken into account in the momentum equation (4.45). However, this is not enough to cause caving of the free surface (gas-liquid).



$(A)\ t = 0$ $(B)\ t = 0.1$ s $(C)\ t = 0.2$ s $(D)\ t = 1$ s

Figure 4.23. Time evolution of the solid, liquid, and gas domains during metal solidification considering $\nabla \cdot \mathbf{u} = 0$.



$(A)\ t = 0$ $(B)\ t = 0.1$ s $(C)\ t = 0.2$ s $(D)\ t = 1$ s

Figure 4.24. Time evolution of the solid, liquid, and gas domains during metal solidification (shrinkage) considering $\nabla \cdot \mathbf{u} \neq 0$.

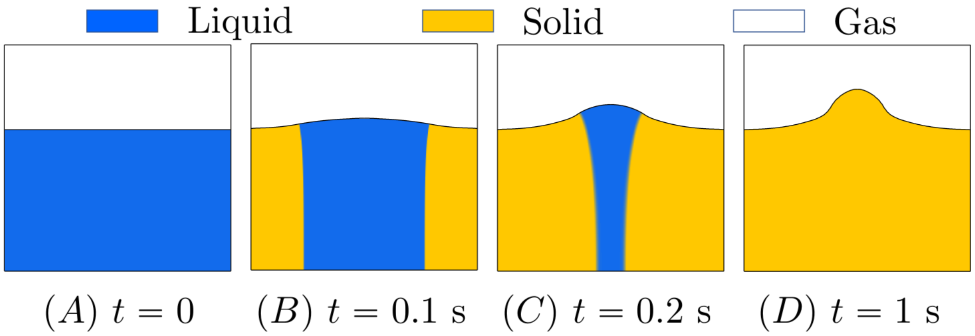


Figure 4.25. Time evolution of the solid, liquid, and gas domains during metal solidification (expansion) considering $\nabla \cdot \mathbf{u} \neq 0$.

The results differ significantly when velocity is not treated as div-free. Fig. 4.24 shows the dynamics of solidification. Around $t = 0.25$ s, the liquid metal solidifies completely, and no further phase changes occur. Supplementary video S1 shows the solidification dynamics. There are no spurious flows or interfacial motions at the gas-solid interface beyond $t = 0.25$ s. It is clear that the gas-liquid surface has caved and a pipe defect has formed. Material volume shrinks as density increases, so this is a physically correct result. In addition, we quantify the simulation results by computing the percentage mass change of the metal as a function of grid resolution. This is presented in the Appendix Sec. 6.7.

Next, consider a hypothetical case where aluminum liquid and solid phase densities are reversed to $\rho^{\mathrm{L}} = 2700$ and $\rho^{\mathrm{S}} = 2475$ kg/m^3, respectively. As a result, metal expands during solidification. Fig. 4.25 shows the dynamics of metal expansion during solidification. It takes $t = 0.25$ s for solidification to complete in this case as well, but we observe a protrusion defect instead of a pipe defect. Supplementary video S2 shows the solidification dynamics for this case.

4.7 Discussion

In this study we presented analytical and numerical methodologies to model phase change phenomena exhibiting density/volume changes. All materials of practical interest change density, and consequently, material volume to conserve mass when they melt, solidify, evaporate, or condense. Different phases of the same material also exhibit differences in other thermophysical properties, such as specific heat and conductivity. By retaining all jump terms arising from the energy equation in the Stefan condition, we derived an analytical solution to the two-phase Stefan problem with variable thermophysical properties. For validation purposes, CFD algorithms that aim to model the phase change of materials can benefit greatly from the two-phase Stefan problem solutions, but these have gone largely unnoticed in the literature. We also presented a novel low Mach version of the enthalpy method that takes into account the density change of PCMs during melting and solidification. Furthermore, the solid-liquid PCM was coupled to a gas phase within the low Mach framework. Evaporation and condensation may also be modeled with the proposed low Mach enthalpy method, but further studies are needed to evaluate its accuracy. Another possibility is to model evaporation and condensation through the level-set or volume of fluid machinery and melting and solidification via the proposed low Mach enthalpy method. In this scenario $\frac{\mathrm{D}H}{\mathrm{D}t}$ and $\frac{\mathrm{D}\phi}{\mathrm{D}t}$ are not equal to zero in the low Mach Eq. (4.33). By using such a framework, all four modes of phase change can be handled in a single simulation, which will enhance the existing modeling fidelity of engineering applications like metal additive manufacturing. Besides presenting novel techniques for modeling phase change processes, this work opens up several new directions for future research.

Acknowledgements

Chapter 4, in part, is a reprint of the material as it appears in A low Mach enthalpy method to model non-isothermal gas–liquid–solid flows with melting and solidification. Thirumalaisamy, Ramakrishnan; Bhalla, Amneet Pal Singh, International Journal of Multiphase Flow, vol. 169, 104605, 2023. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4, in part is currently being prepared for submission for publication of the material. Thirumalaisamy, Ramakrishnan; Bhalla, Amneet Pal Singh. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 5

Ongoing work and future scope

5.1 Allen-Cahn model for phase change

As discussed earlier, one popular method for simulating solidification and melting processes is the enthalpy method. Expanding upon the enthalpy method, we have developed a low Mach enthalpy method to account for volume change during melting and solidification, as described in Chapter 4. Our low Mach enthalpy method has been applied to simulate various two-phase and three-phase flows involving phase change.

Another widely used approach is the Phase-Field method, where the liquid-solid interface has a small but finite thickness [135]. Unlike the enthalpy method, the Phase-Field method solves an additional equation to govern the evolution of the liquid fraction of the phase change material, rather than relying on algebraic relations. This feature enables the Phase-Field method to incorporate more complex physics, such as anisotropy and solute transport in alloys [135].

Ziyang et al., [135] developed consistent and conservative Phase-Field model for thermo-gas-liquid-solid flows with liquid-solid phase change. The interface between the PCM and the gas region is tracked using the Cahn-Hilliard equation, while the liquid-solid interface is tracked using the Allen-Cahn equation. We have developed a model based on level set and Allen-Cahn equations for modeling non-isothermal phase change flows. Below is a brief overview of the governing equations used in our model.:

5.1.1 Governing equations

As discussed in previous chapters, the zero contour of level set function is used to track the gas-PCM interface. The level set is advected with non-divergence velocity as

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \phi \mathbf{u} = 0. \tag{5.1}$$

Next, Heaviside function H is defined based on ϕ as

$$H(\mathbf{x}) = \begin{cases} 0, & \phi(\mathbf{x}) < -n_{\text{smear}}\, h, \\ \frac{1}{2}\left(1 + \frac{1}{n_{\text{smear}} h}\phi(\mathbf{x}) + \frac{1}{\pi}\sin\left(\frac{\pi}{n_{\text{smear}} h}\phi(\mathbf{x})\right)\right), & |\phi(\mathbf{x})| \leq n_{\text{smear}}\, h, \\ 1, & \text{otherwise}. \end{cases} \tag{5.2}$$

Allen-Cahn equation, governing the evolution of liquid fraction φ, is expressed as

$$\frac{\partial \left(H\varphi\right)}{\partial t} + \nabla \cdot \mathbf{u} H\varphi = -M_{\varphi}\zeta_{\varphi} + \varphi H \nabla \cdot \mathbf{u} \text{in} \Omega, \tag{5.3}$$

where

$$\zeta_{\varphi} = \frac{\lambda_{\varphi}}{\eta_{\varphi}^{2}} H g'(\varphi) - \lambda_{\varphi}\nabla \cdot H \nabla \varphi + \frac{\rho_M^L L}{T_M} H \tilde{p}'(\varphi)(T_M - T). \tag{5.4}$$

Here, M_{φ} is the mobility of φ. The parameters γ_{φ} and μ_{φ} are the Gibbs-Thomson and linear kinetic coefficients, respectively, in the Gibbs-Thomson equation of the liquid-solid interface. λ_{φ} is the mixing energy density of φ, which is related to the thickness of the liquid-solid interface η_{φ}. Temperature equation with phase change

$$\frac{\partial (\rho C_p) T}{\partial t} + \nabla \cdot \mathbf{u}(\rho C_p) T + \rho_M^L L \left[\frac{\partial \left(H\varphi\right)}{\partial t} + \nabla \cdot \mathbf{u} H \varphi\right] = \nabla \cdot \kappa \nabla T + Q_T \tag{5.5}$$

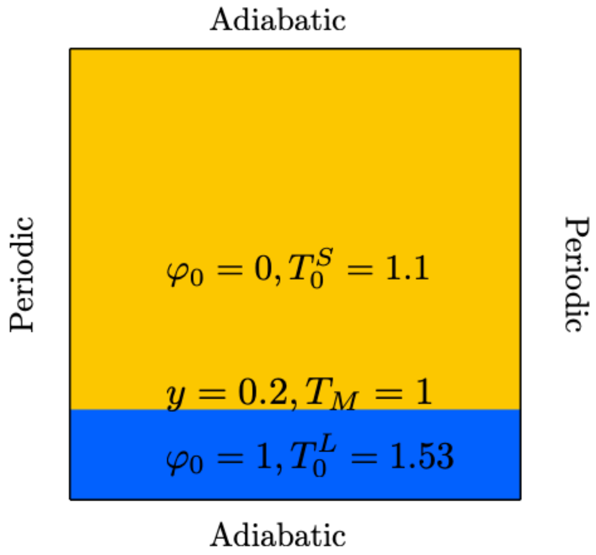


Figure 5.1. Schematic used for the Stefan problem which is solved using LS-AC model.

Finally, the Navier-Stokes equation are solved as follows

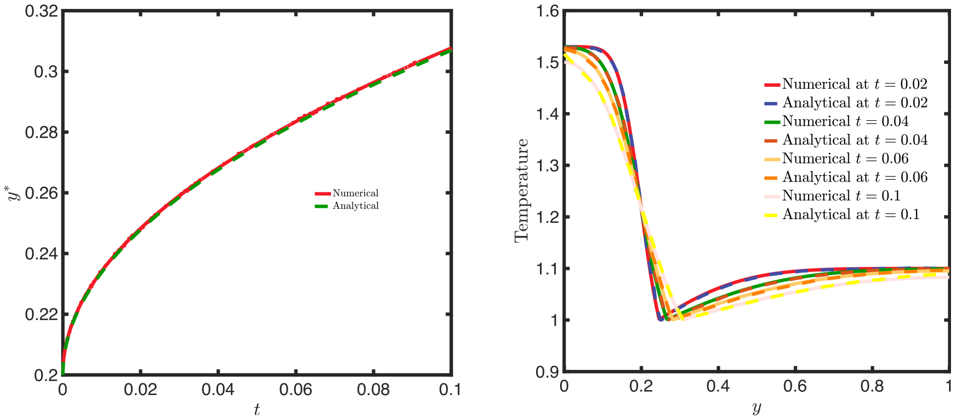
$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla p + \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right] + \rho \mathbf{g} + \mathbf{f}_s + \mathbf{f}_d, \tag{5.6}$$

$$\nabla \cdot \mathbf{u} = \frac{M_\varphi(\rho_M^L - \rho_M^S)}{\rho}\zeta_\varphi. \tag{5.7}$$

Note that the $\nabla \cdot \mathbf{u} \neq 0$ in the mushy region. The mixture model is used to derive the kinematic contraint on the velocity. The present model is referred as the level set - Allen Cahn (LS-AC) model in this thesis. The time stepping algorithm is not detained here, as it closely follows the approach described in Sec. 4.4.4. Let's validate the model that described here using the Stefan problem.

5.1.2 Stefan problem

Consider a square domain with periodic boundary conditions on the x-faces, as shown in Fig. 5.2. The top and bottom boundaries are assumed to be no-slip and adiabatic. The material properties used in this setup are as follows: $\rho = 1$, $\mu = 1$, $C = 1$, $\kappa^{\mathrm{L}} = 0.05$, $\kappa^{\mathrm{S}} = 1$, $T_m = 1$, $L = 0.53$. This problem is adapted from Javierre et al. [136]. The Allen-Cahn model parameters



(A) (B)

Figure 5.2. Stefan problem: (A) Evolution of interface y^* and (B) Temperature distribution along y axis for different times.

are $M_\varphi = 1/(\nu_\varphi \zeta_\varphi^2 L)$, $\lambda_\varphi = 1/(\nu_\varphi M_\varphi)$, $\eta_\varphi = \sqrt{2a_\varphi}\zeta_\varphi$ where $a_\varphi = 0.0625$, $\nu_\varphi = 1$ and, ζ_φ= 0.002. Initially, the liquid-solid interface is located at $y = 0.2$, with the solid phase occupying the domain above the interface with initial temperature of $T^{\mathrm{S}} = 1.1$. Below the interface, the liquid phase is present with an initial temperature of $T^{\mathrm{L}} = 1.53$.The domain is discrtized using 5×2000 cells with constant time step size of 10^{-7}.

Figure 5.2 presents the interface position and temperature distribution comparison with the analytical solution provided in [136, 135]. The numerical results from the present model matches well with the analytical solution.

Despite the promise of the phase field model, a major drawback is the requirement to specify the Allen-Cahn equation parameters. These parameters are not known a priori and necessitate a convergence study for the simulations. Therefore, there is a need for a robust approach to determine these parameters to effectively apply this model to non-isothermal phase change flows.

5.2 Future directions

In the present chapter, we presented a preliminary and ongoing investigations into the LS-AC model for modeling solidification/melting problems. In the future, we plan to validate this model for various phase change applications and simulate complicated multiphysics problems. Additionally, we aim to develop a strategy for determining the Allen-Cahn model parameters for phase change applications. Ultimately, we will compare the performance of the low Mach enthalpy method and the LS-AC method in different applications.

The low Mach enthalpy method introduced in Chapter 4 can be further extended to account for condensation and evaporation as well. Our future objective is to develop a robust and consistent low Mach model capable of handling all modes of phase change: melting, solidification, condensation, and evaporation. Furthermore, we plan to derive an analytical solution similar to the one presented in Section 4.3 for the two-phase Stefan problem with condensation/evaporation. This analytical solution will serve to validate our numerical CFD models.

The ultimate goal of this project is to develop a computational tool capable of simulating additive manufacturing processes such as selective laser melting and laser welding. A comprehensive simulation of an AM process involves several sub-models addressing various aspects, including residual stress, microstructure development, powder-bed raking (for powder-bed AM), or powder trajectories (in blown-powder AM), and the detailed behavior within and around the melt pool [5]. Our current model encompasses multiphase flows with variable surface tension, fluid-structure interaction, and phase change. We plan to enhance this model by incorporating new features such as a laser source, recoil pressure, and radiation models in a consistent manner (in discrete sense) to simulate full-scale AM processes. These models will undergo thorough validation to improve the robustness and accuracy of the solver.

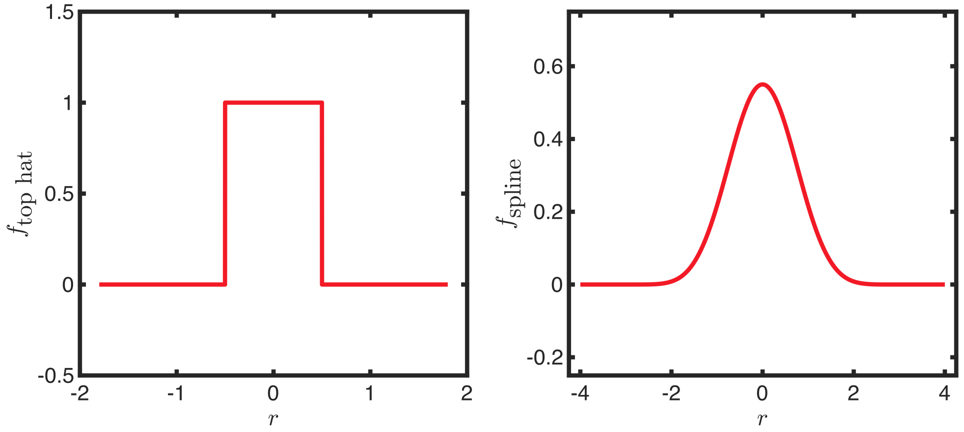
Chapter 6

Appendix

6.1 Comparison of different approaches used for constructing flux-forcing functions

Here, we demonstrate the efficacy of Approach C described in Chapter 2 in comparison to Approach D described in Chapter 2 for numerically constructing the flux-forcing functions. To implement Approach D, we discretize the interface into a set of discrete Lagrangian/marker points with position $\mathbf{X} \equiv (X, Y)$ and spread the two components (in 2D) of $\boldsymbol{\beta}(\mathbf{X}) \equiv (\beta_X, \beta_Y)$ to the nearby x- and y-faces of the Cartesian grid cells, respectively. We consider two kernel functions for the spreading operator: a one-point top hat function and a six-point Gaussian bell-like spline function. The one-dimensional (in the x-direction) form of the top hat function can be defined in terms of $r = (x - X)/h$ and it reads as

$$f_{\text{top hat}} = \begin{cases} 1, & |r| \leq 0.5, \\ 0, & \text{otherwise}. \end{cases} \tag{6.1}$$



(A) One-point top hat function

(B) Six-point spline function

Figure 6.1. One-dimensional representation of the (A) one-point top hat function; and (B) six-point spline function.

Here, x is the face-center location of the x-face and h is the grid cell size. Similarly, the one-dimensional form of the six-point spline function reads as

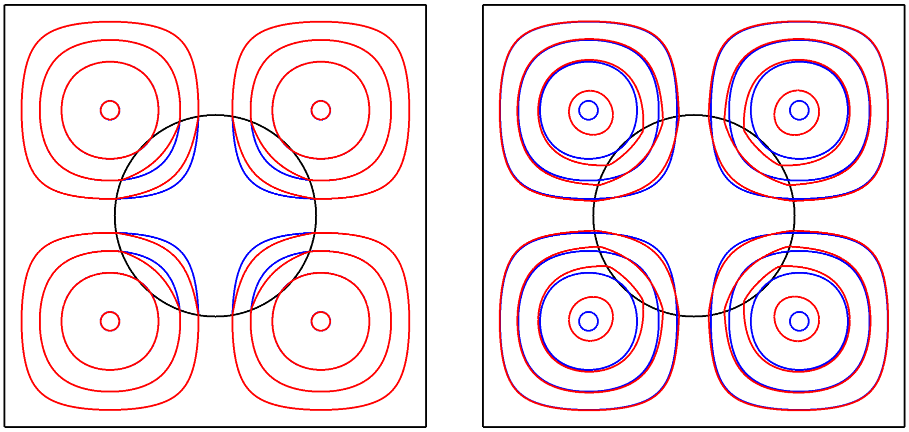
$$f_{\text{spline}} = \begin{cases} \frac{1}{60}\left(-5\sigma^5 + 90\sigma^4 - 630\sigma^3 + 2130\sigma^2 - 3465\sigma + 2193\right), & 0 \le |r| < 1, \\ \frac{1}{120}\left(5\sigma^5 - 120\sigma^4 + 1140\sigma^3 - 5340\sigma^2 + 12270\sigma - 10974\right), & 1 \le |r| < 2, \\ \frac{1}{120}\left(-\sigma^5 + 30\sigma^4 - 360\sigma^3 + 2160\sigma^2 - 6480\sigma + 7776\right), & 2 \le |r| < 3, \\ 0, & 3 \le |r|. \end{cases} \tag{6.2}$$

in which $\sigma = |r| + 3$. The graphical representation of these functions is shown in Fig. 6.1. In dimensions higher than one, a tensor-product form of the one-dimensional functions is used. We refer the readers to Peskin [38] for more details on the spreading operator. The distance between the maker points is kept approximately equal to h, although increasing or decreasing the distance up to a factor of two did not affect the overall accuracy of the scheme (data not shown).

We consider a test problem similar to the one defined in Sec. 2.7.4, in which a circle of radius 3/2 is embedded into a larger computational domain of extents $\Omega \in [0, 2\pi]^2$. Inhomogeneous Neumann boundary conditions with $g(\mathbf{x}) = -\kappa\, \mathbf{n} \cdot \nabla q_{\text{exact}}$ are imposed on the fluid-solid interface $\partial\Omega^{\mathrm{S}}$, whereas Dirichlet boundary conditions are imposed on the external boundaries of the computational domain, i.e., $q|_{\partial\Omega(\mathbf{x})} = q_{\text{exact}}(\mathbf{x})$. We present the contours of the numerical (red) and exact (blue) solutions and the convergence rate of the numerical schemes based on Approach C and D in Fig. 6.2. As observed in the figure, excellent agreement is obtained between the numerical and exact solutions in the fluid domain (considered to be outside the cylinder) with Approach C. However, when Approach D is used considering either the top hat or the spline function, there is a large disagreement between the numerical and exact solutions. Furthermore, Approach C exhibits approximately first-order accuracy using the continuous indicator function, whereas Approach D exhibits zeroth-order accuracy. A similar discrepancy is observed using the discontinuous indicator function with Approach D, although it is slightly less severe than the continuous case. In contrast, the order of accuracy of Approach C improves further when the discontinuous χ is used. Data for the discontinuous function is not shown in the interest of brevity. Based on the results of this section we do not recommend Approach D to impose the spatially varying Neumann/Robin boundary conditions.

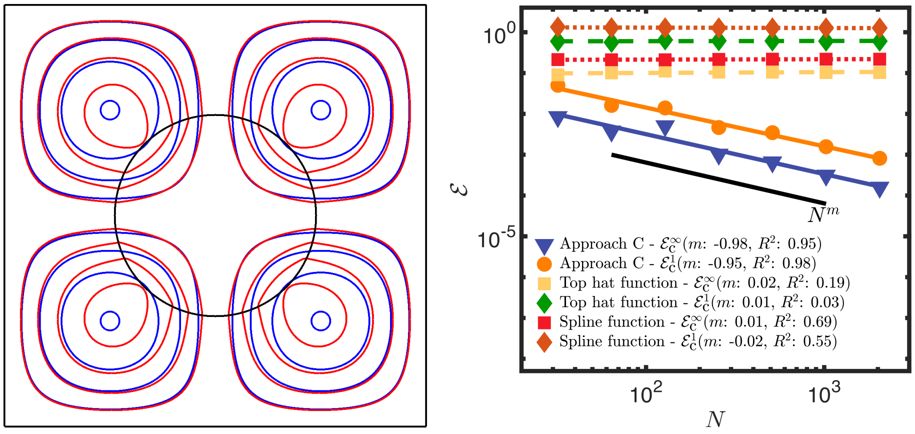
6.2 Effect of the penalization parameter

In this section, we study the effect of the penalization parameter η on the order of accuracy of the flux-based VP method. The test problem described in Sec. 2.7.4 for the hexagram interface is considered here. We solve this problem using four different η values: $\eta = \{10^{-2}, 10^{-4}, 10^{-8}, 10^{-12}\}$. As noted in Fig. 6.3, except for the largest value of $\eta = 10^{-2}$, the convergence rate remains the same for the rest of the η values. Based on the results of this section we chose $\eta = 10^{-8}$ for all our test cases.



(A) Numerical (red) vs. analytical (blue) solution using Approach C

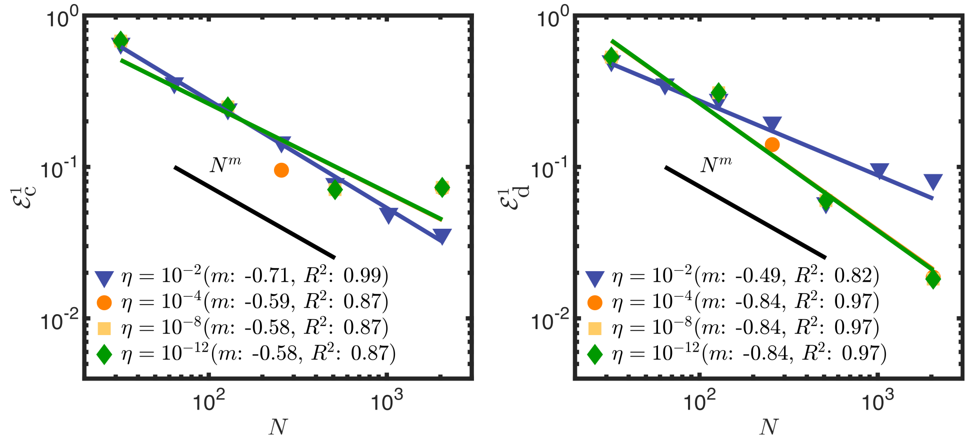
(B) Numerical (red) vs. analytical (blue) solution using Approach D (top hat)



(C) Numerical (red) vs. analytical (blue) solution using Approach D (spline)

(D) Order of convergence using the continuous indicator function

Figure 6.2. Circular domain with spatially varying Neumann boundary conditions. Contours of the numerical (red) and analytical (blue) solutions at $N = 256$ grid using (A) Approach C and using Approach D with (B) one-point top hat function and (C) six-point spline function. (D) Error norms $\mathcal{E}^1$ and $\mathcal{E}^\infty$ as a function of grid size N using the continuous indicator function with Approach C (solid line with symbols), and with Approach D using the top hat (dashed line with symbols) and spline (dotted line with symbols) kernel functions. The penalization parameter η is taken as 10^{-8}. The values of κ is taken to be 1.



(A) Order of convergence using the continuous indicator function

(B) Order of convergence using the discontinuous indicator function

Figure 6.3. Effect of the penalization parameter η on the order of accuracy of the flux-based VP mehod. Spatially varying Neumann boundary conditions are imposed on the hexagram interface using Approach C. (A) Error norms $\mathcal{E}^1$ as a function of grid size N using the continuous indicator function (B) Error norms $\mathcal{E}^1$ as a function of grid size N using the discontinuous indicator function.

6.3 Formal derivation of the projection method

The saddle-point problem

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ -\mathbf{D}\cdot & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x_u} \\ \mathbf{x_p} \end{bmatrix} = \begin{bmatrix} \mathbf{b_u} \\ \mathbf{b_p} \end{bmatrix} \tag{6.3}$$

can formally be solved using the inverse of the Schur-complement

$$\mathbf{S}^{-1} = \left(-\mathbf{D}\cdot\mathbf{A}^{-1}\mathbf{G}\right)^{-1}$$

to obtain the exact pressure and velocity solutions

$$\mathbf{x_p} = -\mathbf{S}^{-1}(\mathbf{D}\cdot\mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p}), \tag{6.4a}$$

$$\mathbf{x_u} = \mathbf{A}^{-1}(\mathbf{b_u} - \mathbf{G}\mathbf{x_p}) = \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{A}^{-1}\mathbf{G}\,\mathbf{S}^{-1}(\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p}). \tag{6.4b}$$

The projection method approximation to the velocity solution is obtained by approximating

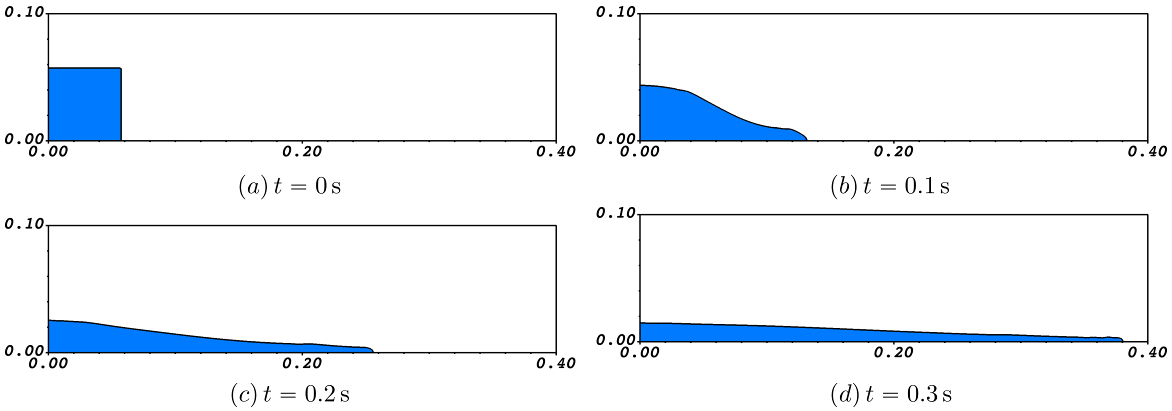
$$\begin{aligned}
\mathbf{A}^{-1}\mathbf{G}\,\mathbf{S}^{-1} &= \mathbf{A}^{-1}\mathbf{G}\left(-\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{G}\right)^{-1} \\
&= \mathbf{A}^{-1}\mathbf{G}\left(-\boldsymbol{\rho}_{\boldsymbol{\chi}}\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{G}\right)^{-1} \\
&\approx \mathbf{A}^{-1}\mathbf{G}\left(-(\mathbf{D} \cdot \boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G})\boldsymbol{\rho}_{\boldsymbol{\chi}}\mathbf{A}^{-1}\right)^{-1} = \mathbf{A}^{-1}\mathbf{G}\left(-\mathbf{L}_{\rho_\chi}\boldsymbol{\rho}_{\boldsymbol{\chi}}\mathbf{A}^{-1}\right)^{-1} \\
&= -\mathbf{A}^{-1}\mathbf{G}\mathbf{A}\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}{\mathbf{L}_{\rho_\chi}}^{-1} \\
&\approx -\mathbf{G}\mathbf{A}^{-1}\mathbf{A}\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}{\mathbf{L}_{\rho_\chi}}^{-1} \\
&= -\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G}{\mathbf{L}_{\rho_\chi}}^{-1}.
\end{aligned}$$

Here, we have commuted a few operators (which is valid only for constant-coefficient operators defined on periodic domains) to simplify $\mathbf{A}^{-1}\mathbf{G}\,\mathbf{S}^{-1}$. This simplification allows to approximate the exact velocity solution $\mathbf{x_u}$ in Eq. (6.4b) as

$$\begin{aligned}
\mathbf{x_u} &= \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{A}^{-1}\mathbf{G}\,\mathbf{S}^{-1}(\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p}) \\
&\approx \mathbf{A}^{-1}\mathbf{b_u} - \boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G}\,{\mathbf{L}_{\rho_\chi}}^{-1}(\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p}) \\
&= \mathbf{A}^{-1}\mathbf{b_u} - \Delta t\;\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G}\,\left({\mathbf{L}_{\rho_\chi}}^{-1}\frac{1}{\Delta t}(\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p})\right) \\
&= \widehat{\mathbf{x}}_{\mathbf{u}} - \Delta t\;\boldsymbol{\rho}_{\boldsymbol{\chi}}^{-1}\mathbf{G}\boldsymbol{\phi}.
\end{aligned}$$

Here, $\boldsymbol{\phi}$ is an auxiliary pressure-like scalar field that satisfies the Poisson Eq. (3.41). Approximating the inverse of the Schur complement as (see Cai et al. [78])

$$\mathbf{S}^{-1} = \left(-\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{G}\right)^{-1} \approx -\frac{1}{\Delta t}{\mathbf{L}_{\rho_\chi}}^{-1} + \boldsymbol{\mu} \tag{6.5}$$



$(a)\, t = 0\,\mathrm{s}$ $(b)\, t = 0.1\,\mathrm{s}$

$(c)\, t = 0.2\,\mathrm{s}$ $(d)\, t = 0.3\,\mathrm{s}$

Figure 6.4. Evolution of the spreading water column in two dimensions considering a density ratio of $\rho^{\mathrm{L}}/\rho^{\mathrm{G}} = 815.66$ and a viscosity ratio of $\mu^{\mathrm{L}}/\mu^{\mathrm{G}} = 63.88$ between dense water and light air fluids.

provides an approximate pressure solution $\mathbf{x_p}$

$$\begin{aligned}
\mathbf{x_p} &= -\mathbf{S}^{-1}(\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p}) \\
&\approx \left(\frac{1}{\Delta t}{\mathbf{L}_{\rho_\chi}}^{-1} - \boldsymbol{\mu}\right)(\mathbf{D} \cdot \mathbf{A}^{-1}\mathbf{b_u} + \mathbf{b_p}) \\
&= {\mathbf{L}_{\rho_\chi}}^{-1}\left(\frac{\mathbf{D} \cdot \widehat{\mathbf{x}}_\mathbf{u} + \mathbf{b_p}}{\Delta t}\right) - \boldsymbol{\mu}(\mathbf{D} \cdot \widehat{\mathbf{x}}_\mathbf{u} + \mathbf{b_p}) \\
&= \boldsymbol{\phi} - \Delta t\, \boldsymbol{\mu}\, \mathbf{L}_{\rho_\chi} \boldsymbol{\phi}.
\end{aligned}$$

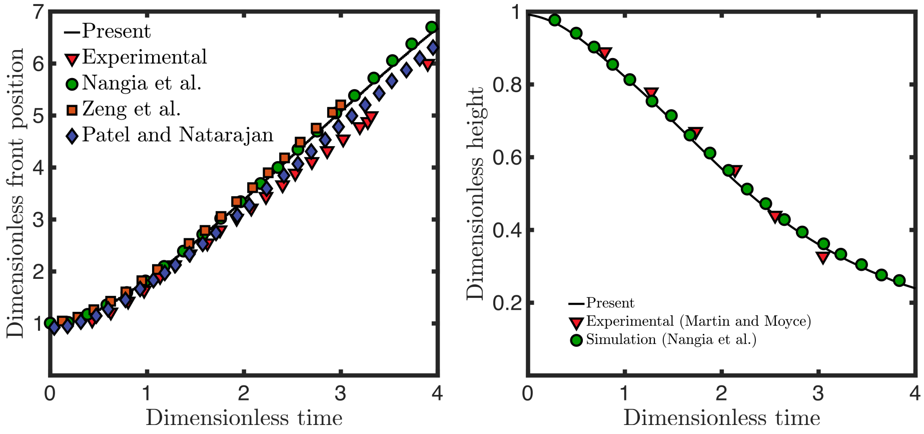
Note that in this work we use the trapezoidal rule for time integrating the viscous Laplacian term. If instead backward Euler scheme is employed then

$$\mathbf{S}^{-1} \approx -\frac{1}{\Delta t}{\mathbf{L}_{\rho_\chi}}^{-1} + 2\boldsymbol{\mu} \tag{6.6}$$

is suggested based on the spectral analysis of the viscous operator (see Cai et al. [78]).

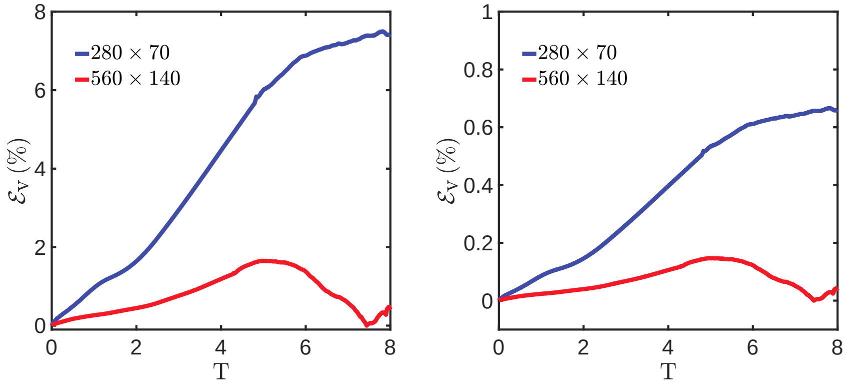
6.4 Two-dimensional dam break problem

Using the two-dimensional dam break test case, we demonstrate two aspects of our multiphase solver: (1) consistent mass and momentum transport scheme that ensures numerical



(A) (B)

Figure 6.5. Comparison of the temporal evolution of dimensionless front position with prior studies: present work (black), experiments of Martin and Moyce (red); and numerical studies of Nangia et al. (green), Zeng et al. (magenta), and Patel and Natarajan (blue).



(A) Percentage change in liquid volume (B) Percentage change in gas volume

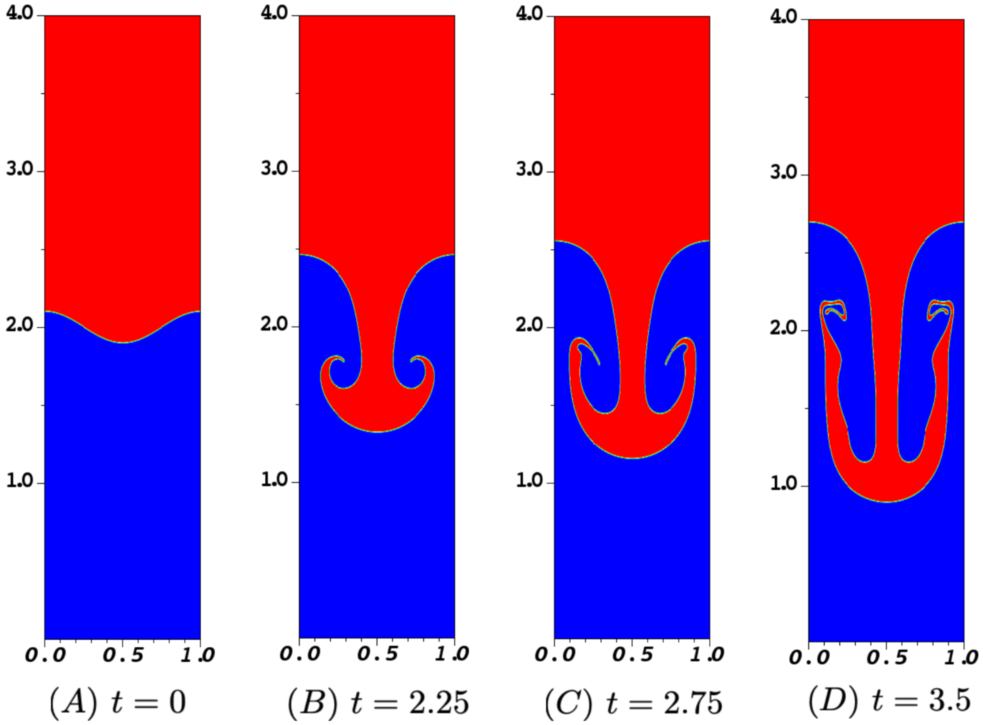
Figure 6.6. Percentage volume change $\mathcal{E}_\mathrm{v}$ of liquid and gas over time for the two-dimensional dam break problem at different grid sizes. A uniform time step size of $\Delta t = 5 \times 10^{-5}$ s is used for coarser grid ($N = 70$) and it is halved for the finer grid ($N = 140$).

stability of high density ratio flows; and (2) the percentage change in the volume of conserved phases (liquid and gas) in the absence of rigid bodies. Initially, the water column occupies a square block of size $a = 0.057$ m within a computational domain $\Omega \in [0, 7a] \times [0, 1.75a]$. The lower left corner of the water column aligns with the lower left corner of the computational domain; the remaining domain is occupied by air. A uniform grid of size $4N \times N$ is used for discretizing the computational domain. Two grid resolutions are considered $N = \{70, 140\}$. The domain has a no-slip velocity boundary condition on all sides. Water and air have densities of $\rho^{\text{L}} = 1000$ kg/m^3 and $\rho^{\text{G}} = 1.226$ kg/m^3, respectively, and viscosities of $\mu^{\text{L}} = 1.137 \times 10^{-3}$ Pa $\cdot$ s and $\mu^{\text{G}} = 1.78 \times 10^{-5}$ Pa $\cdot$ s, respectively. The surface tension coefficient between the air and water phases is taken to be $\gamma = 0.0728$ N/m. The fluids are initially at rest. This problem has been studied numerically by Nangia et al. [59], Patel and Natarajan [163] and Zeng et al. [164], and experimentally by Martin and Moyce [165].

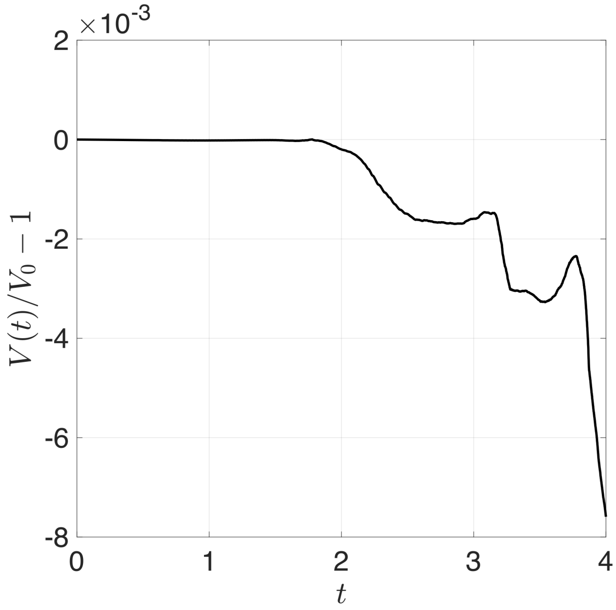
Fig. 6.4 shows the evolution of the air-water interface using a $N = 140$ grid at different time instants. Based on the results, we can conclude that our multiphase formulation provides a physical solution that remains stable over time. Using prior experiments and numerical solutions, we compare the temporal evolution of the dimensionless front position (non-dimensionalized by a) with the numerical solution. The results of our study are in excellent agreement with those of previous studies. Also presented is the percentage volume change of liquid and gas as a function of non-dimensional time $T = t\sqrt{\frac{g}{a}}$ in Fig. 6.6. As can be observed, the percentage volume change is less than 1.5% for the finer grid. This also agrees well with the previous level set-based simulations of Zeng et al. [164], who report a percentage volume change around $\mathcal{E}_{\text{v}} \approx 2\%$ for this problem.

6.5 Rayleigh-Taylor instability problem

For the level set method, different discretization, advection, and time-stepping schemes could result in different mass/volume changes of the conserved phases. Several combinations

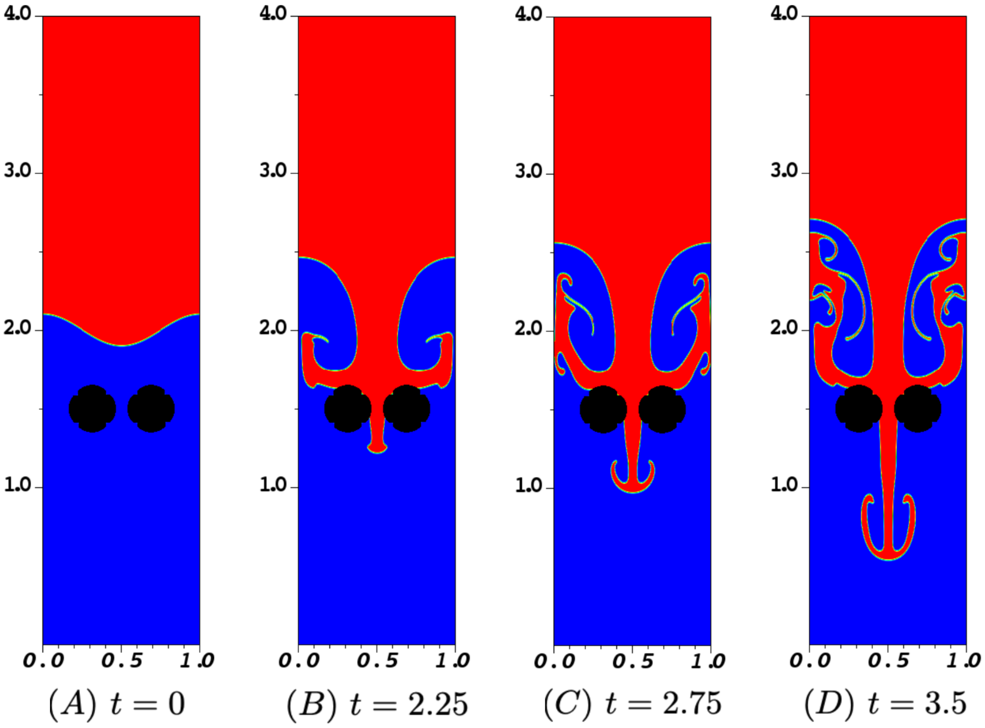


(A) Interfacial dynamics

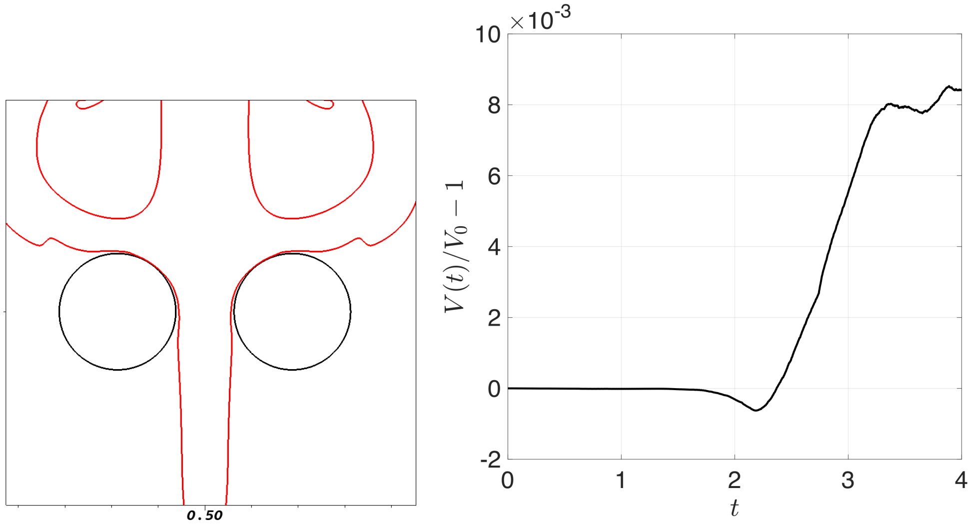


(B) Volume change of the top heavy fluid

Figure 6.7. Two phase Rayleigh-Taylor instability problem at Atwood number $A = 0.5$ and Reynolds number $Re = 3000$. (A) Temporal evolution of the top heavy (red color) and bottom light (blue color) fluids in the domain. (B) Volume change of the heavy fluid over time. Here, time t is non-dimensionalized by $d/\sqrt{gd}$.



(A) Interfacial dynamics



(B) Zoomed-in view near the obstacles at $t = 3.5$

(C) Volume change of the top heavy fluid

Figure 6.8. Three phase Rayleigh-Taylor instability problem at Atwood number $A = 0.5$ and Reynolds number $Re = 3000$. Two rigid cylinders (shown as black disks) are placed in the path of the falling heavy fluid. (B) Zoomed-in view of the fluid-fluid interface (red color) and the outer surface of the cylinders (black color). (A) Temporal evolution of the top heavy (red color) and bottom light (blue color) fluids in the domain. (C) Volume change of the heavy fluid over time. Here, time t is non-dimensionalized by $d/\sqrt{gd}$.

of these schemes have been extensively tested by Solomenko et al. [166] on a number of benchmarking two-phase problems, including the Rayleigh-Taylor instability (RTI) flow. It is important to note, however, that the authors did not include the subcell-fix method in their study as they mention that "*These methods were not tested here because we have found [their] implementation comparatively complex.*" To complement the prior study, as well as to compare the performance of the subcell-fix method with the best performing method(s) reported in [166], we simulate the RTI case using the same grid resolution ($\Delta x = \Delta y = h = 1/192$) and problem parameters as in Sec. 7 of Solomenko et al. Specifically, we consider the Atwood number to be $A = (\rho_h - \rho_l)/(\rho_h + \rho_l) = 0.5$, and the Reynolds number to be $Re = \rho_h d\sqrt{gd}/\mu = 3000$. Here, ρ_h and ρ_l, denote density of heavier and lighter phase, respectively, g is the acceleration to gravity (acting in the negative y direction), and μ is the fluid viscosity, which is the same in both fluid phases. The initial interface between the top heavy and bottom light fluid is considered to be a cosine function with a small amplitude. The initial signed distance is then expressed as $\sigma(x, y, t = 0) = d(2 + 0.1\cos(2\pi x/d)) - y$, in which d is the domain width and $4d$ is the domain height. Fig. 6.7(A) shows the two phase interface evolution at various non-dimensional time instants t (normalized by time scale $d/\sqrt{gd}$), whereas Fig. 6.7(B) reports the change in the volume of the heavier fluid as a function of non-dimensional time. Solomenko et al. report volume changes till $t = 3.5$, and their best performing method (in terms of least amount of volume changes and parasitic oscillations at the interface) leads to a percentage volume change of 0.3% (reported in Table 10 of [166]). As can be seen in Fig. 6.7, the subcell-fix method of Min also leads to similar volume change around $t = 3.5$, and does not cause parasitic oscillations at the evolving interface.

As a variation of the two phase RTI problem, and to examine the effect of the Brinkman penalty term on volume change of the phases, we introduce two cylinders of radius $R = 0.125d$ in the path of the falling heavy fluid. Left and right cylinder centers are located at $(0.3125d, 1.5d)$ and $(0.6875d, 1.5d)$, respectively. These objects are held stationary throughout the simulation. Fig. 6.8(A) shows the perturbed dynamics of the fluid-fluid interface in the presence of rigid obstacles, which results in a more chaotic dynamics than the two phase RTI problem. A zoomed-in view of the fluid-fluid interface at $t = 3.5$ near the stationary cylinders is provided in Fig. 6.8(B). The no-slip condition imposed by the Brinkman penalty term prevents the fluid-fluid interface from penetrating the rigid surfaces. Fig. 6.8(C) shows the volume change of the heavier fluid as a function of time. The percentage volume change for the heavier fluid at $t = 4$ is approximately 0.8%, which is similar to the two-phase RTI problem; see Fig. 6.7(B). We can conclude from this example that the Brinkman penalization term does not result in additional mass/volume changes (in the conserved phases) beyond what is expected from the standard level set method.

6.6 Similarity solution

Here, we derive a similarity solution of the heat equations governing temperature in the solid $\Omega^{\rm S}$ and liquid $\Omega^{\rm L}$ domains

$$\frac{\partial T^{\rm S}}{\partial t} = \alpha^{\rm S}\frac{\partial^2 T^{\rm S}}{\partial x^2} \qquad \in \Omega^{\rm S}(t), \tag{6.7}$$

$$\frac{\partial T^{\rm L}}{\partial t} + (1 - R_\rho)\,\frac{\rm ds}{\rm dt}\frac{\partial T^{\rm L}}{\partial x} = \alpha^{\rm L}\frac{\partial^2 T^{\rm L}}{\partial x^2} \qquad \in \Omega^{\rm L}(t). \tag{6.8}$$

Interface position $s(t)$ can be expressed as $s(t) = \lambda(t)2\sqrt{\alpha^{\rm L} t}$, where $\lambda(t)$ is an unknown function of time. Eqs. (6.20) and (6.21) can be written in terms of similarity variables $\eta^{\rm S}$ and $\eta^{\rm L}$, respectively:

$$T^{\rm S}(x,t) = T^{\rm S}(\eta^{\rm S}) \qquad \text{with} \qquad \eta^{\rm S} = \frac{x}{2\sqrt{\alpha^{\rm S} t}}, \tag{6.9}$$

$$T^{\rm L}(x,t) = T^{\rm L}(\eta^{\rm L}) \qquad \text{with} \qquad \eta^{\rm L} = \frac{x}{2\sqrt{\alpha^{\rm L} t}} + b(t). \tag{6.10}$$

$b(t)$ in η^{L} is yet to be determined. The similarity transformation reduces partial differential equations (6.20) and (6.21) in x and t to ordinary differential equations in η^{S} and η^{L}, respectively. The steps involved in the similarity transformation of (6.21) are detailed in the remainder of this section as this equation is different from the standard heat (6.20), which has been treated in several textbooks. The main steps involve rewriting the derivatives of T^{L} and $s(t)$

$$\frac{\partial T^{\mathrm{L}}}{\partial x} = \frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}}\frac{\partial \eta^{\mathrm{L}}}{\partial x} = \frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}}\frac{1}{2\sqrt{\alpha^{\mathrm{L}}t}} \tag{6.11}$$

$$\frac{\partial^2 T^{\mathrm{L}}}{\partial x^2} = \frac{\mathrm{d}^2\mathrm{T}^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}^2}}\frac{\partial \eta^{\mathrm{L}}}{\partial x}\frac{1}{2\sqrt{\alpha^{\mathrm{L}}t}} = \frac{\mathrm{d}\mathrm{T}^{\mathrm{L}}}{\mathrm{d}^2\eta^{\mathrm{L}^2}}\frac{1}{4\alpha^{\mathrm{L}}t} \tag{6.12}$$

$$\frac{\partial T^{\mathrm{L}}}{\partial t} = \frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}}\frac{\partial \eta^{\mathrm{L}}}{\partial t} = \frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}}\left(\frac{-x}{4t\sqrt{\alpha^{\mathrm{L}}t}} + \frac{\mathrm{d}b}{\mathrm{d}t}\right) \tag{6.13}$$

$$\frac{\mathrm{d}s}{\mathrm{d}t} = \lambda\sqrt{\frac{\alpha^{\mathrm{L}}}{t}} + 2\sqrt{\alpha^{\mathrm{L}}t}\,\frac{\mathrm{d}\lambda}{\mathrm{d}t} \tag{6.14}$$

and substituting them in (6.21). This yields

$$\frac{\mathrm{d}^2\mathrm{T}^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}^2}} + 2\left(\eta^{\mathrm{L}} - b - \lambda(1-R_\rho) - 2t\frac{\mathrm{d}b}{\mathrm{d}t} - 2t(1-R_\rho)\frac{\mathrm{d}\lambda}{\mathrm{d}t}\right)\frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}} = 0. \tag{6.15}$$

Here, $\frac{\mathrm{d}^2\mathrm{T}^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}^2}} = \frac{\mathrm{d}}{\mathrm{d}\eta^{\mathrm{L}}}\left(\frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}}\right)$. Choosing $b(t) = -\lambda(t)(1-R_\rho)$ in (6.15) simplifies the similarity transformation of (6.21) to

$$\frac{\mathrm{d}^2\mathrm{T}^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}^2}} + 2\eta^{\mathrm{L}}\frac{\mathrm{d}T^{\mathrm{L}}}{\mathrm{d}\eta^{\mathrm{L}}} = 0. \tag{6.16}$$

Following similar (but less involved) steps, the similarity transformation of (6.20) reads as

$$\frac{\mathrm{d}^2\mathrm{T}^{\mathrm{S}}}{\mathrm{d}\eta^{\mathrm{S}^2}} + 2\eta^{\mathrm{S}}\frac{\mathrm{d}T^{\mathrm{S}}}{\mathrm{d}\eta^{\mathrm{S}}} = 0. \tag{6.17}$$

Eqs. (6.16) and (6.17) can be integrated analytically and the closed-form expressions of $T^{\mathrm{S}}(x,t) = T^{\mathrm{S}}(\eta^{\mathrm{S}})$ and $T^{\mathrm{L}}(x,t) = T^{\mathrm{L}}(\eta^{\mathrm{L}})$ (satisfying boundary and interface conditions) are provided in the main text.

6.7 Grid convergence study for mass conservation

For the metal melting and solidification cases discussed in the main text, we present the percentage change in the PCM mass. The mass of PCM (solid+liquid) in the domain at time t can be computed using

$$m(t) = \int_{\Omega} \left[\rho^{\mathrm{L}}(H\varphi) + \rho^{\mathrm{S}}(H - H\varphi)\right] \, \mathrm{d}\Omega. \tag{6.18}$$

The relative percentage error in PCM mass is defined as

$$\mathcal{E}(t) = \frac{|m - m_0|}{m_0} \times 100, \tag{6.19}$$

in which m_0 is the initial (at time $t = 0$) mass of the PCM. Fig. 6.9 plots $\mathcal{E}$ versus t for various grids of size $N_x \times N_y = N^2$. With increasing grid refinement, $\mathcal{E}$ decreases and at finer grids there is about 0.007% change in the PCM mass. Fig. 6.10 plots $\mathcal{E}$ versus t for the metal solidification case that exhibits pipe shrinkage. A similar conclusion can be drawn for this case as well: $\mathcal{E}$ decreases with grid refinement and at finer grids PCM mass changes by approximately 0.39%. $\mathcal{E}$ also decreases with decreasing Δt (data not shown).

6.8 Analytical solution of a Two phase Stefan problem

In this Appendix, we provide the analytical solutions of a two-phase Stefan problem as considered in Sec. 5.3. Let's consider a slab $\Omega := 0 \leq x \leq l$ with the solid occupies the entire domain at $t = 0$. Initially, the temperature is uniform throughout the domain and is lower than the melting temperature ($T_i < T_m$). The boundary is open on the right and closed on the left. The

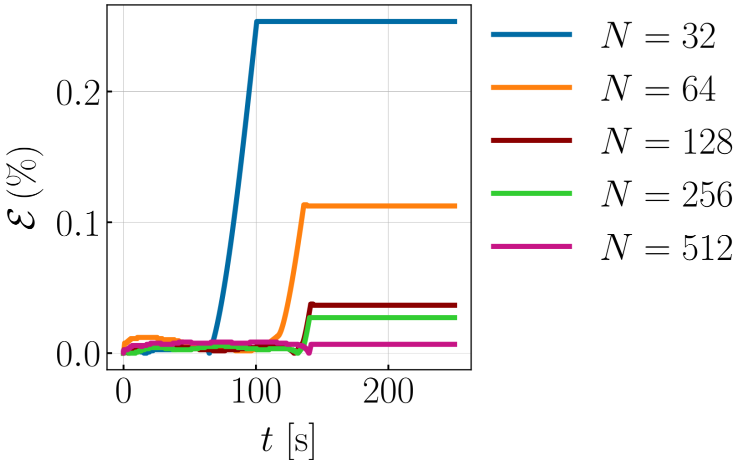


Figure 6.9. Percentage change in PCM mass $\mathcal{E}$ as a function of time for the metal melting case at different grids. Each grid uses a uniform time step size of $\Delta t = 10^{-3}$ s and a temperature interval of $\Delta T = 10$ K.

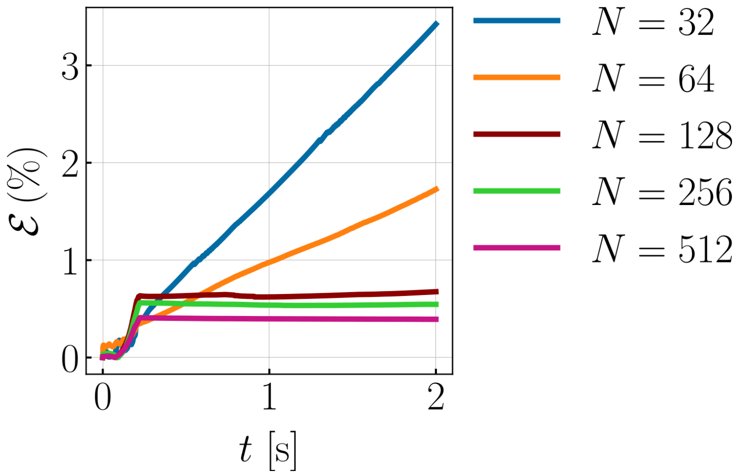


Figure 6.10. Percentage change in PCM mass $\mathcal{E}$ as a function of time for the metal solidification case that exhibits pipe shrinkage defect at different grids. The temperature interval is taken to be $\Delta T = 10$ K for all grids. A uniform time step size is used for all the grids: for coarse grids $N = 32$ and $N = 64$, $\Delta t = 10^{-4}$ s is used; for medium grids $N = 128$ and $N = 256$, $\Delta t = 10^{-5}$ s is used; and for the fine grid $N = 512$, $\Delta t = 10^{-6}$ s is used.

solid is melted by imposing a temperature $T_o > T_m$ on the left boundary. The thermophysical properties of solid phase are $\rho^{\rm S}$, $C^{\rm S}$, and $\kappa^{\rm S}$, and for the liquid phase are: $\rho^{\rm L}$, $C^{\rm L}$, and $\kappa^{\rm L}$. We assume these properties are constant within each phase. When the solid melts, the melt front $x^* = s(t)$ starts to move to the right. The melting continues until all the solid liquefies. The solid-to-liquid density ratio is denoted as $R_\rho = \rho^{\rm L}/\rho^{\rm S}$. When $R_\rho < 1$, the solid expands as it melts; if $R_\rho > 1$, the solid shrinks as it melts. We have derived an analytical solution for the two-phase Stefan solidification problem in [167]. We follow a very similar approach here. For more details about the detailed derivation and jump conditions, please refer to our previous work [167]. The governing equations in the solid and liquid phases are

$$\rho^{\rm S} C^{\rm S} \left(\frac{\partial T^{\rm S}}{\partial t} + u^{\rm S} \frac{\partial T^{\rm S}}{\partial x} \right) = \kappa^{\rm S} \frac{\partial^2 T^{\rm S}}{\partial x^2} \in \Omega^{\rm S}(t), \tag{6.20}$$

$$\rho^{\rm L} C^{\rm L} \left(\frac{\partial T^{\rm L}}{\partial t} + u^{\rm L} \frac{\partial T^{\rm L}}{\partial x} \right) = \kappa^{\rm L} \frac{\partial^2 T^{\rm L}}{\partial x^2} \in \Omega^{\rm L}(t), \tag{6.21}$$

where $u^{\rm S}$ and $u^{\rm L}$ are the velocities of the solid and liquid phases, respectively. When melting occurs, the melt front moves with velocity $u^* = {\rm d}s/{\rm d}t$. The velocity in the liquid domain $\Omega^{\rm L}(t) := 0 \le x < s(t)$ is taken to be zero. The velocity jump across the interface is given by ([167])

$$[\![u]\!] = u^{\rm S}(s^+, t) - u^{\rm L}(s^-, t) = u^{\rm S}(s^+, t) = (1 - R_\rho) \frac{{\rm d}s}{{\rm d}t}, \tag{6.22}$$

where s^+ and s^- represent spatial locations just ahead and behind the interface. From the continuity equation $\partial u^{\rm S}/\partial x = 0$, the solid velocity $u^{\rm S}$ is constant in the solid phase, allowing us to obtain $u^{\rm S}(\Omega^{\rm S}, t) \equiv u^{\rm S}(s^+, t) = (1 - R_\rho) \frac{{\rm d}s}{{\rm d}t}$.

We need to find the solution of Eqs. 6.20 and 6.21 with the boundary conditions

$T^{\mathrm{L}}(0,t) = T_o$ and $T^{\mathrm{S}}(l,t) = T_i$ and the interfacial conditions

$$T^{\mathrm{S}}(x^*,t) = T^{\mathrm{L}}(x^*,t) = T_m, \tag{6.23}$$

$$\rho^{\mathrm{L}}\left[(C^{\mathrm{L}} - C^{\mathrm{S}})(T_m - T_r) + L + \frac{1}{2}(1 - R_\rho^2)\left(\frac{\mathrm{d}s}{\mathrm{d}t}\right)^2\right]\frac{\mathrm{d}s}{\mathrm{d}t}$$
$$= \left(\kappa^{\mathrm{S}}\frac{\partial T^{\mathrm{S}}}{\partial x} - \kappa^{\mathrm{L}}\frac{\partial T^{\mathrm{L}}}{\partial x}\right)_{x^*}. \tag{6.24}$$

The last equation Eq. 6.24 is known as the Stefan condition, stating that the rate of change in latent heat is equal to the heat flux jumps across the interface.

As shown in [167], Eqs. 6.20 and 6.21 admit a similarity solution of the form

$$T^{\mathrm{S}}(x,t) = T_i + A(\lambda(t))\,\mathrm{erfc}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{S}}t}} - \frac{s(t)}{2\sqrt{\alpha^{\mathrm{S}}t}}(1 - R_\rho)\right), \tag{6.25}$$

$$T^{\mathrm{L}}(x,t) = T_o + B(\lambda(t))\,\mathrm{erf}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{L}}t}}\right). \tag{6.26}$$

Here, $\alpha^{\mathrm{S}} = \kappa^{\mathrm{S}}/(\rho^{\mathrm{S}}C^{\mathrm{S}})$, $\alpha^{\mathrm{L}} = \kappa^{\mathrm{L}}/(\rho^{\mathrm{L}}C^{\mathrm{L}})$ are the solid and liquid thermal diffusivities, respectively, and $\lambda(t) = \frac{s(t)}{2\sqrt{\alpha^{\mathrm{S}}t}}$ is a yet to be determined function of time. The coefficients $A(\lambda(t))$ and $B(\lambda(t))$ are found using the interface temperature condition as

$$T_i + A(\lambda(t))\,\mathrm{erfc}\,(\lambda(t)R_\rho) = T_o + B(\lambda(t))\,\mathrm{erf}\left(\frac{s(t)}{2\sqrt{\alpha^{\mathrm{L}}t}}\right) = T_m.$$

This yields

$$A(\lambda(t)) = \frac{T_m - T_i}{\mathrm{erfc}\,(\lambda(t)R_\rho)} \qquad \text{and} \qquad B(\lambda(t)) = \frac{T_m - T_o}{\mathrm{erf}\left(\lambda(t)\sqrt{\frac{\alpha^{\mathrm{S}}}{\alpha^{\mathrm{L}}}}\right)}. \tag{6.27}$$

Therefore, the temperature distribution in the solid phase is given by

$$T^{\mathrm{S}} = T_i + (T_m - T_i)\frac{\mathrm{erfc}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{S}}t}} - \lambda(t)(1 - R_\rho)\right)}{\mathrm{erfc}\,(\lambda(t)R_\rho)} \tag{6.28}$$

and in the liquid phase

$$T^{\mathrm{L}} = T_o + (T_m - T_o)\frac{\text{erf}\left(\frac{x}{2\sqrt{\alpha^{\mathrm{L}}t}}\right)}{\text{erf}\left(\lambda(t)\sqrt{\frac{\alpha^{\mathrm{S}}}{\alpha^{\mathrm{L}}}}\right)}. \tag{6.29}$$

By using the temperature distributions in the liquid (Eq. 6.29) and solid phases (Eqs. 6.28), along with the interface position $s(t) = 2\lambda(t)\sqrt{\alpha^{\mathrm{S}}t}$, we obtain the equation for $\lambda(t)$ from the Stefan condition (Eq. 6.24)

$$\begin{aligned} &\rho^{\mathrm{L}}\left[L^{\text{eff}} - \frac{(1-R_\rho^2)}{2}\left(\frac{\lambda^2\alpha^{\mathrm{S}}}{t}\right)\right]\lambda\sqrt{\alpha^{\mathrm{S}}} = \\ &-\kappa^{\mathrm{S}}\frac{T_m - T_i}{\text{erfc}\,(\lambda R_\rho)}\frac{e^{-\lambda^2 R_\rho^2}}{\sqrt{\pi\alpha^{\mathrm{S}}}} - \kappa^{\mathrm{L}}\frac{T_m - T_o}{\text{erf}\left(\lambda\sqrt{\frac{\alpha^{\mathrm{S}}}{\alpha^{\mathrm{L}}}}\right)}\frac{e^{-\lambda^2\alpha^{\mathrm{S}}/\alpha^{\mathrm{L}}}}{\sqrt{\pi\alpha^{\mathrm{L}}}}, \end{aligned} \tag{6.30}$$

where $L^{\text{eff}} = L + (C^{\mathrm{L}} - C^{\mathrm{S}})(T_m - T_r)$ represents the effective latent heat. In this equation, only the leading-order $1/\sqrt{t}$ term is retained, while other terms related to the derivative of $\lambda(t)$ are dropped. Through numerical experiments, we observed that terms involving $\mathrm{d}\lambda/\mathrm{dt}$ can be omitted [167].

It is also possible to find the pressure inside the fluid and solid phases as demonstrated in [167]. However, for brevity, we do not provide the derivation here. Additionally, note that we assume $L^{\text{eff}} = L$ in our Stefan problem simulations.

6.9 Error norm and curve-fitting of the data

Table 6.1. Error norm data for the concentric circular annulus case using Approach A considered in Sec. 2.7.3

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	4.6978×10^{-1}	–	1.94,0.96	1.1552	–	1.78, 0.99
	64	9.82×10^{-2}	1.9149×10^{-1}	1.29		7.3710×10^{-1}	0.65	
	128	4.91×10^{-2}	5.3379×10^{-2}	1.84		1.9393×10^{-1}	1.93	
	256	2.45×10^{-2}	1.6787×10^{-2}	1.67		4.8113×10^{-2}	2.01	
	512	1.22×10^{-2}	9.8071×10^{-3}	0.78		2.0514×10^{-2}	1.23	
	1024	6.13×10^{-3}	7.8649×10^{-4}	3.64		3.4221×10^{-3}	2.58	
	2048	3.06×10^{-3}	1.1559×10^{-4}	2.77		8.6893×10^{-4}	1.98	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	1.0015×10^{-2}	–	1.98, 0.99	2.7865×10^{-1}	–	1.50, 0.98
	64	9.82×10^{-2}	3.7365×10^{-2}	1.42		2.3764×10^{-2}	0.23	
	128	4.91×10^{-2}	9.7656×10^{-3}	1.94		8.0442×10^{-2}	1.56	
	256	2.45×10^{-2}	2.7216×10^{-3}	1.84		2.1673×10^{-2}	1.89	
	512	1.22×10^{-2}	1.0573×10^{-3}	1.36		9.4256×10^{-3}	1.20	
	1024	6.13×10^{-3}	9.2188×10^{-5}	3.52		2.1399×10^{-3}	2.14	
	2048	3.06×10^{-3}	3.0885×10^{-5}	1.58		7.9984×10^{-4}	1.42	

Table 6.2. Error norm data for the concentric circular annulus case using Approach B considered in Sec. 2.7.3

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	4.8654×10^{-1}	–	1.93, 0.97	1.1256	–	1.78, 0.97
	64	9.82×10^{-2}	1.9149×10^{-1}	1.35		7.4819×10^{-1}	0.59	
	128	4.91×10^{-2}	5.3379×10^{-2}	1.84		1.9643×10^{-1}	1.93	
	256	2.45×10^{-2}	1.6787×10^{-2}	1.67		5.6235×10^{-2}	1.80	
	512	1.22×10^{-2}	9.8071×10^{-3}	0.78		2.8225×10^{-2}	0.99	
	1024	6.13×10^{-3}	8.0832×10^{-4}	3.60		3.0398×10^{-3}	3.21	
	2048	3.06×10^{-3}	1.2137×10^{-4}	2.74		8.2524×10^{-4}	1.88	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	1.0388×10^{-1}	–	1.99, 0.99	2.6491×10^{-1}	–	1.52, 0.98
	64	9.82×10^{-2}	3.7747×10^{-2}	1.46		2.2513×10^{-1}	0.23	
	128	4.91×10^{-2}	9.8976×10^{-3}	1.93		7.4914×10^{-2}	1.59	
	256	2.45×10^{-2}	2.7456×10^{-3}	1.85		2.0745×10^{-2}	1.85	
	512	1.22×10^{-2}	1.0606×10^{-3}	1.37		9.2775×10^{-3}	1.16	
	1024	6.13×10^{-3}	9.2313×10^{-5}	3.52		1.8323×10^{-3}	2.34	
	2048	3.06×10^{-3}	3.1502×10^{-5}	1.55		6.8778×10^{-4}	1.41	

Table 6.3. Error norm data for the concentric circular annulus case using Approach C considered in Sec. 2.7.3

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	7.5945×10^{-1}	–	1.22, 0.82	1.1582	–	1.79, 0.97
	64	9.82×10^{-2}	1.9216×10^{-1}	1.98		7.5692×10^{-1}	0.61	
	128	4.91×10^{-2}	5.3385×10^{-2}	1.85		1.9907×10^{-1}	1.93	
	256	2.45×10^{-2}	8.3493×10^{-2}	-0.64		5.7075×10^{-2}	1.80	
	512	1.22×10^{-2}	2.0512×10^{-2}	2.03		2.8476×10^{-2}	1.00	
	1024	6.13×10^{-3}	1.7283×10^{-3}	3.57		3.0329×10^{-3}	3.23	
	2048	3.06×10^{-3}	9.3256×10^{-3}	-2.43		8.2703×10^{-4}	1.87	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	1.3074×10^{-1}	–	0.91, 0.79	2.6907×10^{-1}	–	1.53, 0.98
	64	9.82×10^{-2}	4.1687×10^{-2}	1.65		2.2643×10^{-1}	0.25	
	128	4.91×10^{-2}	1.3548×10^{-2}	1.62		7.5307×10^{-2}	1.59	
	256	2.45×10^{-2}	2.7071×10^{-2}	-1.00		2.0848×10^{-2}	1.85	
	512	1.22×10^{-2}	6.3965×10^{-3}	2.08		9.3043×10^{-3}	1.16	
	1024	6.13×10^{-3}	1.2647×10^{-3}	2.34		1.8262×10^{-3}	2.35	
	2048	3.06×10^{-3}	4.7482×10^{-3}	-1.91		6.8950×10^{-4}	1.41	

Table 6.4. Error norm data for the hexagram case considered in Sec. 2.7.4

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	6.7780×10^{-1}	–	0.58, 0.87	5.3142×10^{-1}	–	0.84, 0.97
	64	9.82×10^{-2}	3.4265×10^{-1}	0.98		3.6469×10^{-1}	0.54	
	128	4.91×10^{-2}	2.4912×10^{-1}	0.46		3.0923×10^{-1}	0.24	
	256	2.45×10^{-2}	9.4835×10^{-2}	1.39		1.3984×10^{-1}	1.14	
	512	1.22×10^{-2}	7.0835×10^{-2}	0.42		5.9982×10^{-2}	1.22	
	1024	6.13×10^{-3}	7.6675×10^{-2}	-0.11		3.5921×10^{-2}	0.74	
	2048	3.06×10^{-3}	7.3231×10^{-2}	0.07		1.8182×10^{-2}	0.98	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	2.3021×10^{-1}	–	0.56, 0.89	2.6085×10^{-1}	–	0.78, 1.00
	64	9.82×10^{-2}	9.2871×10^{-2}	1.31		1.4826×10^{-1}	0.82	
	128	4.91×10^{-2}	8.3922×10^{-2}	0.15		8.4091×10^{-2}	0.82	
	256	2.45×10^{-2}	4.5081×10^{-2}	0.90		5.7254×10^{-2}	0.55	
	512	1.22×10^{-2}	2.3251×10^{-2}	0.96		2.8368×10^{-2}	1.01	
	1024	6.13×10^{-3}	2.7484×10^{-2}	-0.24		1.9064×10^{-2}	0.57	
	2048	3.06×10^{-3}	2.3980×10^{-2}	0.20		9.4295×10^{-3}	1.02	

Table 6.5. Error norm data for the egg case considered in Sec. 2.7.4

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	4.9929×10^{-2}	–	1.00, 0.95	8.6565×10^{-1}	–	1.37, 0.96
	64	9.82×10^{-2}	3.1338×10^{-2}	0.67		5.6384×10^{-1}	0.62	
	128	4.91×10^{-2}	1.0582×10^{-2}	1.57		5.3923×10^{-1}	0.06	
	256	2.45×10^{-2}	1.2636×10^{-2}	-0.26		1.0306×10^{-2}	2.39	
	512	1.22×10^{-2}	5.0855×10^{-3}	1.31		3.2609×10^{-3}	1.66	
	1024	6.13×10^{-3}	1.8937×10^{-3}	1.43		1.0862×10^{-3}	1.59	
	2048	3.06×10^{-3}	6.3391×10^{-4}	1.58		4.4750×10^{-4}	1.28	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	7.2046×10^{-3}	–	0.67, 0.83	3.0454×10^{-2}	–	0.54, 0.91
	64	9.82×10^{-2}	1.1199×10^{-2}	-0.64		2.5406×10^{-2}	0.26	
	128	4.91×10^{-2}	7.7484×10^{-3}	0.53		1.9357×10^{-2}	0.39	
	256	2.45×10^{-2}	6.2520×10^{-3}	0.31		1.7718×10^{-2}	0.13	
	512	1.22×10^{-2}	2.2338×10^{-3}	1.48		1.0803×10^{-2}	0.71	
	1024	6.13×10^{-3}	1.6059×10^{-3}	0.48		6.7722×10^{-3}	0.67	
	2048	3.06×10^{-3}	5.3624×10^{-4}	1.58		2.7800×10^{-3}	1.28	

Table 6.6. Error norm data for the x-cross case considered in Sec. 2.7.4

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	1.9617×10^{-1}	–	0.94, 1.00	7.9954×10^{-1}	–	1.08, 0.98
	64	9.82×10^{-2}	1.0360×10^{-1}	0.92		2.9599×10^{-1}	1.43	
	128	4.91×10^{-2}	6.3540×10^{-2}	0.71		1.0168×10^{-1}	1.54	
	256	2.45×10^{-2}	3.3087×10^{-2}	0.94		8.4574×10^{-2}	0.27	
	512	1.22×10^{-2}	1.6187×10^{-2}	1.03		4.1020×10^{-2}	1.04	
	1024	6.13×10^{-3}	8.0014×10^{-3}	1.01		1.7976×10^{-2}	1.19	
	2048	3.06×10^{-3}	3.8168×10^{-3}	1.07		6.5705×10^{-3}	1.45	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	9.7360×10^{-2}	–	0.67, 0.98	1.5537×10^{-1}	–	0.85, 1.00
	64	9.82×10^{-2}	7.0146×10^{-2}	0.42		9.9996×10^{-2}	0.64	
	128	4.91×10^{-2}	4.9175×10^{-3}	0.51		6.0211×10^{-2}	0.73	
	256	2.45×10^{-2}	3.5404×10^{-3}	0.47		2.6342×10^{-2}	1.19	
	512	1.22×10^{-2}	2.0327×10^{-2}	0.80		1.5461×10^{-2}	0.77	
	1024	6.13×10^{-3}	1.1164×10^{-2}	0.86		8.8542×10^{-3}	0.80	
	2048	3.06×10^{-3}	5.7269×10^{-3}	0.96		4.9815×10^{-3}	0.83	

Table 6.7. Error norm data for fluid inside the sphere case using Approach B considered in Sec. 2.7.5

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	16	3.92×10^{-1}	1.2034	–	1.96, 0.91	1.2034	–	1.97, 0.93
	32	1.96×10^{-1}	8.3704×10^{-1}	3.85		9.8016×10^{-2}	3.62	
	64	9.82×10^{-2}	4.8865×10^{-1}	0.78		4.9230×10^{-2}	0.99	
	128	4.91×10^{-2}	1.0000×10^{-2}	2.29		1.0273×10^{-2}	2.26	
	256	2.45×10^{-2}	8.7930×10^{-3}	0.19		8.7922×10^{-3}	0.22	
	320	1.96×10^{-2}	9.7462×10^{-4}	9.86		1.0631×10^{-3}	9.47	
$\mathcal{E}^\infty$	16	3.92×10^{-1}	9.7340×10^{-2}	–	1.83, 0.93	1.0115×10^{-1}	–	1.51, 0.98
	32	1.96×10^{-1}	7.2565×10^{-3}	3.75		2.1147×10^{-2}	2.26	
	64	9.82×10^{-2}	4.9534×10^{-3}	0.55		1.0438×10^{-2}	1.02	
	128	4.91×10^{-2}	1.1726×10^{-3}	2.08		2.5863×10^{-3}	2.01	
	256	2.45×10^{-2}	7.4807×10^{-4}	0.65		1.5037×10^{-3}	0.78	
	320	1.96×10^{-2}	1.4668×10^{-4}	7.30		8.9006×10^{-4}	2.35	

Table 6.8. Error norm data for fluid inside the sphere case using Approach C considered in Sec. 2.7.5

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	16	3.92×10^{-1}	1.2034	–	0.94, 0.74	1.2034	–	1.97, 0.93
	32	1.96×10^{-1}	1.4493×10^{-1}	3.05		1.0073×10^{-1}	3.58	
	64	9.82×10^{-2}	1.2962×10^{-1}	0.16		4.9176×10^{-2}	1.03	
	128	4.91×10^{-2}	3.4356×10^{-2}	1.92		1.0362×10^{-2}	2.25	
	256	2.45×10^{-2}	4.9902×10^{-2}	-0.54		8.7919×10^{-3}	0.24	
	320	1.96×10^{-2}	5.6696×10^{-2}	-0.57		1.0625×10^{-3}	9.47	
$\mathcal{E}^\infty$	16	3.92×10^{-1}	1.5668×10^{-1}	–	0.07, 0.05	1.0083×10^{-1}	–	1.52, 0.99
	32	1.96×10^{-1}	4.9131×10^{-2}	1.67		2.3972×10^{-2}	2.07	
	64	9.82×10^{-2}	1.0494×10^{-1}	-1.09		9.9420×10^{-3}	1.27	
	128	4.91×10^{-2}	6.0320×10^{-2}	0.80		2.7054×10^{-3}	1.88	
	256	2.45×10^{-2}	8.8743×10^{-2}	-0.56		1.4489×10^{-3}	0.90	
	320	1.96×10^{-2}	9.3188×10^{-2}	-0.22		9.2746×10^{-4}	2.00	

Table 6.9. Error norm data for fluid outside the sphere case using Approach B considered in Sec. 2.7.5

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	16	3.92×10^{-1}	6.3869	–	2.00, 1.00	12.6388	–	2.17, 1.00
	32	1.96×10^{-1}	1.6017	2.00		2.3072	2.45	
	64	9.82×10^{-2}	4.0250×10^{-1}	1.99		3.9686×10^{-1}	2.54	
	128	4.91×10^{-2}	1.0047×10^{-1}	2.00		1.0267×10^{-1}	1.95	
	256	2.45×10^{-2}	2.5157×10^{-2}	2.00		2.8038×10^{-2}	1.87	
	320	1.96×10^{-2}	1.6090×10^{-2}	2.00		1.7741×10^{-2}	2.05	
$\mathcal{E}^\infty$	16	3.92×10^{-1}	3.8521×10^{-2}	–	2.00, 1.00	1.3960×10^{-1}	–	1.66, 0.97
	32	1.96×10^{-1}	9.6372×10^{-3}	2.00		3.6464×10^{-2}	1.94	
	64	9.82×10^{-2}	2.4095×10^{-3}	2.00		5.3588×10^{-3}	2.77	
	128	4.91×10^{-2}	6.0239×10^{-4}	2.00		3.1807×10^{-3}	0.75	
	256	2.45×10^{-2}	1.5059×10^{-4}	2.00		1.0506×10^{-3}	1.60	
	320	1.96×10^{-2}	9.6382×10^{-5}	2.00		9.9368×10^{-4}	0.25	

Table 6.10. Error norm data for fluid outside the sphere case using Approach C considered in Sec. 2.7.5

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	16	3.92×10^{-1}	11.4085	–	0.87, 0.90	11.3418	–	2.13, 1.00
	32	1.96×10^{-1}	3.3852	1.75		1.9539	2.54	
	64	9.82×10^{-2}	1.9619	0.79		3.8731×10^{-1}	2.33	
	128	4.91×10^{-2}	8.3413×10^{-1}	1.23		9.6877×10^{-2}	2.00	
	256	2.45×10^{-2}	6.6588×10^{-1}	0.33		3.1445×10^{-2}	1.62	
	320	1.96×10^{-2}	8.8538×10^{-1}	-1.28		1.5689×10^{-2}	3.12	
$\mathcal{E}^\infty$	16	3.92×10^{-1}	3.7012×10^{-1}	–	0.41, 0.67	1.1447×10^{-1}	–	1.60, 0.98
	32	1.96×10^{-1}	1.4986×10^{-1}	1.30		3.0673×10^{-2}	1.90	
	64	9.82×10^{-2}	1.4881×10^{-1}	0.01		5.9817×10^{-3}	2.36	
	128	4.91×10^{-2}	6.7817×10^{-2}	1.13		2.8228×10^{-3}	1.08	
	256	2.45×10^{-2}	9.2122×10^{-2}	-0.44		1.1423×10^{-3}	1.31	
	320	1.96×10^{-2}	1.0649×10^{-1}	-0.65		9.3613×10^{-4}	0.89	

Table 6.11. Error norm data for the torus case using Approach C considered in Sec. 2.7.5

‘

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	16	3.92×10^{-1}	2.0494	–	1.25, 0.93	2.7261	–	1.45, 0.97
	32	1.96×10^{-1}	5.1628×10^{-1}	1.99		1.2559	1.12	
	64	9.82×10^{-2}	1.1226×10^{-1}	2.20		2.1936×10^{-1}	2.52	
	128	4.91×10^{-2}	8.5599×10^{-2}	0.39		8.7687×10^{-2}	1.32	
	256	2.45×10^{-2}	4.7889×10^{-2}	0.84		5.2780×10^{-2}	0.73	
	320	1.96×10^{-2}	4.0430×10^{-2}	0.76		3.9756×10^{-2}	1.27	
$\mathcal{E}^\infty$	16	3.92×10^{-1}	7.1337×10^{-2}	–	0.63, 0.59	1.8828×10^{-1}	–	0.95, 0.90
	32	1.96×10^{-1}	1.6299×10^{-2}	2.13		9.5677×10^{-2}	0.98	
	64	9.82×10^{-2}	4.7456×10^{-2}	1.78		2.5818×10^{-2}	1.89	
	128	4.91×10^{-2}	8.9841×10^{-3}	-0.92		1.2753×10^{-2}	1.01	
	256	2.45×10^{-2}	7.6094×10^{-3}	0.24		1.4638×10^{-2}	-0.20	
	320	1.96×10^{-2}	6.7303×10^{-3}	0.55		1.1201×10^{-2}	1.20	

Table 6.12. Error norm data for the concentric annulus case with spatially constant Robin boundary conditions using Approach B considered in Sec. 2.7.6

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	5.9218×10^{-1}	–	1.97, 1.00	3.1462	–	1.67, 0.95
	64	9.82×10^{-2}	1.5389×10^{-1}	1.94		1.5899	0.98	
	128	4.91×10^{-2}	3.9792×10^{-2}	1.95		4.3190×10^{-1}	1.88	
	256	2.45×10^{-2}	1.0035×10^{-2}	1.99		4.4843×10^{-2}	3.27	
	512	1.22×10^{-2}	2.5860×10^{-3}	1.96		4.9145×10^{-2}	-0.13	
	1024	6.13×10^{-3}	6.7235×10^{-4}	1.94		2.4537×10^{-2}	1.00	
	2048	3.06×10^{-3}	1.6595×10^{-4}	2.02		2.0854×10^{-3}	3.56	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	7.3004×10^{-2}	–	1.97, 1.00	3.3663×10^{-1}	–	1.45, 0.99
	64	9.82×10^{-2}	1.9263×10^{-2}	1.92		1.7209×10^{-2}	0.97	
	128	4.91×10^{-2}	4.9511×10^{-3}	1.96		5.1747×10^{-2}	1.73	
	256	2.45×10^{-2}	1.2529×10^{-3}	1.98		1.4644×10^{-2}	1.82	
	512	1.22×10^{-2}	3.1817×10^{-4}	1.98		7.3577×10^{-3}	0.99	
	1024	6.13×10^{-3}	8.2175×10^{-5}	1.95		3.2291×10^{-3}	1.19	
	2048	3.06×10^{-3}	2.0329×10^{-5}	2.02		7.7661×10^{-4}	2.06	

Table 6.13. Error norm data for the concentric annulus case with spatially constant Robin boundary condition using Approach C considered in Sec. 2.7.6

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	1.2927	–	0.92, 0.56	3.2195	–	1.68, 0.95
	64	9.82×10^{-2}	1.1437×10^{-1}	3.50		1.6086	1.00	
	128	4.91×10^{-2}	2.8402×10^{-2}	2.01		4.3717×10^{-1}	1.88	
	256	2.45×10^{-2}	2.2619×10^{-1}	-2.99		4.5382×10^{-2}	3.27	
	512	1.22×10^{-2}	4.7478×10^{-2}	2.25		4.9493×10^{-2}	-0.13	
	1024	6.13×10^{-3}	3.8797×10^{-3}	3.61		2.4447×10^{-2}	1.02	
	2048	3.06×10^{-3}	2.6897×10^{-2}	-2.79		2.0692×10^{-3}	3.56	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	1.5846×10^{-1}	–	0.71, 0.48	3.3871×10^{-1}	–	1.45, 0.99
	64	9.82×10^{-2}	2.1296×10^{-2}	2.88		1.7262×10^{-2}	0.97	
	128	4.91×10^{-2}	7.1544×10^{-3}	1.57		5.1901×10^{-2}	1.73	
	256	2.45×10^{-2}	5.0560×10^{-2}	-2.82		1.4598×10^{-2}	1.83	
	512	1.22×10^{-2}	1.2498×10^{-2}	2.02		7.3909×10^{-3}	0.98	
	1024	6.13×10^{-3}	1.2720×10^{-3}	3.30		3.2262×10^{-3}	1.20	
	2048	3.06×10^{-3}	8.6497×10^{-3}	-2.77		7.7599×10^{-4}	2.06	

Table 6.14. Error norm data for the hexagram case with spatially varying Robin boundary conditions using Approach C considered in Sec. 2.7.7

	N	h	Continuous indicator function			Discontinuous indicator function		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	6.4450e-1	–	0.75, 0.96	8.4947e-1	–	1.00, 0.96
	64	9.82×10^{-2}	3.8669×10^{-1}	0.74		5.5409e-1	0.62	
	128	4.91×10^{-2}	2.1738×10^{-1}	0.83		3.7542×10^{-1}	0.56	
	256	2.45×10^{-2}	8.3523×10^{-2}	1.38		8.0590×10^{-2}	2.21	
	512	1.22×10^{-2}	6.1208×10^{-2}	0.45		9.1875×10^{-2}	-0.19	
	1024	6.13×10^{-3}	4.2044×10^{-2}	0.54		3.0769×10^{-2}	1.58	
	2048	3.06×10^{-3}	3.4124×10^{-2}	0.30		1.3971×10^{-3}	1.14	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	1.6739×10^{-1}	–	0.71, 0.95	1.8868×10^{-1}	–	0.72, 0.95
	64	9.82×10^{-2}	1.2126×10^{-1}	0.47		1.4155×10^{-1}	0.41	
	128	4.91×10^{-2}	1.0764×10^{-1}	0.17		1.2363×10^{-1}	0.20	
	256	2.45×10^{-2}	3.4357×10^{-2}	1.65		4.2007×10^{-2}	1.56	
	512	1.22×10^{-2}	3.7526×10^{-2}	-0.13		4.2895×10^{-2}	-0.03	
	1024	6.13×10^{-3}	1.5288×10^{-2}	1.30		1.8924×10^{-2}	1.18	
	2048	3.06×10^{-3}	9.5637×10^{-3}	0.68		9.5005×10^{-3}	0.99	

Table 6.15. Error norm data for the hexagram with smooth exterior corners case using Approach C with the discontinuous indicator function considered in Sec. 2.7.8

	N	h	Neumann problem			Robin problem		
			Error	Order	Fit (m, R^2)	Error	Order	Fit (m, R^2)
$\mathcal{E}^1$	32	1.96×10^{-1}	7.6209×10^{-1}	–	1.08, 0.91	3.9049e-1	–	1.26, 0.89
	64	9.82×10^{-2}	1.4221×10^{-1}	2.42		1.0627e-1	1.88	
	128	4.91×10^{-2}	1.4759×10^{-1}	-0.05		1.0857×10^{-1}	-0.03	
	256	2.45×10^{-2}	2.2453×10^{-2}	2.72		1.4303×10^{-2}	2.92	
	512	1.22×10^{-2}	1.4344×10^{-2}	0.65		2.9291×10^{-3}	2.29	
	1024	6.13×10^{-3}	1.5886×10^{-2}	-0.15		4.7328×10^{-3}	-0.69	
	2048	3.06×10^{-3}	6.7013×10^{-3}	1.25		2.9825×10^{-3}	0.67	
$\mathcal{E}^\infty$	32	1.96×10^{-1}	2.3668×10^{-1}	–	0.95, 0.95	1.2099×10^{-1}	–	1.00, 0.88
	64	9.82×10^{-2}	4.8712×10^{-2}	2.28		4.0233×10^{-2}	1.59	
	128	4.91×10^{-2}	4.3138×10^{-2}	0.18		4.9885×10^{-2}	-0.31	
	256	2.45×10^{-2}	2.5443×10^{-2}	0.76		1.0488×10^{-2}	2.25	
	512	1.22×10^{-2}	7.8196×10^{-3}	1.70		3.4723×10^{-3}	1.59	
	1024	6.13×10^{-3}	6.8699×10^{-3}	0.19		1.9522×10^{-3}	0.83	
	2048	3.06×10^{-3}	3.3329×10^{-3}	1.04		3.4365×10^{-3}	-0.82	

Table 6.16. Error norm data for the circle case with spatially varying Neumann boundary conditions using Approach C and D considered in Appendix 6.1. The continuous indicator function is used here. A convergence rate of $\mathcal{O}(h^{0.98})$ (respectively, $\mathcal{O}(h^{0.95})$) with an R^2 value of 0.95 (respectively, 0.98) in the L^∞ (respectively, L^1) norm is obtained using Approach C. In the case of Approach D with the top hat kernel function, a convergence rate of $\mathcal{O}(h^{-0.02})$ (respectively, $\mathcal{O}(h^{-0.01})$) with an R^2 value of 0.19 (respectively, 0.03) in the L^∞ (respectively, L^1) norm is obtained. With the spline kernel function, Approach D achieves a convergence rate of $\mathcal{O}(h^{-0.01})$ (respectively, $\mathcal{O}(h^{0.02})$) with an R^2 value of 0.69 (respectively, 0.55) in the L^∞ (respectively, L^1) norm.

	N	Approach C		Approach D (top hat function)		Approach D (spline function)	
		Error	Order	Error	Order	Error	Order
$\mathcal{E}^1$	32	5.0137×10^{-2}	–	5.9980×10^{-1}	–	1.3436	–
	64	1.6119×10^{-2}	1.64	5.6372×10^{-1}	0.9	1.2869	0.06
	128	1.3963×10^{-2}	0.21	6.4900×10^{-1}	-0.20	1.2772	0.01
	256	4.6304×10^{-3}	1.59	6.1927×10^{-1}	0.07	1.2737	4.0×10^{-3}
	512	3.4634×10^{-3}	0.42	6.0978×10^{-1}	0.02	1.2729	9.1×10^{-4}
	1024	1.5665×10^{-3}	1.14	6.0404×10^{-1}	0.01	1.2720	1.1×10^{-3}
	2048	8.1497×10^{-4}	0.94	6.0404×10^{-1}	0.00	1.2714	6.5×10^{-4}
$\mathcal{E}^\infty$	32	8.4450×10^{-3}	–	9.0222×10^{-2}	–	2.1357×10^{-1}	–
	64	3.7481×10^{-3}	1.17	1.0070×10^{-1}	-0.16	2.1124×10^{-1}	0.02
	128	5.0424×10^{-3}	-0.43	1.1195×10^{-1}	-0.15	2.1110×10^{-1}	0.01
	256	1.0171×10^{-3}	2.31	1.0500×10^{-1}	0.09	2.1627×10^{-1}	9.6×10^{-4}
	512	6.5633×10^{-4}	0.63	1.0480×10^{-1}	2.7×10^{-3}	2.1855×10^{-1}	-0.03
	1024	3.1095×10^{-4}	1.08	1.0284×10^{-1}	0.03	2.1849×10^{-1}	-0.02
	2048	1.5776×10^{-4}	0.98	1.0284×10^{-1}	0.00	2.1842×10^{-1}	4.6×10^{-4}

Bibliography

[1] T. Sakurai, K. Yoshimatsu, N. Okamoto, K. Schneider, Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry, Journal of Computational Physics 390 (2019) 452–469.

[2] A. Pathak, M. Raessi, A 3d, fully eulerian, vof-based solver to study the interaction between two fluids and moving rigid bodies using the fictitious domain method, Journal of Computational Physics 311 (2016) 87–113.

[3] E.-M. Yettou, A. Desrochers, Y. Champoux, Experimental study on the water impact of a symmetrical wedge, Fluid Dynamics Research 38 (1) (2006) 47.

[4] N. Nangia, N. A. Patankar, A. P. S. Bhalla, A DLM immersed boundary method based wave-structure interaction solver for high density ratio multiphase flows, Journal of Computational Physics 398 (2019) 108804.

[5] P. S. Cook, A. B. Murphy, Simulation of melt pool behaviour during additive manufacturing: Underlying physics and progress, Additive Manufacturing 31 (2020) 100909.

[6] B. E. Griffith, N. A. Patankar, Immersed methods for fluid-structure interaction, Annual Review of Fluid Mechanics 52 (2020) 421–448.

[7] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numerische Mathematik 81 (4) (1999) 497–520.

[8] I. Ramière, P. Angot, M. Belliard, A fictitious domain approach with spread interface for elliptic problems with general boundary conditions, Computer Methods in Applied Mechanics and Engineering 196 (4–6) (2007) 766–781.

[9] B. Kadoch, D. Kolomenskiy, P. Angot, K. Schneider, A volume penalization method for incompressible flows and scalar advection–diffusion with moving obstacles, Journal of Computational Physics 231 (12) (2012) 4365–4383.

[10] M. Gazzola, P. Chatelain, W. M. Van Rees, P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries, Journal of Computational Physics 230 (19) (2011) 7093–7114.

[11] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, Journal of Computational Physics 230 (2) (2011) 329–348.

[12] T. Engels, D. Kolomenskiy, K. Schneider, J. Sesterhenn, Numerical simulation of fluid–structure interaction with the volume penalization method, Journal of Computational Physics 281 (2015) 96–115.

[13] K. Khedkar, N. Nangia, R. Thirumalaisamy, A. P. S. Bhalla, The inertial sea wave energy converter (ISWEC) technology: Device-physics, multiphase modeling and simulations, Ocean Engineering 229 (2021) 108879.

[14] K. Khedkar, A. P. S. Bhalla, A model predictive control (MPC)-integrated multiphase immersed boundary (IB) framework for simulating wave energy converters (WECs), Ocean Engineering 260 (2022) 111908. `doi:10.1016/j.oceaneng.2022.111908`. URL https://www.sciencedirect.com/science/article/pii/S0029801822012471

[15] E. L. Sharaborin, O. A. Rogozin, A. R. Kasimov, The coupled volume of fluid and Brinkman penalization methods for simulation of incompressible multiphase flows, Fluids 6 (9) (2021) 334.

[16] M. Bergmann, Numerical modeling of a self-propelled dolphin jump out of water, Bioinspiration & Biomimetics 17 (6) (2022) 065010.

[17] P. C. Carman, Fluid flow through granular beds, Transactions of the Institute of Chemical Engineers 15 (1937) 150–166.

[18] V. R. Voller, C. Prakash, A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems, International journal of heat and mass transfer 30 (8) (1987) 1709–1719.

[19] Z. Huang, G. Lin, A. M. Ardekani, A consistent and conservative phase-field model for thermo-gas-liquid-solid flows including liquid-solid phase change, Journal of Computational Physics 449 (2022) 110795.

[20] A. El Khadraoui, S. Bouadila, S. Kooli, A. Farhat, A. Guizani, Thermal behavior of indirect solar dryer: Nocturnal usage of solar air collector with PCM, Journal of cleaner production 148 (2017) 37–48.

[21] A. Allouhi, A. A. Msaad, M. B. Amine, R. Saidur, M. Mahdaoui, T. Kousksou, A. Pandey, A. Jamil, N. Moujibi, A. Benbassou, Optimization of melting and solidification processes of PCM: Application to integrated collector storage solar water heaters (ICSSWH), Solar Energy 171 (2018) 562–570.

[22] Z. Badiei, M. Eslami, K. Jafarpur, Performance improvements in solar flat plate collectors by integrating with phase change materials and fins: A CFD modeling, Energy 192 (2020) 116719.

[23] M. Hossain, A. Pandey, J. Selvaraj, N. Abd Rahim, M. Islam, V. Tyagi, Two side serpentine flow based photovoltaic-thermal-phase change materials (PVT-PCM) system: Energy, exergy and economic analysis, Renewable Energy 136 (2019) 1320–1336.

[24] C. Nie, S. Deng, J. Liu, Numerical investigation of PCM in a thermal energy storage unit with fins: consecutive charging and discharging, J. Energy Storage 29 (2020) 101319.

[25] J. Buffo, C. Meyer, J. Parkinson, Dynamics of a Solidifying Icy Satellite Shell, Journal of Geophysical Research: Planets 126 (5) (2021) e2020JE006741.

[26] J. Buffo, B. Schmidt, C. Huber, C. Meyer, Characterizing the ice-ocean interface of icy worlds: A theoretical approach, Icarus 360 (2021) 114318.

[27] W. E. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, A. M. Rubenchik, Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges, Applied Physics Reviews 2 (4) (2015) 041304.

[28] W. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, Overview of modelling and simulation of metal powder bed fusion process at Lawrence Livermore National Laboratory, Materials Science and Technology 31 (8) (2015) 957–968.

[29] S. A. Khairallah, A. T. Anderson, A. Rubenchik, W. E. King, Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones, Acta Materialia 108 (2016) 36–45.

[30] S. Ly, A. M. Rubenchik, S. A. Khairallah, G. Guss, M. J. Matthews, Metal vapor micro-jet controls material redistribution in laser powder bed fusion additive manufacturing, Scientific reports 7 (1) (2017) 1–12.

[31] C. Panwisawas, C. Qiu, M. J. Anderson, Y. Sovani, R. P. Turner, M. M. Attallah, J. W. Brooks, H. C. Basoalto, Mesoscale modelling of selective laser melting: Thermal fluid dynamics and microstructural evolution, Computational Materials Science 126 (2017) 479–490.

[32] Y.-C. Wu, C.-H. San, C.-H. Chang, H.-J. Lin, R. Marwan, S. Baba, W.-S. Hwang, Numerical modeling of melt-pool behavior in selective laser melting with random powder distribution and experimental validation, Journal of Materials Processing Technology 254 (2018) 72–78.

[33] Y.-C. Wu, W.-S. Hwang, C.-H. San, C.-H. Chang, H.-J. Lin, Parametric study of surface morphology for selective laser melting on Ti6Al4V powder bed with numerical and experimental methods, International Journal of Material Forming 11 (6) (2018) 807–813.

[34] F.-J. Gürtler, M. Karg, K.-H. Leitz, M. Schmidt, Simulation of laser beam melting of steel powders using the three-dimensional volume of fluid method, Physics Procedia 41 (2013) 881–886.

[35] E. Attar, C. Körner, Lattice Boltzmann model for thermal free surface flows with liquid–solid phase transition, International Journal of Heat and Fluid Flow 32 (1) (2011) 156–163.

[36] C. Panwisawas, B. Perumal, R. M. Ward, N. Turner, R. P. Turner, J. W. Brooks, H. C. Basoalto, Keyhole formation and thermal fluid flow-induced porosity during laser fusion welding in titanium alloys: Experimental and modelling, Acta Materialia 126 (2017) 251–263.

[37] A. Aggarwal, A. Kumar, Particle scale modelling of selective laser melting-based additive manufacturing process using open-source CFD code OpenFOAM, Transactions of the Indian Institute of Metals 71 (11) (2018) 2813–2817.

[38] C. S. Peskin, The immersed boundary method, Acta Numer 11 (2002) 479–517.

[39] M. Bergmann, G. Bracco, F. Gallizio, A. Giorcelli, E. Iollo, G. Mattiazzo, M. Ponzetta, A two-way coupling CFD method to simulate the dynamics of a wave energy converter., in: OCEANS 2015 - Genova, Italy, IEEE, pp. 1–6.

[40] P. Dafnakis, A. P. S. Bhalla, S. A. Sirigu, M. Bonfanti, G. Bracco, G. Mattiazzo, Comparison of wave–structure interaction dynamics of a submerged cylindrical point absorber with three degrees of freedom using potential flow and computational fluid dynamics models, Physics of Fluids 32 (9) (2020) 093307.

[41] A. P. S. Bhalla, N. Nangia, P. Dafnakis, G. Bracco, G. Mattiazzo, Simulating water-entry/exit problems using Eulerian-Lagrangian and fully-Eulerian fictitious domain methods within the open-source IBAMR library, Applied Ocean Research 94 (2020) 101932.

[42] A. P. S. Bhalla, R. Bale, B. E. Griffith, N. A. Patankar, A unified mathematical framework and an adaptive numerical method for fluid–structure interaction with rigid, deforming, and elastic bodies, Journal of Computational Physics 250 (2013) 446–476.

[43] W. Kou, A. P. S. Bhalla, B. E. Griffith, J. E. Pandolfino, P. J. Kahrilas, N. A. Patankar, A fully resolved active musculo-mechanical model for esophageal transport, Journal of computational physics 298 (2015) 446–465.

[44] B. E. Griffith, Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions, International Journal for Numerical Methods in Biomedical Engineering 28 (3) (2012) 317–345.

[45] C. S. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[46] R. Thirumalaisamy, N. Nangia, A. P. S. Bhalla, Critique on "Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry", Journal of Computational Physics 433 (2021) 110163.

[47] T. Sakurai, K. Yoshimatsu, N. Okamoto, K. Schneider, Corrigendum to "Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry" [J. Comput. Phys. 390 (2019) 452–469], Journal of Computational Physics (2021) 110497`doi:https://doi.org/10.1016/j.jcp.2021.`

`110497`.
URL https://www.sciencedirect.com/science/article/pii/S0021999121003922

[48] J. A. Bærentzen, H. Aanaes, Signed distance computation using the angle weighted pseudonormal, IEEE Transactions on Visualization and Computer Graphics 11 (3) (2005) 243–253.

[49] X. Li, J. Lowengrub, A. Rätz, A. Voigt, Solving PDEs in complex geometries: a diffuse domain approach, Communications in mathematical sciences 7 (1) (2009) 81.

[50] E. Brown-Dymkoski, N. Kasimov, O. V. Vasilyev, A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows, Journal of Computational Physics 262 (2014) 344–357.

[51] B. Hardy, J. De Wilde, G. Winckelmans, A penalization method for the simulation of weakly compressible reacting gas-particle flows with general boundary conditions, Computers & Fluids 190 (2019) 294–307.

[52] B. Bensiali, G. Chiavassa, J. Liandrat, Penalization of Robin boundary conditions, Applied Numerical Mathematics 96 (2015) 134–152.

[53] D. Kolomenskiy, R. Nguyen van yen, K. Schneider, Analysis and discretization of the volume penalized Laplace operator with Neumann boundary conditions, Applied Numerical Mathematics 95 (2015) 238–249.

[54] K. Schneider, Immersed boundary methods for numerical simulation of confined fluid and plasma turbulence in complex geometries: a review, Journal of Plasma Physics 81 (6) (2015).

[55] W. Ren, C. Shu, W. Yang, An efficient immersed boundary method for thermal flow problems with heat flux boundary conditions, International Journal of Heat and Mass Transfer 64 (2013) 694–705.

[56] Y. Wang, C. Shu, L. M. Yang, Boundary condition-enforced immersed boundary-lattice Boltzmann flux solver for thermal flows with Neumann boundary conditions, Journal of Computational Physics 306 (2016) 237–252.

[57] T. Guo, E. Shen, Z. Lu, Y. Wang, L. Dong, Implicit heat flux correction-based immersed boundary-finite volume method for thermal flows with Neumann boundary conditions, Journal of Computational Physics 386 (2019) 64–83.

[58] J. Lou, J. Johnston, N. Tilton, Application of projection and immersed boundary methods to simulating heat and mass transport in membrane distillation, Computers & Fluids 212 (2020) 104711.

[59] N. Nangia, B. E. Griffith, N. A. Patankar, A. P. S. Bhalla, A robust incompressible Navier-Stokes solver for high density ratio multiphase flows, Journal of Computational Physics 390 (2019) 548–594.

[60] IBAMR: An adaptive and distributed-memory parallel implementation of the immersed boundary method, https://github.com/IBAMR/IBAMR.

[61] R. D. Hornung, S. R. Kohn, Managing application complexity in the SAMRAI object-oriented framework, Concurrency Comput Pract Ex 14 (5) (2002) 347–368.

[62] SAMRAI: Structured Adaptive Mesh Refinement Application Infrastructure, http://www.llnl.gov/CASC/SAMRAI.

[63] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[64] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.6, Argonne National Laboratory (2015).
URL http://www.mcs.anl.gov/petsc

[65] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, PETSc Web page, http://www.mcs.anl.gov/petsc (2015).
URL http://www.mcs.anl.gov/petsc

[66] J.-S. Yoo, Dual free-convective flows in a horizontal annulus with a constant heat flux wall, International Journal of Heat and Mass Transfer 46 (13) (2003) 2499–2503.

[67] E. Arquis, J. P. Caltagirone, Sur les conditions hydrodynamiques au voisinage dâune interface milieu fluide milieu poreux: application Ã la convection naturelle, Comptes Rendus de l'AcadÃ©mie des Sciences - Series IIB 299 (1984) 1–4.

[68] H. C. Brinkman, A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles, Applied Science Research 1 (1949) 27.

[69] R. Thirumalaisamy, N. A. Patankar, A. P. S. Bhalla, Handling Neumann and Robin boundary conditions in a fictitious domain volume penalization framework, Journal of Computational Physics 448 (2022) 110726.

[70] J. Kou, S. Joshi, A. Hurtado-de-Mendoza, K. Puri, C. Hirsch, E. Ferrer, Immersed boundary method for high-order flux reconstruction based on volume penalization, Journal of Computational Physics 448 (2022) 110721.

[71] Z. Xie, T. Stoesser, A three-dimensional cartesian cut-cell/volume-of-fluid method for two-phase flows with moving bodies, Journal of Computational Physics 416 (2020) 109536.

[72] M. van der Eijk, P. Wellens, Two-phase free-surface flow interaction with moving bodies using a consistent, momentum preserving method, Journal of Computational Physics 474 (2023) 111796.

[73] D. Kolomenskiy, K. Schneider, A fourier spectral method for the Navier–Stokes equations with volume penalization for moving solid obstacles, Journal of Computational Physics 228 (16) (2009) 5687–5709.

[74] M. Gazzola, O. V. Vasilyev, P. Koumoutsakos, Shape optimization for drag reduction in linked bodies using evolution strategies, Computers & Structures 89 (11-12) (2011) 1224–1231.

[75] H. Beaugendre, F. Morency, Penalization of the Spalart–Allmaras turbulence model without and with a wall function: Methodology for a vortex in cell scheme, Computers & Fluids 170 (2018) 313–323.

[76] D. Rossinelli, M. Bergdorf, G.-H. Cottet, P. Koumoutsakos, GPU accelerated simulations of bluff body flows using vortex particle methods, Journal of Computational Physics 229 (9) (2010) 3316–3333.

[77] B. E. Griffith, An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner, Journal of Computational Physics 228 (20) (2009) 7565–7595.

[78] M. Cai, A. Nonaka, J. B. Bell, B. E. Griffith, A. Donev, Efficient variable-coefficient finite-volume stokes solvers, Communications in Computational Physics 16 (5) (2014) 1263–1297.

[79] J. Ahlkrona, D. Elfverson, A cut finite element method for non-newtonian free surface flows in 2d-application to glacier modelling, Journal of Computational Physics: X 11 (2021) 100090.

[80] A. Löfgren, J. Ahlkrona, C. Helanow, Increasing stable time-step sizes of the free-surface problem arising in ice-sheet simulations, Journal of Computational Physics: X 16 (2022) 100114.

[81] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, Communications in Applied Mathematics and Computational Science 11 (1) (2016) 79–141.

[82] F. Balboa Usabiaga, B. Kallemov, B. Delmotte, A. Bhalla, B. Griffith, A. Donev, Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach, Communications in Applied Mathematics and Computational Science 11 (2) (2017) 217–296.

[83] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49.

[84] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational Physics 114 (1) (1994) 146–159.

[85] C. Zhang, C. Wu, K. Nandakumar, Effective geometric algorithms for immersed boundary method using signed distance field, Journal of Fluids Engineering 141 (6) (2019).

[86] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, The physics of fluids 8 (12) (1965) 2182–2189.

[87] J.-L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Computer methods in applied mechanics and engineering 195 (44-47) (2006) 6011–6045.

[88] S. Patankar, Numerical heat transfer and fluid flow, Taylor & Francis, 2018.

[89] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, Journal of Computational Physics 171 (2) (2001) 822–842.

[90] E. M. Kolahdouz, A. P. S. Bhalla, L. N. Scotten, B. A. Craven, B. E. Griffith, A sharp interface Lagrangian-Eulerian method for rigid-body fluid-structure interaction, Journal of computational physics 443 (2021) 110442.

[91] F. Gibou, R. P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the poisson equation on irregular domains, Journal of Computational Physics 176 (1) (2002) 205–227.

[92] W. J. Rider, J. A. Greenough, J. R. Kamm, Accurate monotonicity-and extrema-preserving methods through adaptive nonlinear hybridizations, Journal of Computational Physics 225 (2) (2007) 1827–1848.

[93] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced numerical approximation of nonlinear hyperbolic equations, Springer, 1998, pp. 325–432.

[94] G. Russo, P. Smereka, A remark on computing distance functions, Journal of Computational Physics 163 (1) (2000) 51–67.

[95] C. Min, On reinitializing level set functions, Journal of Computational Physics 229 (8) (2010) 2764–2772.

[96] A. A. Howard, A. M. Tartakovsky, A conservative level set method for N-phase flows with a free-energy-based surface tension model, Journal of Computational Physics 426 (2021) 109955.

[97] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J Sci Comput 14 (2) (1993) 461–469.

[98] D. L. Brown, R. Cortez, M. L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, Journal of Computational Physics 168 (2) (2001) 464–499.

[99] S. McCormick, J. Thomas, The fast adaptive composite grid (fac) method for elliptic equations, Mathematics of Computation 46 (174) (1986) 439–456.

[100] A. Hamed, Y. Jin, L. Chamorro, On the transient dynamics of the wake and trajectory of free falling cones with various apex angles, Experiments in Fluids 56 (2015).

[101] B. E. Griffith, On the volume conservation of the immersed boundary method, Communications in Computational Physics 12 (2) (2012) 401–432.

[102] R. Bale, A. P. S. Bhalla, B. E. Griffith, M. Tsubokura, A one-sided direct forcing immersed boundary method using moving least squares, Journal of Computational Physics 440 (2021) 110359.

[103] V. Chéron, F. Evrard, B. van Wachem, A hybrid immersed boundary method for dense particle-laden flows, Computers & Fluids 259 (2023) 105892.

[104] C. Huh, L. E. Scriven, Hydrodynamic model of steady movement of a solid/liquid/fluid contact line, Journal of colloid and interface science 35 (1) (1971) 85–101.

[105] Y. Zhang, Q. Zou, D. Greaves, D. Reeve, A. Hunt-Raby, D. Graham, P. James, X. Lv, A level set immersed boundary method for water entry and exit, Comm. Comput. Phys 8 (2) (2010) 265–288.

[106] J. K. Patel, G. Natarajan, Diffuse interface immersed boundary method for multi-fluid flows with arbitrarily moving rigid bodies, Journal of Computational Physics 360 (2018) 202–228.

[107] A. Calderer, S. Kang, F. Sotiropoulos, Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures, Journal of Computational Physics 277 (2014) 201–227.

[108] J. Sanders, J. E. Dolbow, P. J. Mucha, T. A. Laursen, A new method for simulating rigid body motion in incompressible two-phase flow, International Journal for Numerical Methods in Fluids 67 (6) (2011) 713–732.

[109] J.-P. Kruth, B. Van der Schueren, J. Bonse, B. Morren, Basic powder metallurgical aspects in selective metal powder sintering, CIRP annals 45 (1) (1996) 183–186.

[110] K. Karayagiz, A. Elwany, G. Tapia, B. Franco, L. Johnson, J. Ma, I. Karaman, R. Arróyave, Numerical and experimental analysis of heat distribution in the laser powder bed fusion of ti-6al-4v, IISE Transactions 51 (2) (2019) 136–152.

[111] T. Heeling, M. Cloots, K. Wegener, Melt pool simulation for the evaluation of process parameters in selective laser melting, Additive Manufacturing 14 (2017) 116–125.

[112] M. J. Matthews, G. Guss, S. A. Khairallah, A. M. Rubenchik, P. J. Depond, W. E. King, Denudation of metal powder layers in laser powder bed fusion processes, Acta Materialia 114 (2016) 33–42.

[113] N. Nangia, B. E. Griffith, N. A. Patankar, A. P. S. Bhalla, A robust incompressible Navier-Stokes solver for high density ratio multiphase flows, Journal of Computational Physics 390 (2019) 548–594.

[114] A. Pathak, M. Raessi, A 3D, fully Eulerian, VOF-based solver to study the interaction between two fluids and moving rigid bodies using the fictitious domain method, Journal of computational physics 311 (2016) 87–113.

[115] J. K. Patel, G. Natarajan, Diffuse interface immersed boundary method for multi-fluid flows with arbitrarily moving rigid bodies, Journal of Computational Physics 360 (2018) 202–228.

[116] S. J. Wolff, H. Wu, N. Parab, C. Zhao, K. F. Ehmann, T. Sun, J. Cao, In-situ high-speed x-ray imaging of piezo-driven directed energy deposition additive manufacturing, Scientific reports 9 (1) (2019) 1–14.

[117] M. Herrmann, J. Lopez, P. Brady, M. Raessi, Thermocapillary motion of deformable drops and bubbles, in: Proceedings of the Summer program, 2008, p. 155.

[118] K. Ali, S. Bilal, S. Siddiqi, et al., Concentration and temperature dependence of surface parameters of some aqueous salt solutions, Colloids and Surfaces A: Physicochemical and Engineering Aspects 272 (1-2) (2006) 105–110.

[119] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (2) (1992) 335–354.

[120] S. Nas, G. Tryggvason, Thermocapillary interaction of two bubbles or drops, International journal of multiphase flow 29 (7) (2003) 1117–1135.

[121] C. Ma, D. Bothe, Direct numerical simulation of thermocapillary flow based on the volume of fluid method, International Journal of Multiphase Flow 37 (9) (2011) 1045–1058.

[122] I. Seric, S. Afkhami, L. Kondic, Direct numerical simulation of variable surface tension flows using a volume-of-fluid method, Journal of Computational Physics 352 (2018) 615–636.

[123] M. K. Tripathi, K. C. Sahu, Motion of an air bubble under the action of thermocapillary and buoyancy forces, Computers & Fluids 177 (2018) 58–68.

[124] V. Voller, C. Prakash, A fixed grid numerical modeling methodology for convection diffusion mushy region phase-change problem, International Journal Heat Mass Transfer 30 (9) (1987) 1709–1719.

[125] V. R. Voller, C. Swaminathan, ERAL Source-based method for solidification phase change, Numerical Heat Transfer, Part B Fundamentals 19 (2) (1991) 175–189.

[126] P. Galione, O. Lehmkuhl, J. Rigola, A. Oliva, Fixed-grid numerical modeling of melting and solidification using variable thermo-physical properties–application to the melting of n-octadecane inside a spherical capsule, International Journal of Heat and Mass Transfer 86 (2015) 721–743.

[127] M. Hassab, M. M. Sorour, M. K. Mansour, M. M. Zaytoun, Effect of volume expansion on the melting process's thermal behavior, Applied Thermal Engineering 115 (2017) 350–362.

[128] J. Dallaire, L. Gosselin, Numerical modeling of solid-liquid phase change in a closed 2d cavity with density change, elastic wall and natural convection, International Journal of Heat and Mass Transfer 114 (2017) 903–914.

[129] M. Faden, A. König-Haagen, D. Brüggemann, An optimum enthalpy approach for melting and solidification with volume change, Energies 12 (5) (2019) 868.

[130] J. Yan, W. Yan, S. Lin, G. Wagner, A fully coupled finite element formulation for liquid–solid–gas thermo-fluid flow with melting and solidification, Computer Methods in Applied Mechanics and Engineering 336 (2018) 444–470.

[131] S. Lin, Z. Gan, J. Yan, G. J. Wagner, A conservative level set method on unstructured meshes for modeling multiphase thermo-fluid flow in additive manufacturing processes, Computer Methods in Applied Mechanics and Engineering 372 (2020) 113348.

[132] T. G. Myers, M. G. Hennessy, M. Calvo-Schwarzwälder, The Stefan problem with variable thermophysical properties and phase change temperature, International Journal of Heat and Mass Transfer 149 (2020) 118975.

[133] J.-M. Delhaye, Jump conditions and entropy sources in two-phase systems. Local instant formulation, International Journal of Multiphase Flow 1 (3) (1974) 395–409.

[134] V. Alexiades, A. D. Solomon, Mathematical modeling of melting and freezing processes, Routledge, 2018.

[135] Z. Huang, G. Lin, A. M. Ardekani, A consistent and conservative phase-field model for thermo-gas-liquid-solid flows including liquid-solid phase change, Journal of Computational Physics 449 (2022) 110795.

[136] E. Javierre, C. Vuik, F. Vermolen, S. Van der Zwaag, A comparison of numerical models for one-dimensional Stefan problems, Journal of Computational and Applied Mathematics 192 (2) (2006) 445–459.

[137] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, Journal of Computational Physics 222 (2) (2007) 536–555.

[138] M. Khalloufi, R. Valette, E. Hachem, Adaptive eulerian framework for boiling and evaporation, Journal of Computational Physics 401 (2020) 109030.

[139] D. W. Hahn, M. N. Özisik, Heat conduction, John Wiley & Sons, 2012.

[140] W. J. Boettinger, J. A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Annual review of materials research 32 (1) (2002) 163–194.

[141] H. Hu, S. A. Argyropoulos, Mathematical modelling of solidification and melting: a review, Modelling and Simulation in Materials Science and Engineering 4 (4) (1996) 371.

[142] F. Moukalled, L. Mangani, M. Darwish, The finite volume method in computational fluid dynamics, Vol. 113, Springer, 2016.

[143] F. Tan, S. Hosseinizadeh, J. Khodadadi, L. Fan, Experimental and computational study of constrained melting of phase change materials (PCM) inside a spherical capsule, International Journal of Heat and Mass Transfer 52 (15-16) (2009) 3464–3472.

[144] C. Beckermann, R. Viskanta, Natural convection solid/liquid phase change in porous media, International journal of heat and mass transfer 31 (1) (1988) 35–46.

[145] W. Fiveland, J. Jessee, An adaptive projection method for unsteady, low-mach number combustion, Combustion Science and Technology 140 (1-6) (1998) 123–168.

[146] S. A. Hosseini, N. Darabiha, D. Thévenin, Low mach number lattice boltzmann model for turbulent combustion: flow in confined geometries, Proceedings of the Combustion Institute (2022).

[147] J. Bell, M. Day, C. Rendleman, S. Woosley, M. Zingale, Adaptive low Mach number simulations of nuclear flame microphysics, Journal of Computational Physics 195 (2) (2004) 677–694.

[148] C. Gilet, A. Almgren, J. Bell, A. Nonaka, S. Woosley, M. Zingale, Low mach number modeling of core convection in massive stars, The Astrophysical Journal 773 (2) (2013) 137.

[149] A. Donev, A. Nonaka, Y. Sun, T. Fai, A. Garcia, J. Bell, Low mach number fluctuating hydrodynamics of diffusively mixing fluids, Communications in Applied Mathematics and Computational Science 9 (1) (2014) 47–105.

[150] A. Donev, A. Nonaka, A. K. Bhattacharjee, A. L. Garcia, J. B. Bell, Low mach number fluctuating hydrodynamics of multispecies liquid mixtures, Physics of Fluids 27 (3) (2015) 037103.

[151] A. Nonaka, Y. Sun, J. Bell, A. Donev, Low mach number fluctuating hydrodynamics of binary liquid mixtures, Communications in Applied Mathematics and Computational Science 10 (2) (2015) 163–204.

[152] D. M. Stefanescu, Science and engineering of casting solidification, Springer, 2015.

[153] V. Alexiades, J. Drake, A weak formulation for phase-change problems with bulk movement due to unequal densities, PITMAN RESEARCH NOTES IN MATHEMATICS SERIES (1993) 82–82.

[154] L. Durlofsky, J. Brady, Analysis of the Brinkman equation as a model for flow in porous media, The Physics of fluids 30 (11) (1987) 3329–3341.

[155] Z. Saldi, Marangoni driven free surface flows in liquid weld pools, Ph.D. thesis (2012).

[156] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, M. 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) (2006) 141–173.

[157] R. Thirumalaisamy, K. Khedkar, P. Ghysels, A. P. S. Bhalla, An effective preconditioning strategy for volume penalized incompressible/low mach multiphase flow solvers, Journal of Computational Physics 490 (2023) 112325. `doi:https://doi.org/10.1016/j.jcp.2023.112325`.
URL https://www.sciencedirect.com/science/article/pii/S0021999123004205

[158] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, et al., VisIt: An end-user tool for visualizing and analyzing very large data, Tech. rep. (2012).

[159] A. P. S. Bhalla, N. Nangia, P. Dafnakis, G. Bracco, G. Mattiazzo, Simulating water-entry/exit problems using eulerian–lagrangian and fully-eulerian fictitious domain methods within the open-source ibamr library, Applied Ocean Research 94 (2020) 101932.

[160] K. Khedkar, N. Nangia, R. Thirumalaisamy, A. P. S. Bhalla, The inertial sea wave energy converter (iswec) technology: Device-physics, multiphase modeling and simulations, Ocean Engineering 229 (2021) 108879.

[161] N. Young, J. S. Goldstein, M. Block, The motion of bubbles in a vertical temperature gradient, Journal of Fluid Mechanics 6 (3) (1959) 350–356.

[162] F. Denner, F. Evrard, B. van Wachem, Breaching the capillary time-step constraint using a coupled vof method with implicit surface tension, Journal of Computational Physics 459 (2022) 111128.

[163] J. K. Patel, G. Natarajan, A novel consistent and well-balanced algorithm for simulations of multiphase flows on unstructured grids, Journal of Computational Physics 350 (2017) 207–236.

[164] Y. Zeng, H. Liu, Q. Gao, A. Almgren, A. P. S. Bhalla, L. Shen, A consistent adaptive level set framework for incompressible two-phase flows with high density ratios and high reynolds numbers, Journal of Computational Physics 478 (2023) 111971.

[165] J. C. Martin, W. J. Moyce, Penney, Part iv. an experimental study of the collapse of liquid columns on a rigid horizontal plane, Phil. Trans. R. Soc. Lond. A 244 (882) (1952) 312–324.

[166] Z. Solomenko, P. D. Spelt, L. O. Naraigh, P. Alix, Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: A comparative study, International Journal of Multiphase Flow 95 (2017) 235–256.

[167] R. Thirumalaisamy, A. P. S. Bhalla, A low mach enthalpy method to model non-isothermal gas–liquid–solid flows with melting and solidification, International Journal of Multiphase Flow 169 (2023) 104605.