LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TH-732004

On the Preconditioning of a Newton-Krylov Solver for a High-Order reconstructed Discontinuous Galerkin Discretization of All-Speed Compressible Flow with Phase Change for Application in Laser-Based Additive Manufacturing

B. Weston

May 25, 2017

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

On the Preconditioning of a Newton-Krylov Solver for a High-Order reconstructed Discontinuous Galerkin Discretization of All-Speed Compressible Flow with Phase Change for Application to Laser-Based Additive Manufacturing

By

Brian T. Weston

B.S. (University of California, Santa Cruz) 2012

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

MECHANICAL AND AEROSPACE ENGINEERING

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Jean-Pierre Delplanque

Robert Nourgaliev

Robert Guy

Committee in Charge

2017

Contents

Brian T. Weston
June 2017
Mechanical & Aerospace Engineering

On the Preconditioning of a Newton-Krylov Solver for a High-Order reconstructed Discontinuous Galerkin Discretization of All-Speed Compressible Flow with Phase Change for Application to Laser-Based Additive Manufacturing

Abstract

This dissertation focuses on the development of a fully-implicit, high-order compressible flow solver with phase change. The work is motivated by laser-induced phase change applications, particularly by the need to develop large-scale multi-physics simulations of the selective laser melting (SLM) process in metal additive manufacturing (3D printing). Simulations of the SLM process require precise tracking of multi-material solid-liquid-gas interfaces, due to laser-induced melting/solidification and evaporation/condensation of metal powder in an ambient gas. These rapid density variations and phase change processes tightly couple the governing equations, requiring a fully compressible framework to robustly capture the rapid density variations of the ambient gas and the melting/evaporation of the metal powder. For non-isothermal phase change, the velocity is gradually suppressed through the mushy region by a variable viscosity and Darcy source term model. The governing equations are discretized up to 4^{th}-order accuracy with our reconstructed Discontinuous Galerkin spatial discretization scheme and up to 5^{th}-order accuracy with L-stable fully implicit time discretization schemes (BDF2 and ESDIRK3-5). The resulting set of non-linear equations is solved using a robust Newton-Krylov method, with the Jacobian-free version of the GMRES solver for linear iterations. Due to the stiffnes associated with the acoustic waves and thermal and viscous/material strength effects, preconditioning the GMRES solver is essential. A robust and scalable approximate block factorization preconditioner was developed, which utilizes the velocity-pressure ($\mathbf{v}P$) and velocity-temperature ($\mathbf{v}T$) Schur complement systems. This multigrid block reduction preconditioning technique converges for high CFL/Fourier numbers and exhibits excellent parallel and algorithmic scalability on classic benchmark problems in fluid dynamics (lid-driven cavity flow and natural convection heat transfer) as well as for laser-induced phase change problems in 2D and 3D.

Acknowledgments

I would first like to thank my entire PhD dissertation committee: Dr. Robert Nouragliev, Professor Jean-Pierre "JP" Delplanque, and Professor Bob Guy for all of the helpful feedback on my dissertation. I've learned an incredible amount from all of your comments and individual discussions that we've had.

Robert was my LLNL technical supervisor and my primary technical resource who has spent countless days helping me during my time at LLNL. As the primary developer of the code I worked with, I'd go to his office multiple times a day and he'd always be willing to answer any questions I had. He's a brilliant researcher and he's been an incredibly supportive and encouraging mentor and role model. During my time at LLNL, Robert has given me a lot of strategic advice, such as how to choose "battles" in one's career and how writing a journal paper is similar to being a prosecutor presenting evidence to the judge. I'm looking forward to continuing my work with him during my Postdoc at LLNL.

I've known Bob since my first year at UC Davis. I enjoyed his teaching style so much that I decided to take every graduate class he offered. All of my knowledge of finite-difference methods for PDE's comes from his classes. Bob is also very approachable and has been a great resource for discussing numerical methods.

JP, my PhD dissertation advisor, has given me excellent strategic guidance as a mentor and helped forge our initial collaboration between LLNL and UC Davis. Typically students who go off to do their research at external labs outside the University rarely talk to their PhD advisor. JP, however, continued to stay involved in my work (with research updates every week), and he's been more available than just about any other professor I know. He's also very approachable and I've asked him for advice on just about everything: CV and interview questions, writing and presentation tips, how to manage professional relationships with collaborators, what being a professor is like, applying for grants, etc.

I would also like to thank Dr. Rose McCallen, the former ALE3D Project lead at LLNL, for hiring me into the ALE3D team and connecting me to my current mentor, Robert. During my time at LLNL, Rose has given me genuine and unbiased career advice, always wanting the best for me. Many of the students at LLNL or at UC Davis who know Rose, would describe Rose as a "motherly" mentor, which has been very refreshing. Rose has also shared a lot of insight on effective communication and helped me improve my research presentations tremendously. Without Rose I would not be at LLNL, so I am very grateful for meeting her.

I would like to thank other members of LLNL's ALE3D team: Dr. Andy Anderson, Dr. Saad Khairallah, Dr. Sam Schofield, Dr. Aaron Fisher, Dr. Patrick Greene, Dr. Paul Tsuji, Dr. Chad Noble, and Dr. Tim Dunn, who have attended most of my dry-runs and provided lots of insightful feedback. Saad has been supportive of me and I've enjoyed my many conversations with him. He has also given me lots of great career advice and I'm looking forward to working more with him on additive manufacturing. I also wanted to thank George Zagaris from the Ares team. George and I became friends right away, since we were both in the 'cooler' (Building 219) together. We've had many deep discussions about applied mathematics, high-performance computing, and the finite element method, as well as software engineering practices. I'm looking forward to working more with George in the future, since our backgrounds are complementary (he's the computer science guy and I'm the computational physics and math guy).

I would like to thank all of the graduate students in my research group at UC Davis, especially Carlos Ruvalcaba and Alex Wolfer. Carlos and Alex let me crash on their couch every Thursday evening before our Friday group meetings in Davis. I've had many motivating and stimulating conversations with Carlos about research, academic careers, outreach, mentoring, and world issues. Since Alex also had a similar research topic, we've had useful research discussions, and he also knows Python inside and out, which came in handy.

In the summer of 2014, I did a weekly lecture series on the finite-element method at LLNL. I'd like to thank all of the interns and friends who attended the lectures: Matthias Maitterth, Doru

Bercea, and Timmy Meyer. I learned quite a bit about computer architecture from Matthias and I remember the time in Davis when he taught me and Carlos about what a 5D torus topology was. I've also had many interesting discussions with Doru about PETSc, supercomputers, and strong scaling. Doru and I also had a "fun time" studying Gilbert Strang's textbook on Computational Science and Engineering :)

I would like to thank my family for all of the support over the years: my dad Gary, my mom Joanie, and my older brother Justin. My dad inspired me to pursue science at a young age by showing me the planets in his telescope. I also remember the days when he would come to my 3^{rd} and 4^{th} grade science classes as a guest instructor and show cool demonstrations with liquid nitrogen. I'll also never forget that he taught me basic powers of ten (the power rule in Calculus) back in Kindergarten. My mom has been an inspiring figure in my life and I am thankful for all of the freedom and trust she has given me over the years. I've enjoyed having weekly conversations with my brother, Justin. He encouraged me to pursue the research opportunity at LLNL back in 2014, and he's been very supportive, giving me excellent career and life advice over the years, which has turned out to be very fruitful. I'll never forget the time when he tutored me in my high school philosophy class. I was struggling with understanding a section in "Plato's Allegory of the Cave", and he taught me how to deconstruct and decode complex passages, line by line. That lesson of breaking down complicated logic step by step ultimately proved to be very valuable in my career when I needed to decode very sophisticated mathematical papers.

Last, but not least, I would like to thank all of my sponsors for the financial support: the Project lead of ALE3D, Chad Noble, the ASC AM lead, Bob Ferencz, and Rob Neely for support from the LEARN portfolio.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was partially funded by the Laboratory Directed Research and Development Program at LLNL under project tracking code 13-SI-002. The LLNL document review and release number is LLNL-TH-732004.

CHAPTER 1

Introduction

This work is motivated by the need to develop large-scale multi-physics simulations of 3D metal additive manufacturing processes, such as the selective laser melting (SLM) process. Simulating the SLM process requires solving the low-Mach compressible Navier-Stokes equations with phase change. This dissertation focuses on the development of a scalable and robust solver for numerical simulations of laser-based additive manufacturing processes.

1.1. Motivation: Additive Manufacturing

Additive manufacturing (AM) is the process of constructing a pre-designed solid object by adding material, as opposed to subtractive manufacturing in which material is removed to form the final shape. In 3D printing, a three-dimensional part is additively manufactured, layer by layer, using a computer aided design (CAD) model. 3D printing processes enable the production of components that would otherwise be impossible to produce from traditional manufacturing methods. In Figure 1.1 (right image), a complex manifold with a built in piping system was created with a metal 3D printer.

Powder Bed Fusion (PBF) methods are a type of 3D printing that use a laser or electron beam to sinter or melt material powder together. Common PBF processes include: Electron beam melting (EBM), Selective laser sintering (SLS), Direct metal laser sintering (DMLS), and Selective laser melting (SLM). For manufacturing metal components, SLM is a powder bed process that creates a part by using a high powered laser to selectively melt metallic powder together. As seen in Figure 1.2, the powdered metal is spread evenly on a substrate plate in the build chamber. An inert gas, typically argon or nitrogen, occupies the rest of the chamber in order to reduce oxidation. Before production, the powder bed is preheated in preparation for melting. During operation, the laser

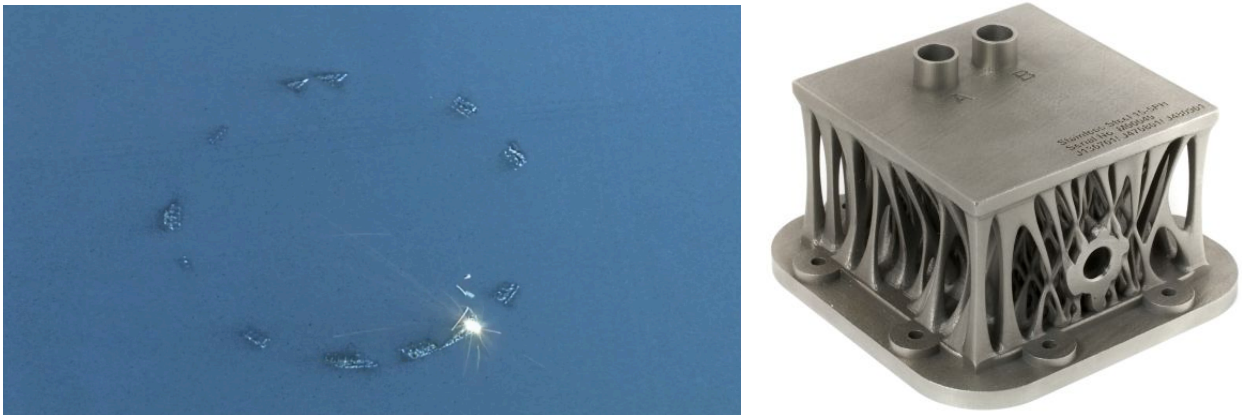


FIGURE 1.1. Left: Photo of the selective laser melting process [77]. Right: Complex manifold produced with a 3D SLM printer [1].

rapidly scans a 2D cross section of the part on a powder layer. Along the path of the laser, the powder rapidly melts and then solidifies. A piston then lowers the substrate and a new layer of powder is rolled on top, repeating this process until the part is finished.

Metal AM is poised to substantially change the design and production of metal parts in the aerospace, automotive, military, and medical industries [31, 38]. Comparable mechanical properties of additively manufactured metal parts to their cast or wrought counterparts, cannot be achieved, however, until several scientific and technical challenges are addressed. Currently, metal parts manufactured using powder-based methods have different micro-structural properties and performances than those produced with traditional manufacturing techniques [25]. Physics-based models, however, can elucidate the driving mechanisms in the SLM process. Coupled with Uncertainty Quantification (UQ) techniques, computational simulations of the AM process will be an indispensable tool for optimizing the design of metal parts [38].

1.2. Challenges

Developing physics-based computer simulations of the selective laser melting process for metal-additive manufacturing, is a challenging task, however. Multiple physical processes are involved

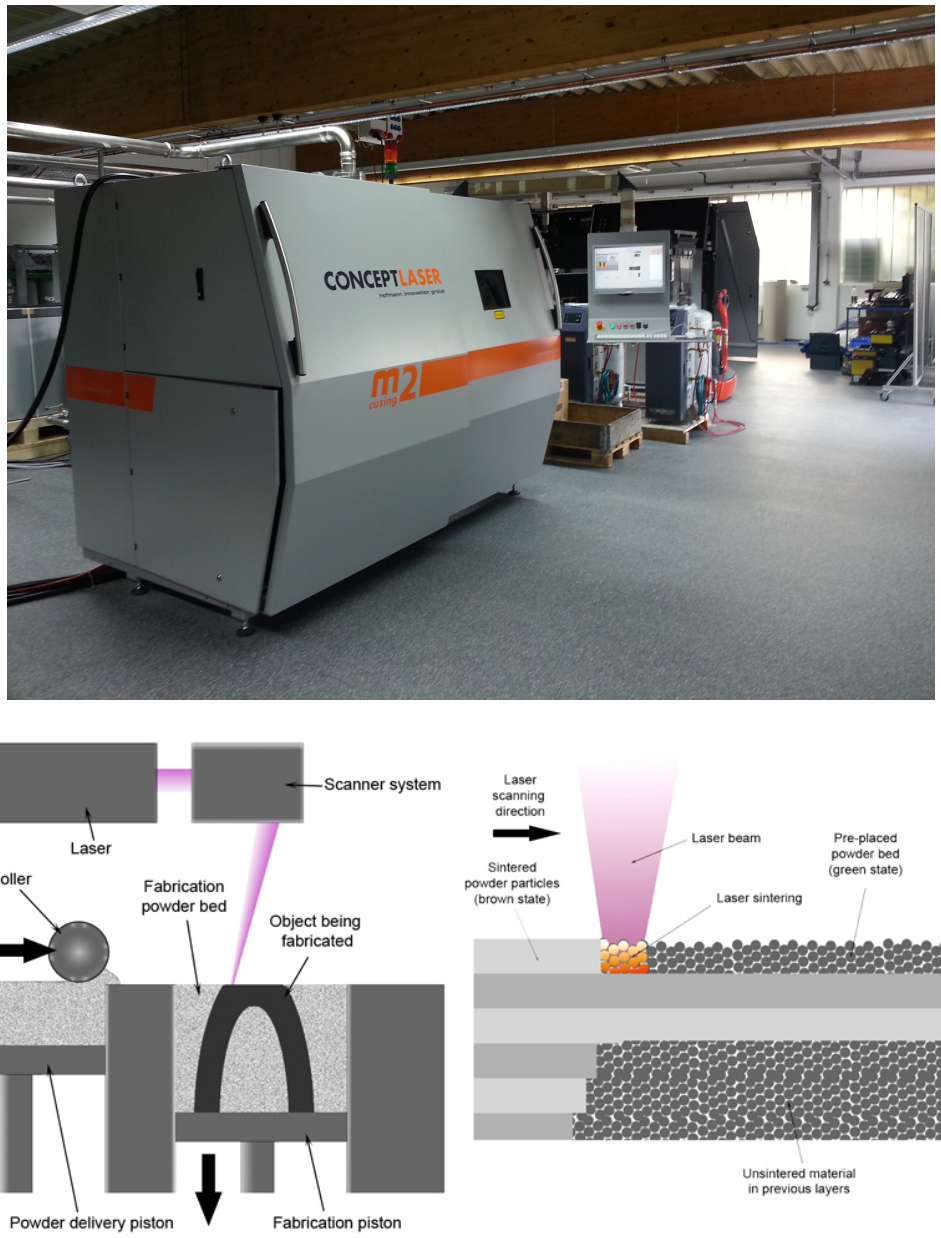


FIGURE 1.2. Top: Concept Laser M2 machine at LLNL [39]. Bottom: 2D cross-section of the build chamber [60].

including laser radiation and absorption, rapid phase change (melting/solidification and evaporation/condensation), heat transfer (conduction, convection, and radiation), fluid dynamics, and complex interfacial physics at multi-material interfaces, such as surface tension, Marangoni convection, and wetting [36, 45]. Furthermore, all of the physics occurs on a similar time-scale and

cannot be de-coupled from one another.

Besides the difficulties in modeling all of the physical processes relevant to SLM, there are several numerical modeling challenges. Simulations of the powder melting process require interface tracking of multiple materials and phases, which have large variations in the thermo-physical properties (density, thermal conductivity, viscosity, specific heat, internal energy, etc.) across sharp solid-liquid-gas interfaces. These rapid density variations and phase change processes (laser-induced melting/solidification and evaporation/condensation) tightly couple the governing equations, rendering incompressible flow solvers, such as the SIMPLE/projection-family of algorithms, ineffective for these applications. Therefore, an adequate model of the laser melt dynamics requires an all-speed, fully-compressible formulation, since the fluid dynamics in the melt pool are at low-Mach number. For non-isothermal phase change problems, the thermo-physical properties transition between the solid and liquid phases through the presence of a mushy region. The solid phase must enforce a no velocity condition and there are various models in the literature for suppressing the velocity in the solid phase, which introduce additional non-linearities in the governing equations [19, 26, 84]. Lastly, the rapid heating due to the laser source introduces large and localized temperature gradients, which further increases the non-linearity and coupling of the equations.

The low-Mach compressible flow regime constitutes a common regime in many engineering processes. Applications include combustion processes [33, 34, 42, 62, 80], high temperature gases in nuclear reactors [24, 47], and laser welding/melting processes [4, 37, 46, 69]. As outlined in [9, 47, 64], simulating low-Mach compressible flow is challenging because the governing equations change nature from hyperbolic to mixed hyperbolic-elliptic, as the flow transitions from compressible to incompressible flow. As a result, two categories of solvers have been developed for simulating flows in this transitional regime: density-based methods from high-speed compressible solvers and pressure-based methods from incompressible solvers. For small density variations, incompressible flow solvers utilizing the Boussinesq approximation are widely used in the literature [19, 26, 84]. For large temperature gradients, which subsequently result in large density variations, the Boussinesq approximation, however, is no longer valid [9, 48, 59] and a compressible formulation is required.

Compressible flow solvers suffer from challenges of their own in the low-Mach regime. Due to the disparity of the acoustic and velocity time scales, the eigenvalues of the system become highly ill-conditioned and convergence stalls [15, 50, 64, 83]. Furthermore, the numerical solutions are known to become inaccurate in the limit of low-Mach numbers due to lack of numerical dissipation [30, 50].

Since we are solving the compressible flow equations with melting and solidification, there are several physical time scales associated with the governing equations: acoustic waves, viscous diffusion, thermal diffusion, material advection, and melting front dynamics. In the low-Mach regime, there is a huge discrepancy between the acoustic and advection timescales leading to a numerically stiff system. Due to the stiffness and necessity of stepping over acoustic and advection timescales[1], numerical algorithms using explicit time integrators or operator-splitting algorithms, such as projection algorithms or SIMPLE, have severe time step restrictions due to stability requirements, making large-scale simulations prohibitively expensive and requiring weeks to months of HPC time on even the largest supercomputers [36, 46]. With a fully-implicit time discretization, large time steps can be taken, chosen based on the adequate resolution of the dynamic time scales of the problem, rather than by numerical stability restrictions dictated by the physical time-scales. Stepping over these time-scales, however, results in numerical systems that develop strong hyperbolic and parabolic stiffnesses [13]. To effectively step over these time-scales, we employ L-stable time integrators, like BDF_2 or the p^{th}-order *Explicit, Singly-Diagonal Implicit Runge-Kutta* schemes, ESDIRK_p, [10, 12, 68].

1.3. Background on Fully-Implicit Framework

The Discontinuous Galerkin (DG) methods, built upon fully-implicit Newton-Krylov-based solvers, are a promising class of solution algorithms for high-fidelity simulations of the SLM process on large-scale supercomputers. Popular within the high-speed aerodynamics community, a DG-based framework retains attractive features from both the finite element and finite volume methods and has excellent conservation and convergence properties. The DG framework allows

[1]Dynamic time scales of AM processes with melting/solidification are rather slow compared to CFL/Fourier number based time scales. Cost-effective simulations necessitate stepping over advection timescales.

for the development of higher-order methods that are well suited for handling hybrid hyperbolic/elliptic/parabolic equations within the finite element/volume framework for complex geometries on unstructured meshes [68]. In addition, the methods are locally conservative allowing for spatially compact stencils and non-conforming elements. As a result, DG methods are highly parallelizable due to minimal communication requirements between elements. Furthermore, DG methods have a high arithmetic intensity, allowing the methods to readily exploit the massive parallelism from future computer architectures, such as the Graphics Processing Units (GPUs).

In order to solve the system of non-linear equations at each time step, a robust solution algorithm is needed. The Newton-Krylov framework, which uses an outer Newton method combined with an inner Krylov iterative method, provides such an approach [40]. The Newton-Krylov framework has been successfully applied to the non-equilibrium radiation diffusion equations [63], the shallow water equations [65], the incompressible Navier-Stokes equations [43, 70, 72], low-Mach combustion [42], and more recently to the compressible Navier-Stokes equations [14, 68, 73]. The framework has also been applied to Stefan problems for both pure-materials and alloys [41], as well for solidifying flow applications involving 2D melt convection for incompressible flow [26, 44].

We use a globalized line search strategy, where the descent direction and step size is computed with a Jacobian-Free Newton-Krylov (JFNK) framework. GMRES is chosen as the Krylov (linear) solver because it is robust and guarantees a monotonically decreasing residual. Since the linear system is the result of a discretization of the compressible Navier-Stokes equations with phase change, it has a 1) mixed hyperbolic and parabolic nature, 2) non-symmetry due to upwinding in the approximate Riemann solver, 3) non-diagonally dominant structure when large time-steps are chosen. Furthermore, the governing equations are tightly-coupled within the stencil and the resulting (global) linear system of discrete equations is highly ill-conditioned, with condition numbers exceeding a million [68]. Thus, preconditioning the GMRES-Newton solver is required for convergence.

There are two primary design requirements for the preconditioner. First, the preconditioner must be able to converge the solution of highly ill-conditioned linear systems corresponding to large CFL and Fourier numbers. Second, the preconditioning strategy must be scalable and computationally efficient in run time and memory storage so that large-scale multi-physics simulations are cost-effective. Classic iterative methods such as Jacobi, Gauss-Seidel, Sucessive-Over-Relaxation (SOR), and Algebraic Multigrid (AMG) are effective preconditioners when the matrix is diagonally dominant [74]. Due to the lack of diagonal dominance in our systems, however, these techniques are not guaranteed to converge (theoretically) and fail to converge in practice. On the other extreme, the most robust preconditioner is an LU factorization of the original Jacobian matrix. Although direct solvers are effective for small problems, they are not well-suited for massive computations as their computational cost does not scale linearly with the size of the computational problem [74].

Domain decomposition methods, such as additive Schwarz variants, are natural for unstructured meshes [82] and take a divide-and-conquer approach to parallelism [40, 74]. Additive Schwarz techniques have been effective preconditioners for Newton-Krylov solvers in compressible flow [79], reactive flow problems [76], and low-Mach compressible combustion [42]. A drawback of one-level additive Schwarz methods is the locality assumption, since neighboring degrees of freedom to an element are strongly coupled while long-range interactions are ignored. As a result, the performance of additive Schwarz preconditioners generally degrades for elliptic problems as the number of processor domains increase, due to the lack of global coupling [78, 81]. To address the lack of global coupling in domain decomposition methods, a field-block approach segregates all of the degrees of freedom of a particular field into separate blocks. These reduced scalar block (pressure, temperature, or velocity) systems are more amenable for iterative methods to approximate the action of their inverse. Variations of this physics-block preconditioning approach have been successfully applied to the non-equilibrium radiation diffusion equations [63], the shallow water equations [65], MHD [13, 18], solidifying flow applications [44], the incompressible Navier-Stokes equations [26, 72], and more recently to the compressible Navier-Stokes equations [68, 70, 73].

This work explores approximate block decomposition strategies on the segregated physics blocks. Approximate physics-block factorizations have been explored as a preconditioner for the incompressible Navier-Stokes equations and MHD using various approximations of the Schur complement matrix [17, 18, 22, 23]. This work differs as we extend these techniques to the all-speed compressible Navier-Stokes equations with phase change using a high-order reconstructed Discontinuous Galerkin discretization in the limit of large CFL and Fourier numbers. These approximate block factorizations are multigrid block reductions and algebraic multigrid (AMG) methods can be effectively applied on these reduced systems, which are linearly scalable algorithms, $O(n)$.

The framework is implemented and tested within LLNL's ALE3D code [3, 29, 61]. ALE3D is a multi-physics numerical simulation tool, focusing on modeling hydrodynamics and structural mechanics in all-speed multi-material applications. Additional ALE3D features include heat transfer, chemical kinetics and species transport, incompressible flow, a wide range of material models, chemistry models, multi-phase flow, and magneto-hydrodynamics for long- (implicit) and short- (explicit) time-scale applications.

1.4. Overview of the Dissertation

The rest of this dissertation is organized as follows. The physical and mathematical models required for simulating laser-induced melt convection problems are discussed in Chapter 2. In Chapter 3, a brief introduction is given on the reconstructed Discontinuous Galerkin (rDG) spatial discretization scheme and its advantages for the underlying linear algebra are highlighted. The fully-implicit time discretization schemes are outlined along with the Jacobian-Free Newton-Krylov (JFNK) solver. The preconditioning (Jacobian) matrices for the rDG schemes are also explained. In Chapter 4, verification of the different rDG discretization schemes is demonstrated with the Method of Manufactured Solutions (MMS) to verify that the schemes converge to the design order of accuracy for the compressible Navier-Stokes equations. To illustrate the benefit of a 4^{th}-order over a 2^{nd}-order rDG spatial discretization scheme, a qualitative comparison of the flow features is

conducted for a laser-induced phase change problem. Various combinations of the velocity suppression models for a solid-liquid phase change problem are analyzed in Chapter 5. A variable viscosity method combined with a drag force model is found to be the most robust and computationally efficient model for suppressing the velocity in the solid and mushy phases. In Chapter 6, sensitivity studies are performed for the non-dimensional numbers (Rayleigh, Prandtl, and Stefan Numbers) and different melting and solidification configurations are tested, demonstrating this algorithm's ability to robustly handle both melting and solidification problems. In Chapter 7, 2D and 3D moving laser-induced melt convection problems are tested using the developed velocity suppression and laser beam models. Chapter 8 compares different preconditioning strategies and finds that the most robust and efficient preconditioner is an approximate physics-block factorization technique (multigrid block reduction), utilizing the velocity-pressure ($\mathbf{v}P$) and velocity-temperature ($\mathbf{v}T$) Schur complement systems. Lastly, Chapter 9 concludes the dissertation and provides an outline of future directions. Future work involves moving to larger block systems due to higher-order (greater than 2^{nd}-order) schemes and binary alloy systems, incorporating additional physics, such as surface tension, Marangoni convection, radiation and evaporation, and coupling to solid-thermal mechanics solvers. The larger block (4×4, 5×5, $N \times N$) systems will need to be preconditioned in a scalable and efficient manner. I plan to extend my preconditioning techniques to these larger block systems using an $N \times N$ multigrid block reduction technique.

The work was primarily conducted at the Lawrence Livermore National Laboratory in Livermore, California and occasionally at the University of California, Davis. This dissertation summarizes the work performed by the author during the time period of June, 2014 to June, 2017. It resulted in 11 conference presentations, 2 peer-reviewed papers, and 2 additional papers currently in preparation.

CHAPTER 2

Mathematical Model

2.1. Governing Equations

2.1.1. Conservation Laws. In this work, we consider the time-dependent compressible Navier-Stokes equations with solid-liquid phase change. The governing conservation equations in vector form are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2.1}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \times \mathbf{v}^{\mathbf{T}}) = -\nabla P + \nabla \cdot \bar{\bar{\sigma}} + \rho g - f_D \tag{2.2}$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{v} e) = \nabla \cdot (k \nabla T) - P \nabla \cdot \mathbf{v} + \nabla \cdot (\mathbf{v} \cdot \bar{\bar{\sigma}}) + q_v, \tag{2.3}$$

where $\mathbf{v} = (v_x, v_y, v_z)$ is the material velocity vector in Cartesian coordinates, P is the pressure, ρ is the density, g is gravity, $\bar{\bar{\sigma}}$ is the viscous stress tensor defined in 2.1.2, $e = u + \frac{\mathbf{v}^2}{2}$ is the specific total energy, u is the specific internal energy[1], k is the thermal conductivity, and T is the temperature. The drag force due to solidification is represented by $f_D = -K\mathbf{v}$, where K increases from zero to a very large number as the local solid fraction varies from zero to one, suppressing the motion in the solid phase. Laser heating is modeled as a volumetric source term, q_v, in the energy equation, which is defined in Section 2.3. In this work, we neglect energy dissipation due to viscous stresses and pressure forces, since we are studying compressible flows in the low-Mach limit. Radiation, evaporation, and free-surface dynamics are also neglected and will be considered in future work.

The same governing equations can also be written in following flux vector form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{F} - \mathbf{D}) = \mathbf{S}, \tag{2.4}$$

[1]We choose to discretize the energy equation in the internal energy form, as opposed to the enthalpy or total energy form. Mathematically, all energy formulations are equivalent, but when considering our rDG numerical discretization, we find that the internal energy formulation is easier to implement.

where $\mathbf{U}$ is the solution vector of conservative variables, defined as

$$\mathbf{U} = \left[\ \rho, \rho\mathbf{v}, \rho e \ \right]^{\mathrm{T}}, \tag{2.5}$$

while $\mathbf{F}$, $\mathbf{D}$, and $\mathbf{S}$ are the vectors of hyperbolic fluxes, diffusion fluxes, and sources, respectively, defined as

$$\mathbf{F} = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \times \mathbf{v}^T + pI \\ \rho\mathbf{v}e + p\mathbf{v} \end{bmatrix}, \ \mathbf{D} = \begin{bmatrix} 0 \\ \bar{\bar{\sigma}} \\ -k\nabla T + \mathbf{v} \cdot \bar{\bar{\sigma}} \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} 0 \\ \mathbf{f_b} \\ q_v \end{bmatrix}. \tag{2.6}$$

We also introduce a vector of "primitive" variables, $\mathbf{W}$, which is generally different from $\mathbf{U}$, and chosen based on the "better system conditioning" considerations as defined in Section 3.4.

2.1.2. Constitutive Equations. In this work we consider Newtonian fluids. As defined in [5, 8, 86], the viscous stress tensor is

$$\bar{\bar{\sigma}} = 2\mu\bar{\bar{\epsilon}} + \lambda(\nabla \cdot \mathbf{v})I, \tag{2.7}$$

where μ is the dynamic viscosity and λ is the second viscosity coefficient. Following Stokes hypothesis, the second viscosity is taken as $\lambda = -\frac{2}{3}\mu$ [5]. The strain rate tensor, $\bar{\bar{\epsilon}}$, is defined as

$$\bar{\bar{\epsilon}} = \frac{1}{2}\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right). \tag{2.8}$$

The final constitutive relationship is due to the equation of state (EOS). This dissertation uses the simple *2-parameter* (ρ_0, c) EOS to directly control the sound speed

$$P(\rho) = \rho_0 c^2 \left(\frac{\rho}{\rho_0} - 1\right), \tag{2.9}$$

since we are interested in the low Mach regime, $Ma = \frac{|\mathbf{v}|}{c} < 10^{-2}$, where ρ_0 and c are the given reference density and sound speed, respectively. With direct control of the sound speed in our EOS, we can artificially reduce the sound speed and thus Mach numbers to tractable values $(10^{-5} \leq Ma \leq 10^{-2})$, which is favorable for the underlying linear algebra. We numerically verified that the numerical solution is independent of the Mach number in this range and this approach is

similar to artificial compressibility methods used in incompressible flow solvers [16][2].

2.2. Modeling of Melting and Solidification

2.2.1. Phase Transition. Phase change is modeled with an energy-based (homogenous thermal equilibrium) approach. Phase transitions are implicitly tracked with our thermal model $u(T)$ as follows.

I. **Solid**. ($T < T_S$, $u < u_S$), where T_S and u_S are the solid melt temperature and specific internal energy, respectively.

II. **Liquid**. ($T > T_L$, $u > u_L$), where T_L and u_L are the liquid melt temperature and specific internal energy, respectively.

III. **Two-Phase**. ($T_S < T < T_L$, $u_S < u < u_L$).

The three material zones are shown in Figure 2.1. We introduce a transitional two-phase region, between the solid and liquid phases to avoid a non-singular mapping between u and T. The thickness of the two-phase mushy region is defined by $\epsilon = T_L - T_S$. The jump in internal energy between the solid and liquid phases is stored in the latent heat term, defined as

$$u_f = u_L - u_S. \tag{2.10}$$

To suppress the velocity in the mushy and solid phases, we use a velocity suppression model as outlined in Section 2.2.2. More details are described in our paper, [68], explaining how the thermodynamic properties transition between the phases, such as viscosity, thermal conductivity, internal energy, and specific heat.

[2]In this work, we do not intend to account for realistic sound speeds in metal, which would drive the Mach number to restrictively small values, since the sound speeds are on the order of km/s. In future work, we will implement the AUSM$^+$-*up* Riemann solver, which correctly mimics the pressure fluctuations of an incompressible flow solver in the asymptoptic limit of small Mach number [49, 50, 51].

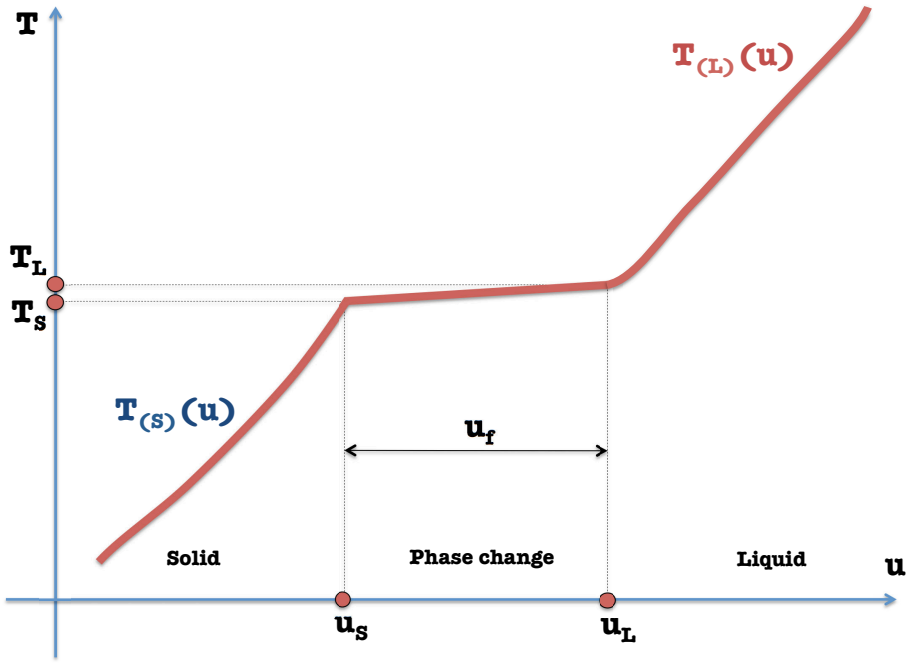


FIGURE 2.1. Thermal model, $u(T)$, for equilibrium phase change.

2.2.2. Velocity Suppression Models. In this Section, we briefly review two of the well known velocity suppression models used in phase change problems to extinguish the velocity in the solid and mushy phases. Based on these two models, we develop a combined velocity suppression model. A more extensive review of velocity suppression techniques can be found in [58].

2.2.2.1. *Darcy Source Term Model.* To suppress the velocity in the solid and mushy phases, many authors [26, 32, 84] use a drag force/momentum sink term, inspired from Darcy's Law for porous media flow

$$\mathbf{v} = -\frac{\kappa}{\mu}\nabla P, \tag{2.11}$$

where κ is the permeability coefficient of the medium. In this work, we approximate the permeability to be a function of the porosity, where the porosity is defined as

$$\lambda = 1 - \phi_s = x, \tag{2.12}$$

where x is the thermodynamic quality and ϕ_s is the local solid faction. ϕ_s varies linearly as a function of temperature from 0 (in a liquid state) to 1 (in a solid state). As the local solid fraction approaches 1, the velocities in the material tend to 0. To numerically mimic the Darcy's law, a source term in the momentum equation can be introduced

$$f_{\mathrm{D}} = -K\mathbf{v}, \tag{2.13}$$

where K is a number that increases from zero to a very large number, as the local solid fraction varies from 0 to 1, effectively suppressing the motion in the solid phase. Inspired by the Carman-Kozeny equation in [11], K is typically modeled as

$$K = -C\frac{(1-\lambda)^2}{\lambda^3 + \epsilon}, \tag{2.14}$$

where C is some constant that depends on the permeability of the porous medium and ϵ is typically set to a small value $\sim 10^{-3}$, to avoid division by zero [84]. For simplicity, the drag force model used in this work has a linear dependence on the porosity

$$K = C(1-\lambda), \tag{2.15}$$

which is also used in [26, 28]. More complex permeability models based on the Darcy's law have been developed in [21], which we will consider in future work for solidification and re-melting problems.

2.2.2.2. *Variable Viscosity Model.* Another common velocity suppression model is known as the enhanced viscosity method or variable viscosity method [19, 20, 58]. In this model, material strength is a linear function of the strain tensor where the dynamic viscosity is a smooth function of temperature between the solid and liquid phases

$$\mu(T) = \begin{cases} \mu_L & \text{if } T \geq T_L \\ \mu_L f_\mu(T) & \text{if } T_S < T < T_L \\ \mu_S & \text{if } T \leq T_S \end{cases} . \tag{2.16}$$

$f_\mu(T)$ is the viscosity factor, a function that smoothly varies from 1 to a large number, $\frac{\mu_s}{\mu_l}$. As seen in Figure 2.2, this model progressively increases the viscosity of the material from the mushy region to the solid region, inhibiting material deformation.

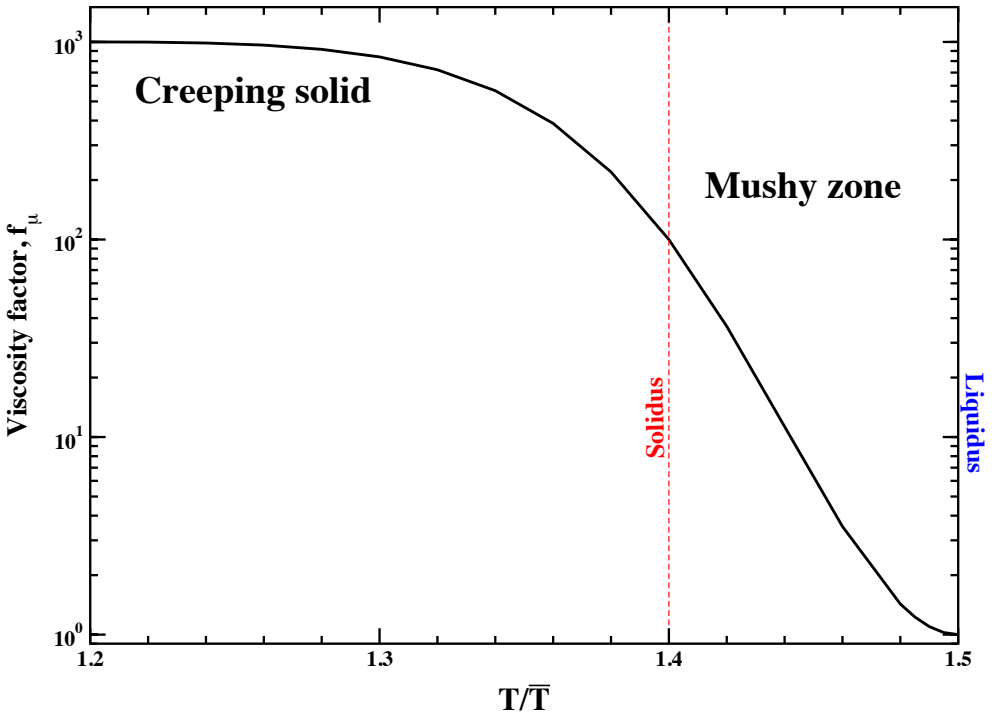


FIGURE 2.2. An example viscosity factor, as a function of temperature.

2.2.2.3. *Combined Velocity Suppression Model.* In this work, we use a combination of the Darcy source term and variable viscosity models to extinguish the velocity in the mushy and solid phases. Recall that in the Darcy model we introduce a source term in the momentum equation, $f_D = -C(1-\lambda)\mathbf{v}$, which introduces an additional model parameter, the drag coefficient, C. In the viscosity model, we introduce a variable coefficient parabolic operator in the momentum equation

$\nabla\mu(T)\nabla\mathbf{v}$, which also has one model parameter, the solid-liquid viscosity ratio, $\frac{\mu_s}{\mu_l}$. The combined velocity suppression model is analyzed in Section 5.2.

2.2.2.4. *Non-Linearity Associated with Velocity Suppression Models.* Both of the velocity suppression models introduce non-linearities to the governing equations. The Darcy strength model enforces a source-term coupling between the momentum and energy equations. The variable viscosity model, however, introduces a very strong coupling between the momentum and energy equations through the parabolic (viscous) operator, which presents challenges for solvability of the underlying systems. More details on the non-linearity of the velocity suppression models are given in Section 5.2[3].

2.3. Laser Beam Model

To simulate laser-induced melt convection, we implement a laser source heating term in our non-dimensional energy equation. The effects of different laser beam modes on melt pools was studied in [32]. In this work, we only consider source heating terms with a cylindrical, Gaussian profile. We define $\mathbf{r} = (r, z)$, which is a function of the radius, r, in 2D and a function of both the radius and depth, z, in 3D.

In 2D, the source heating term is modeled with a Gaussian profile. Relating the power of the laser, Q, to the max volumetric flux, q_v^{max},

$$Q = q_v^{max} \int_0^{2\pi} \int_0^{r_0} e^{-ar^2} r \, dr \, d\theta, \tag{2.17}$$

where $a \equiv \frac{1}{2r_0^2}$, r_0 is the radius of the laser's spot size, $r^2 = x^2 + y^2$. After integration, we solve for q_v^{max}

$$q_v^{max} = \frac{Qa}{\Delta z \pi (1 - e^{ar_0^2})}, \tag{2.18}$$

[3]Future work will seek to replace the velocity suppression model in the solid phase with realistic material strength. In alloy systems, a velocity suppression models will, however, still be used to model the dynamics in the mushy region.

where we set $\Delta z \equiv 10 r_0$. The flux can now be evaluated anywhere in the domain

$$q_v = q_v^{\max} f(\mathbf{r}) = q_v^{max} e^{-ar^2}, \tag{2.19}$$

where in 2D, $f(\mathbf{r}) = e^{-ar^2}$.

In 3D, we model the vertical (exponential) attenuation of the laser with a quadratic function in the z dimension. The relation between power and max volumetric flux becomes

$$Q = q_v^{max} \int_{z_0}^{h} \int_0^{2\pi} \int_0^{r_0} e^{-ar^2} \frac{(z - z_0)^2}{(h - z_0)^2} r \, dr \, d\theta \, dz, \tag{2.20}$$

where h and z_0 are the maximum and minimum heights of energy deposition (user defined), respectively. Solving for q_v^{max}

$$q_v^{max} = \frac{3Qa}{(h - z_0)\pi(1 - e^{ar_0^2})}. \tag{2.21}$$

Again, the volumetric flux can be evaluated anywhere in the domain

$$q_v = q_v^{\max} f(\mathbf{r}) = q_v^{max} e^{-ar^2} \frac{(z - z_0)^2}{(h - z_0)^2}, \tag{2.22}$$

where in 3D, $f(\mathbf{r}) = e^{-ar^2} \frac{(z - z_0)^2}{(h - z_0)^2}$.

2.4. Non-Dimensionalization

To non-dimensionalize the system of Eq.'s (2.1-2.3), we define four scaling parameters: length, velocity, temperature, and density:

$$\bar{\rho}, \bar{L}, \bar{\mathbf{v}}, \overline{\Delta T}. \tag{2.23}$$

For natural convection flows, the velocity scale is defined from the buoyancy term in the momentum equation, known as Grashof scaling

$$\bar{\mathbf{v}} = \sqrt{\bar{g}\bar{\beta}\bar{L}\overline{\Delta T}}, \tag{2.24}$$

where $\bar{\beta}$ is the volumetric thermal expansion coefficient. Given the base scaling parameters, $\bar{\rho}, \bar{L}, \bar{\mathbf{v}}, \overline{\Delta T}$ (Appendix A.3), we can define the remaining scaling parameters

$$\bar{t} = \frac{\bar{L}}{\bar{\mathbf{v}}} \qquad \bar{P} = \bar{\rho}\bar{\mathbf{v}}^2 \qquad \bar{\mu} = \bar{\rho}\bar{\mathbf{v}}\bar{L} \tag{2.25}$$

$$\bar{k} = \bar{\mu}\bar{C}_v \qquad \overline{\Delta T} = T_H - T_C \qquad \bar{E} = \bar{C}_v\overline{\Delta T} \tag{2.26}$$

$$\bar{q_v} = \frac{\bar{\rho}\bar{E}}{\bar{t}} \qquad \bar{K} = \frac{\bar{\rho}}{\bar{t}}. \tag{2.27}$$

The dimensionless variables that appear in the governing equations are defined as

$$\hat{T} = \frac{T - T_0}{\overline{\Delta T}} \qquad \hat{t} = \frac{t}{\bar{t}} \qquad \hat{\mathbf{v}} = \frac{\mathbf{v}}{\bar{\mathbf{v}}} \tag{2.28}$$

$$\hat{\rho} = \frac{\rho}{\bar{\rho}} \qquad \hat{k} = \frac{k}{\bar{k}} = \frac{1}{PrRe} \qquad \hat{\mu} = \frac{\mu}{\bar{\mu}} = \frac{1}{Re} \tag{2.29}$$

$$\hat{e} = \frac{e}{\bar{e}} \qquad \hat{P} = \frac{P}{\bar{P}} \qquad \hat{\mathbf{r}} = \frac{\mathbf{r}}{\bar{L}} \tag{2.30}$$

$$\hat{C_v} = \frac{C_v}{\bar{C_v}} \qquad \hat{q_v} = \frac{q_v}{\bar{q_v}} = \frac{q_v^{\max} f(\hat{\mathbf{r}})}{\bar{q_v}} \qquad \hat{K} = \frac{K}{\bar{K}} = \frac{K\bar{t}}{\bar{\rho}}. \tag{2.31}$$

The latent heat of fusion associated with solid-liquid phase change, is related to the Stefan number

$$Ste = \frac{\overline{C_v\Delta T}}{\overline{u_f}}, \tag{2.32}$$

which is defined as the ratio of sensible heat to latent heat. Appearing in both the momentum and energy equations are the Grashof and Prandtl numbers

$$Gr = \frac{\bar{g}\bar{\beta}\overline{\Delta T}\bar{L}^3}{\bar{\nu}^2} = \frac{\bar{\mathbf{v}}^2\bar{L}^2}{\bar{\nu}^2} = Re^2 \tag{2.33}$$

$$\mathrm{Pr} = \frac{\bar{\nu}}{\bar{\alpha}}. \tag{2.34}$$

Defining the external Rayleigh number for buoyancy driven flows

$$\mathrm{Ra}_e = \frac{\bar{g}\bar{\beta}\overline{\Delta T}\bar{L}^3}{\bar{\nu}\bar{\alpha}} = GrPr = Re^2Pr \tag{2.35}$$

and defining the internal Rayleigh number for laser-source heating

$$Ra_I = \frac{\bar{g}\bar{\beta}q_v^{\max}\bar{L}^4}{\bar{\nu}\bar{\alpha}\bar{k}}. \tag{2.36}$$

We note that the Mach number is defined as

$$Ma = \frac{|\bar{\mathbf{v}}|}{c}, \tag{2.37}$$

which arises from the equation of state. With these scaling parameters, the non-dimensional compressible Navier-Stokes equations are

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\mathbf{v}}) = 0 \tag{2.38}$$

$$\frac{\partial(\hat{\rho}\hat{\mathbf{v}})}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\mathbf{v}} \times \hat{\mathbf{v}}^T) = -\hat{\nabla}\hat{P} + \frac{1}{Re}\hat{\nabla} \cdot (\hat{\mu}\hat{\nabla}\hat{\mathbf{v}}) + \frac{Ra_e}{PrRe^2}\hat{\rho}\hat{g}\hat{\beta}\hat{T} - \hat{K}\hat{\mathbf{v}} \tag{2.39}$$

$$\frac{\partial(\hat{\rho}\hat{e})}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\mathbf{v}}\hat{e}) = \frac{1}{RePr}\hat{\nabla} \cdot (\hat{k}\hat{\nabla}\hat{T}) + \frac{Ra_I}{Ra_e PrRe} f(\hat{\mathbf{r}}), \tag{2.40}$$

where we use an approximate buoyancy term in the momentum equation. Using $Re = \sqrt{\frac{Ra}{Pr}}$,

$$\frac{\partial \hat{\rho}}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\mathbf{v}}) = 0 \tag{2.41}$$

$$\frac{\partial(\hat{\rho}\hat{\mathbf{v}})}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\mathbf{v}} \times \hat{\mathbf{v}}^T) = -\hat{\nabla}\hat{P} + \frac{\sqrt{Pr}}{\sqrt{Ra}}\hat{\nabla} \cdot (\hat{\mu}\hat{\nabla}\hat{\mathbf{v}}) + \hat{\rho}\hat{g}\hat{\beta}\hat{T} - \hat{K}\hat{\mathbf{v}} \tag{2.42}$$

$$\frac{\partial(\hat{\rho}\hat{e})}{\partial \hat{t}} + \hat{\nabla} \cdot (\hat{\rho}\hat{\mathbf{v}}\hat{e}) = \frac{1}{\sqrt{RaPr}}\hat{\nabla} \cdot (\hat{k}\hat{\nabla}\hat{T}) + \frac{Ra_I}{\sqrt{Pr}Ra_e^{\frac{3}{2}}} f(\hat{\mathbf{r}}). \tag{2.43}$$

CHAPTER 3

Numerical Methods

To solve the non-dimensionalized equations in Section 2.4, we use the fully-implicit, Newton-Krylov framework with the rDG spatial discretization scheme, developed in [68]. In this Chapter, we describe our numerical approach and highlight its major advantages for the solvability of the underlying linear algebra.

Recently, the Discontinuous Galerkin (DG) method has become increasingly popular in computational fluid dynamics, owing to its flexibility of handling complex geometry, its compact stencil for arbitrarily higher-order solutions, and its amenability to parallelization and *hp*-adaptation. In contrast to more traditional finite volume (FV) methods in CFD, high-order accuracy is achieved by simply adding additional degrees of freedom (DoFs) per element, per variable. As a result, the DG(P_p) of any order p has the same stencil (i.e., only face-neighbors are involved in discretization), which is a very attractive feature in terms of parallelization and code design. On the other hand, the size of the solution vector grows significantly, as more DoFs must be solved for. Such an increase in the size of the solution vector is unfavorable in the context of implicit solvers, imposing significant memory requirements (for storage of the matrices) and adversely affecting solution scalability, as a majority of linear solvers do not scale linearly [74]. In order to reduce high costs associated with DG, the reconstructed DG (rDG) methods have been developed [56, 57, 89].

We capitalize on the recent work of extending the rDG discretization to a Newton-Krylov framework for solving highly ill-conditioned multi-physics problems [68]. We use the orthogonal modal Legendre-based tensor-product basis functions. These basis functions are hierarchical, which naturally facilitates p-refinement, and can be easy implemented on hybrid meshes with AMR. The 0^{th}-order degrees of freedom are cell-averaged quantities, while the higher order degrees of freedom correspond to derivatives of the cell-averaged quantities (slopes, quadratics, cubics, etc). In the

rDG P_NP_M schemes, we solve for a P_N scheme of polynomial order N and reconstruct to a P_M scheme of polynomial order M, which is a $(M+1)$ order-accurate method in space.

In this work, we study the performance of the P_0P_1, P_1P_3, and P_2P_3 (2^{nd}-order and 4^{th}-order) accurate discretization schemes. The benefits of reconstructed DG vs. unreconstructed DG with regards to solution vector size can be seen in Table 3.1. The DG P_3 scheme in 2D has a total of 10 DoFs per equation per element: 1 cell-averaged value + 2 slopes + 3 quadratics + 4 cubics, while in 3D it has a total of 20 DoFs per equation per element: 1 cell-averaged value + 3 slopes + 6 quadratics + 10 cubics. The rDG P_2P_3 scheme operates on the same polynomial space and has the same total number of DoFs and order of accuracy as the DG P_3 scheme, but with a significantly smaller solution vector size. This is because only the cell-averaged value, slopes, and curvatures are **solved for**, while the cubic DoFs are **reconstructed**. It can be easily observed that the benefit of reconstructed DG vs. unreconstructed DG increases as the dimension increases. Thus the required number of degrees of freedom to be **solved for** per equation per element for each discretization scheme is listed in Table 3.2. Since the base 0^{th}-order degrees of freedom are cell-averaged quantities, this DG method can be viewed as a generalized extension of the finite-volume (FV) algorithm to high-order (greater than 2^{nd}-order) on unstructured hybrid meshes, without the need to extend the stencil. The stencil for our rDG scheme includes neighbors of neighbors, a total 12 neighboring elements in 2D, and 24 neighboring elements in 3D. This stencil is in fact identical to 2^{nd}-order finite-volume methods, common in most commercial CFD solvers.

Dim.	**DG** P_3	**rDG** P_2P_3
1D	4 DoFs	3 DoFs
2D	10 DoFs	6 DoFs
3D	20 DoFs	10 DoFs

TABLE 3.1. Solution vector size per equation per element for unreconstructed vs. reconstructed 4^{th}-order DG.

The beauty of rDG methods is that they provide a unified formulation for both FV and DG, and contain both classical FV and standard DG as two special cases of the rDG formulation. In [68], we developed the rDG method which is specifically designed for solving stiff multiphysics problems using fully-implicit formulation. Two major innovations are *1)* the combination of the

Scheme	2D	3D
rDG P_0P_1	1 DoF	1 DoF
rDG P_1P_3	3 DoF	4 DoF
rDG P_2P_3	6 DoF	10 DoF

TABLE 3.2. Solution vector size per equation per element for rDG schemes considered here.

in-cell and inter-cell reconstructions using orthogonal basis/test functions on unstructured meshes, and *2)* an ability to solve for primitive variables, chosen on the requirement of solvability (better conditioning) of the underlying physics. Both developments are aimed at better conditioning of linear steps during the Newton-based non-linear iterative procedure in fully-implicit solver. In the contrast to early rDG efforts [52, 53, 54, 55, 56, 57, 87, 88, 89], which used conservation variables as a solution vector, the new method is designed for **all-speed flow** capabilities, with **phase change (melting/solidification)**. In these cases, the set of conservation variables (i.e., mass, momentum and total energy) is poorly conditioned and restrictive in terms of feasible flow regimes (Mach number limitations). Thus, we use the sets of primitive variables, which boost solvability, such as pressure, velocity and either specific internal energy, enthalpy or temperature, as an energy transport variable. Note that residuals are always formed for mass, momentum and energy, to ensure conservation upon convergence of the non-linear solver, within the framework of the Newton-Krylov algorithm.

3.1. Spatial Discretization

The computational domain Ω is subdivided into a collection of non-overlapping linear QUAD4 (4-node) and HEX8 (8-node) elements, Ω_e. The solution is represented in the broken Sobolev space $\mathbb{V}_h^p$, consisting of discontinuous vector-values polynomial functions of degree p

$$\mathbb{V}_h^p = \left\{ \upsilon_h \in \left[\mathcal{L}_2\left(\Omega\right)\right]^m : \upsilon_h|_{\Omega_e} \in \left[\mathcal{V}_p^m\right] \forall \Omega_e \in \Omega \right\}, \tag{3.1}$$

where m is the dimension of the unknown vector and $\mathcal{V}_p$ is the space of all polynomials of degree $\leq p$. The governing equations (2.1), (2.2) and (2.3) are represented in the following weak formulation, which is obtained by multiplying by a test function $\boldsymbol{\mathcal{W}}_h$, integrating over an element Ω_e, and then

performing an integration by parts

$$\begin{aligned}\mathbf{R}_h\left(\mathbf{U}_h\right) = \frac{\partial}{\partial t}\int\limits_{\Omega_e}\mathbf{U}_h\boldsymbol{\mathcal{W}}_h d\Omega + \int\limits_{\Gamma_e}\left(\mathbf{F}_j\left(\mathbf{U}_h\right) - \mathbf{D}_j\left(\mathbf{U}_h\right)\right)n_j\boldsymbol{\mathcal{W}}_h d\Gamma - \\ - \int\limits_{\Omega_e}\left[\left(\mathbf{F}_j\left(\mathbf{U}_h\right) - \mathbf{D}_j\left(\mathbf{U}_h\right)\right)\frac{\partial\boldsymbol{\mathcal{W}}_h}{\partial x_j} + \mathbf{S}\left(\mathbf{U}_h\right)\boldsymbol{\mathcal{W}}_h\right]d\Omega,\ \forall\boldsymbol{\mathcal{W}}_h \in \mathbb{V}_h^p,\end{aligned} \tag{3.2}$$

where $\mathbf{U}_h$ and $\boldsymbol{\mathcal{W}}_h$ are represented by piecewise-polynomial functions of degrees p, which are discontinuous between the cell interfaces, and $\mathbf{n} = n_j$ denotes the unit outward normal vector to the element face Γ_e (i.e., the boundary of Ω_e). The local residual function $\mathbf{R}_h\left(\mathbf{U}_h\right)$ is an inner product between the solution residue representation (with a chosen set of basis functions) and the test functions, $\boldsymbol{\mathcal{W}}_h$. In our fully-implicit solution procedure, we are minimizing this inner product.

The hyperbolic flux function $\mathbf{F}_j\left(\mathbf{U}_h\right)n_j$ appearing in the face integral term of eq.(3.2) is replaced by a numerical Riemann flux function, $\mathbf{H}_j\left(\mathbf{U}_h^{\mathrm{L}}, \mathbf{U}_h^{\mathrm{R}}\right)n_j$, which is computed by some (approximate) Riemann solver (see [68] for details). Here, $\mathbf{U}_h^{\mathrm{L}}$ and $\mathbf{U}_h^{\mathrm{R}}$ are the conservative state vectors at the left and right side of the element boundary.

Numerical polynomial solutions $\mathbf{U}_h$ in each element are expressed using a chosen set of basis functions $\mathcal{B}_{(k)}\left(\mathbf{x}\right)$, as

$$\mathbf{U}_h\left(\mathbf{x}, t\right) = \sum_{k=0}^{K-1}\mathbf{U}_{(k)_e}\left(t\right)\mathcal{B}_{(k)}\left(\mathbf{x}\right), \tag{3.3}$$

where $\mathbf{U}_{(k)_e}$ denotes degrees of freedom (DoF) in an element e. Here, we use tensor-product *Legendre-polynomial-based* basis functions, described in [68]. It is instructive to note that, with these basis functions, the first degree of freedom is the *cell-averaged* quantity, which naturally connects this method to finite-volume methods. In addition, these basis functions are *modal* and *hierarchical*.

3.2. Temporal Discretization

To prevent severe time step restrictions due to either explicit [36] or semi-implicit [46] time discretizations, we use fully-implicit methods, employing L-stable time integrators, like BDF_2 or

the p^{th}-order *Explicit, Singly-Diagonal Implicit Runge-Kutta* schemes, ESDIRK$_p$, [10, 12, 68] [1]. With these, our time stepping is dictated by accuracy requirements, rather than by numerical stability, which can be prohibitively expensive in explicit or operator-splitting algorithms [36]. For low-Mach flows, there is a large discrepancy between the acoustic and material velocities, leading to a numerically stiff system. Furthermore, for melting and solidification phase change problems, there is a strong nonlinearity in the momentum equation, since the viscosity operator is now a variable coefficient diffusion operator that is a strong function of space and temperature. We are thus required to tightly couple all three conservation equations (mass, momentum, and energy) and cannot employ operator splitting strategies. By tightly coupling all of the physics, we can pick time steps significantly exceeding the material CFL number (without loss of accuracy), which is the major time-stepping limit in operator-splitting based algorithms [46].

3.3. Jacobian-Free Newton-Krylov (JFNK) Solver

In this Section, we briefly review the Jacobian-free Newton-Krylov (JFNK) framework. Once the equations are discretized in space and time, we seek to minimize the residual equation using a globalized line search method[2]. Newton's method is used to compute the step direction by solving the non-linear system of equations

$$\mathbf{F}(\mathbf{x}) = 0, \tag{3.4}$$

where $\mathbf{F}$ is the nonlinear residual function and $\mathbf{x}$ is the solution vector, representing all of the degrees of freedom. Using Newton's method, we iteratively find better roots to Eq. (3.4) by solving a sequence of linear problems

$$\mathbf{J}^k \delta\mathbf{x}^k = -\mathbf{F}(\mathbf{x}^k), \tag{3.5}$$

where the Jacobian matrix is defined as $\mathbf{J} \equiv \frac{\partial \mathbf{F}}{\partial \mathbf{x}}$. Once the update vector, $\delta\mathbf{x}^k$, is solved for, it is added to the previous non-linear solution vector

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x}^k, \tag{3.6}$$

[1] Most of the calculations in this work were done with the 2$^{\text{nd}}$-order L-stable BDF$_2$ time integrator.
[2] In this work, we use either the backtracking or critical point line search strategies in PETSc [7].

until the Newton convergence criterion is satisfied

$$||\mathbf{F}(\mathbf{x}^k)||_2 < \text{tol}_N ||\mathbf{F}(\mathbf{x}^0)||_2. \tag{3.7}$$

In this study, we choose a relative Newton tolerance of $\text{tol}_N = 10^{-5}$. For the linear solver, we use the Arnoldi-based Generalized Minimal Residual method (GMRES) [75]. Since GMRES does not require individual elements of the Jacobian matrix, only the action of matrix-vector products are required and an explicit Jacobian matrix does not need to be formed. The action of the Jacobian matrix-vector products is approximated by Fréchet derivatives

$$J\vec{\kappa} \approx \frac{F(\mathbf{x} + \epsilon\vec{\kappa}) - F(\mathbf{x})}{\epsilon}, \tag{3.8}$$

where ϵ is a small but finite number and $\vec{\kappa}$ is a Krylov vector. Eq. (3.8) is a first-order Taylor series expansion of the Jacobian times a vector, $\vec{\kappa}$. An inexact Newton method is used to ensure that the linear system is tightly solved only when the accuracy matters – i.e. at the end of the nonlinear iterations. With this approach, the convergence criteria of the linear residual is proportional to the non-linear residual

$$||\mathbf{J}^k \delta\mathbf{x}^k + \mathbf{F}(\mathbf{x}^k)||_2 < \text{tol}_L ||\mathbf{F}(\mathbf{x}^k)||_2, \tag{3.9}$$

where tol_L is a constant[3]. Our (inexact) JFNK solver is implemented within PETSc, a high-performance suite of non-linear and linear solvers developed at the Argonne National Laboratory [7].

Because GMRES stores all of the previous vectors that form the Krylov basis, it is necessary to keep the number of iterations relatively small, to prevent the storage and CPU time from becoming prohibitive. This is accomplished by preconditioning the linear system. A mathematically good preconditioner should efficiently cluster the eigenvalues of the iteration matrix [40, 74]. Finding an efficient preconditioner is often a combination of art, science, and intuition.

[3] In the present study, we use $\text{tol}_L = 10^{-8}$.

3.4. Preconditioning

Although the governing equations are discretized in conservative form (mass, momentum, total energy), we choose to solve for the primitive set of variables, $[P\mathbf{v}T]$ (pressure, velocity, temperature), since it is a better conditioned set of variables for low-speed flow [15, 64]. Introducing the transformation

$$\delta\mathbf{U} = \frac{\partial\mathbf{U}}{\partial\mathbf{W}}\delta\mathbf{W}, \tag{3.10}$$

where $\mathbf{U} = (\rho, \rho\mathbf{v}, E)$ is the vector of conservative variables and $\mathbf{W} = (P, \mathbf{v}, T)$ is the vector of primitive variables. The linear system in Eq. (3.5) can now be transformed to

$$\frac{\partial\mathbf{F}(\mathbf{x})}{\partial\mathbf{U}}\frac{\partial\mathbf{U}}{\partial\mathbf{W}}\delta\mathbf{W} = -\mathbf{F}(\mathbf{x}). \tag{3.11}$$

It is important to emphasize that the change of variables does not affect conservation, since the residual function is still written to satisfy the underlying conservation laws. The equations are thus conserved to the non-linear tolerance level. One of the great strengths of the non-linear Newton-Krylov algorithm is that it is not required to solve for conservative variables. Instead, one can solve for another (mathematically equivalent) set of unknowns, which will render a better conditioned system.

The right-preconditioned form of the system is

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{M}\delta\mathbf{x} = -\mathbf{F}(\mathbf{x}), \tag{3.12}$$

where $\mathbf{M}$ is the preconditioning matrix[4]. Taking $\mathbf{M}$ as the approximate (finite-differenced) Jacobian matrix, the degrees of freedom can be ordered by physics fields in a 3×3 block matrix

$$\begin{bmatrix} \mathbf{M}_{\mathbf{vv}} & \mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{\mathbf{v}T} \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{M}_{PP} & \mathbf{M}_{PT} \\ \mathbf{M}_{TV} & \mathbf{M}_{TP} & \mathbf{M}_{TT} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathbf{v}} \\ \mathbf{x}_P \\ \mathbf{x}_T \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\mathbf{v}} \\ \mathbf{b}_P \\ \mathbf{b}_T \end{bmatrix}, \tag{3.13}$$

[4]Although no matrix needs to be explicitly formed for the Jacobian-free Newton-Krylov (JFNK) method, we choose to explicitly form the matrix, $\mathbf{M}$, for preconditioning purposes.

where $\mathbf{b}$ is an incoming Krylov vector and $\mathbf{x}$ is the outgoing Krylov (solution) vector. Each block in Eq. (3.13) is a block matrix of size $n_{\text{elems}} \times n_{\text{eqns}} \times n_{\text{DoFs/elem}}$. The blocks corresponding to degrees of freedom associated with velocity, for example, have two equations in 2D and three equations in 3D (v_x, v_y, v_z).

3.4.1. High-Order rDG. Since the P_0P_1 scheme only solves for cell-averaged quantities, the sub-blocks in Eq. (3.13) represent scalar (sparse) matrices for degrees of freedom corresponding to that physics field. The higher-order schemes such as, P_1P_3 or P_2P_3, however, have additional degrees of freedom per equation per element (as listed in Table 3.2). In the higher-order schemes, each block in Eq. (3.13) is now a block matrix itself, e.g. each physics block for the rDG P_2P_3 scheme in 2D is a 6×6 block system corresponding to a cell-averaged value + 2 slopes + 3 curvatures

$$\mathbf{M}_{PP} = \begin{pmatrix} \mathbf{M}_{P_0P_0} & \mathbf{M}_{P_0P_x} & \mathbf{M}_{P_0P_y} & \mathbf{M}_{P_0P_{xx}} & \mathbf{M}_{P_0P_{yy}} & \mathbf{M}_{P_0P_{xy}} \\ \mathbf{M}_{P_xP_0} & \mathbf{M}_{P_xP_x} & \mathbf{M}_{P_xP_y} & \mathbf{M}_{P_xP_{xx}} & \mathbf{M}_{P_xP_{yy}} & \mathbf{M}_{P_xP_{xy}} \\ \mathbf{M}_{P_yP_0} & \mathbf{M}_{P_yP_x} & \mathbf{M}_{P_yP_y} & \mathbf{M}_{P_yP_{xx}} & \mathbf{M}_{P_yP_{yy}} & \mathbf{M}_{P_yP_{xy}} \\ \mathbf{M}_{P_{xx}P_0} & \mathbf{M}_{P_{xx}P_x} & \mathbf{M}_{P_{xx}P_y} & \mathbf{M}_{P_{xx}P_{xx}} & \mathbf{M}_{P_{xx}P_{yy}} & \mathbf{M}_{P_{xx}P_{xy}} \\ \mathbf{M}_{P_{yy}P_0} & \mathbf{M}_{P_{yy}P_x} & \mathbf{M}_{P_{yy}P_y} & \mathbf{M}_{P_{yy}P_{xx}} & \mathbf{M}_{P_{yy}P_{yy}} & \mathbf{M}_{P_{yy}P_{xy}} \\ \mathbf{M}_{P_{xy}P_0} & \mathbf{M}_{P_{xy}P_x} & \mathbf{M}_{P_{xy}P_y} & \mathbf{M}_{P_{xy}P_{xx}} & \mathbf{M}_{P_{xy}P_{yy}} & \mathbf{M}_{P_{xy}P_{xy}} \end{pmatrix} \tag{3.14}$$

$$\mathbf{M}_{\mathbf{vv}} = \begin{pmatrix} \mathbf{M}_{\mathbf{vv}_0\mathbf{vv}_0} & \mathbf{M}_{\mathbf{vv}_0\mathbf{vv}_x} & \mathbf{M}_{\mathbf{vv}_0\mathbf{vv}_y} & \mathbf{M}_{\mathbf{vv}_0\mathbf{vv}_{xx}} & \mathbf{M}_{\mathbf{vv}_0\mathbf{vv}_{yy}} & \mathbf{M}_{\mathbf{vv}_0\mathbf{vv}_{xy}} \\ \mathbf{M}_{\mathbf{vv}_x\mathbf{vv}_0} & \mathbf{M}_{\mathbf{vv}_x\mathbf{vv}_x} & \mathbf{M}_{\mathbf{vv}_x\mathbf{vv}_y} & \mathbf{M}_{\mathbf{vv}_x\mathbf{vv}_{xx}} & \mathbf{M}_{\mathbf{vv}_x\mathbf{vv}_{yy}} & \mathbf{M}_{\mathbf{vv}_x\mathbf{vv}_{xy}} \\ \mathbf{M}_{\mathbf{vv}_y\mathbf{vv}_0} & \mathbf{M}_{\mathbf{vv}_y\mathbf{vv}_x} & \mathbf{M}_{\mathbf{vv}_y\mathbf{vv}_y} & \mathbf{M}_{\mathbf{vv}_y\mathbf{vv}_{xx}} & \mathbf{M}_{\mathbf{vv}_y\mathbf{vv}_{yy}} & \mathbf{M}_{\mathbf{vv}_y\mathbf{vv}_{xy}} \\ \mathbf{M}_{\mathbf{vv}_{xx}\mathbf{vv}_0} & \mathbf{M}_{\mathbf{vv}_{xx}\mathbf{vv}_x} & \mathbf{M}_{\mathbf{vv}_{xx}\mathbf{vv}_y} & \mathbf{M}_{\mathbf{vv}_{xx}\mathbf{vv}_{xx}} & \mathbf{M}_{\mathbf{vv}_{xx}\mathbf{vv}_{yy}} & \mathbf{M}_{\mathbf{vv}_{xx}\mathbf{vv}_{xy}} \\ \mathbf{M}_{\mathbf{vv}_{yy}\mathbf{vv}_0} & \mathbf{M}_{\mathbf{vv}_{yy}\mathbf{vv}_x} & \mathbf{M}_{\mathbf{vv}_{yy}\mathbf{vv}_y} & \mathbf{M}_{\mathbf{vv}_{yy}\mathbf{vv}_{xx}} & \mathbf{M}_{\mathbf{vv}_{yy}\mathbf{vv}_{yy}} & \mathbf{M}_{\mathbf{vv}_{yy}\mathbf{vv}_{xy}} \\ \mathbf{M}_{\mathbf{vv}_{xy}\mathbf{vv}_0} & \mathbf{M}_{\mathbf{vv}_{xy}\mathbf{vv}_x} & \mathbf{M}_{\mathbf{vv}_{xy}\mathbf{vv}_y} & \mathbf{M}_{\mathbf{vv}_{xy}\mathbf{vv}_{xx}} & \mathbf{M}_{\mathbf{vv}_{xy}\mathbf{vv}_{yy}} & \mathbf{M}_{\mathbf{vv}_{xy}\mathbf{vv}_{xy}} \end{pmatrix} \tag{3.15}$$

$$\mathbf{M}_{TT} = \begin{pmatrix} \mathbf{M}_{T_0T_0} & \mathbf{M}_{T_0T_x} & \mathbf{M}_{T_0T_y} & \mathbf{M}_{T_0T_{xx}} & \mathbf{M}_{T_0T_{yy}} & \mathbf{M}_{T_0T_{xy}} \\ \mathbf{M}_{T_xT_0} & \mathbf{M}_{T_xT_x} & \mathbf{M}_{T_xT_y} & \mathbf{M}_{T_xT_{xx}} & \mathbf{M}_{T_xT_{yy}} & \mathbf{M}_{T_xT_{xy}} \\ \mathbf{M}_{T_yT_0} & \mathbf{M}_{T_yT_x} & \mathbf{M}_{T_yT_y} & \mathbf{M}_{T_yT_{xx}} & \mathbf{M}_{T_yT_{yy}} & \mathbf{M}_{T_yT_{xy}} \\ \mathbf{M}_{T_{xx}T_0} & \mathbf{M}_{T_{xx}T_x} & \mathbf{M}_{T_{xx}T_y} & \mathbf{M}_{T_{xx}T_{xx}} & \mathbf{M}_{T_{xx}T_{yy}} & \mathbf{M}_{T_{xx}T_{xy}} \\ \mathbf{M}_{T_{yy}T_0} & \mathbf{M}_{T_{yy}T_x} & \mathbf{M}_{T_{yy}T_y} & \mathbf{M}_{T_{yy}T_{xx}} & \mathbf{M}_{T_{yy}T_{yy}} & \mathbf{M}_{T_{yy}T_{xy}} \\ \mathbf{M}_{T_{xy}T_0} & \mathbf{M}_{T_{xy}T_x} & \mathbf{M}_{T_{xy}T_y} & \mathbf{M}_{T_{xy}T_{xx}} & \mathbf{M}_{T_{xy}T_{yy}} & \mathbf{M}_{T_{xy}T_{xy}} \end{pmatrix}. \tag{3.16}$$

In 3D, these matrices would be 10×10 block-matrices, instead of 6×6 block systems.

CHAPTER 4

Code Verification and Mesh Convergence

Since finding analytical solutions to the Navier-Stokes equations is difficult, the Method of Manufactured Solutions (MMS) provides a technique to measure the numerical errors associated with the discrete equations. This allows for the determination of the numerical scheme's order of accuracy, without using an analytical solution. In Section 4.1, we use the MMS to verify the convergence rates in the L^2 norm for different rDG spatial discretization schemes. In Section 4.2, we compare the computational efficiency between a high-order and a low-order rDG scheme by conducting a qualitative convergence study.

4.1. Method of Manufactured Solutions for the Compressible Navier-Stokes Equations

To test convergence in space and time for the problem with both hyperbolic and diffusion operators, the following solution is manufactured in 2D

$$\begin{aligned} T(x,y) &= \bar{T} + A_{_T} \cos\left(2\pi\left(x + \mathrm{v}_0 \mathrm{t}\right)\right) \sin\left(2\pi\left(y + \mathrm{v}_1 \mathrm{t}\right)\right) \\ P(x,y) &= \bar{P} + A_{_P} \sin\left(2\pi\left(x + \mathrm{v}_0 \mathrm{t}\right)\right) \cos\left(2\pi\left(y + \mathrm{v}_1 \mathrm{t}\right)\right) \\ v_0(x,y) &= A_{_v} \cos\left(2\pi\left(x + \mathrm{v}_0 \mathrm{t}\right)\right) \sin\left(2\pi\left(y + \mathrm{v}_1 \mathrm{t}\right)\right) \\ v_1(x,y) &= A_{_v} \sin\left(2\pi\left(x + \mathrm{v}_0 \mathrm{t}\right)\right) \cos\left(2\pi\left(y + \mathrm{v}_1 \mathrm{t}\right)\right), \end{aligned} \tag{4.1}$$

where

$$\begin{aligned} A_{_T} &= \delta T_0 + a_{_T} \sin\left(2\pi t\right) \\ A_{_P} &= \delta P_0 + a_{_P} \sin\left(2\pi t\right) \\ A_{_v} &= \delta V_0 + a_{_v} \sin\left(2\pi t\right), \end{aligned} \tag{4.2}$$

and $\bar{T}$, $\bar{P}$, δT_0, δP_0, δV_0, $a_{_T}$, $a_{_P}$, $a_{_v}$, v_0, v_1 are given constants.

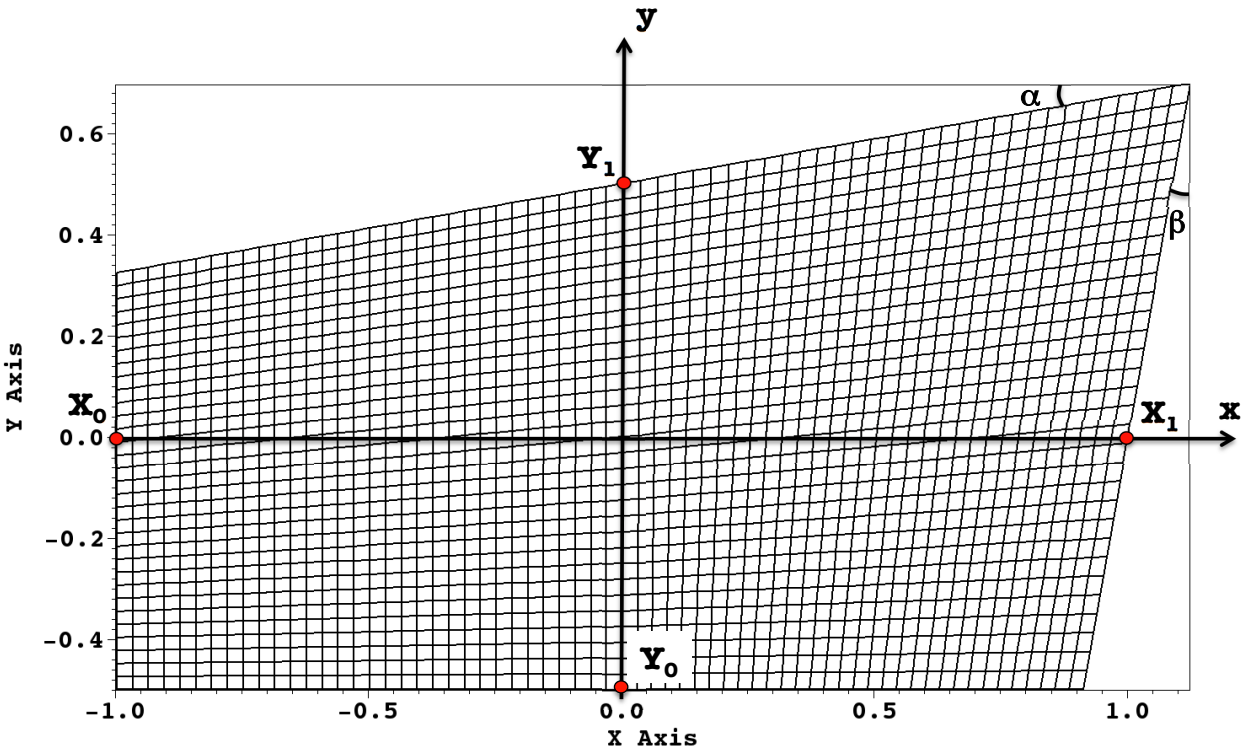


FIGURE 4.1. On domain and mesh setup for 2D manufactured solution test problem. $\alpha = \beta = 10^o$.

Solution Eq. (4.1) corresponds to translating (with velocity $\mathbf{w} = (\mathrm{v}_0, \mathrm{v}_1)$) and oscillating (with amplitudes a_T, a_P and a_v) waves. In the following simulations, we set

$\mathbf{w}$	$=$	$(\frac{1}{10}, \frac{1}{10})$
$\bar{P}$	$=$	1.0
$\bar{T}$	$=$	1.0
δP_0	$=$	0.1
δT_0	$=$	0.1
δV_0	$=$	10^{-4}
a_P	$=$	0.05
a_v	$=$	0.01
a_T	$=$	0.05.

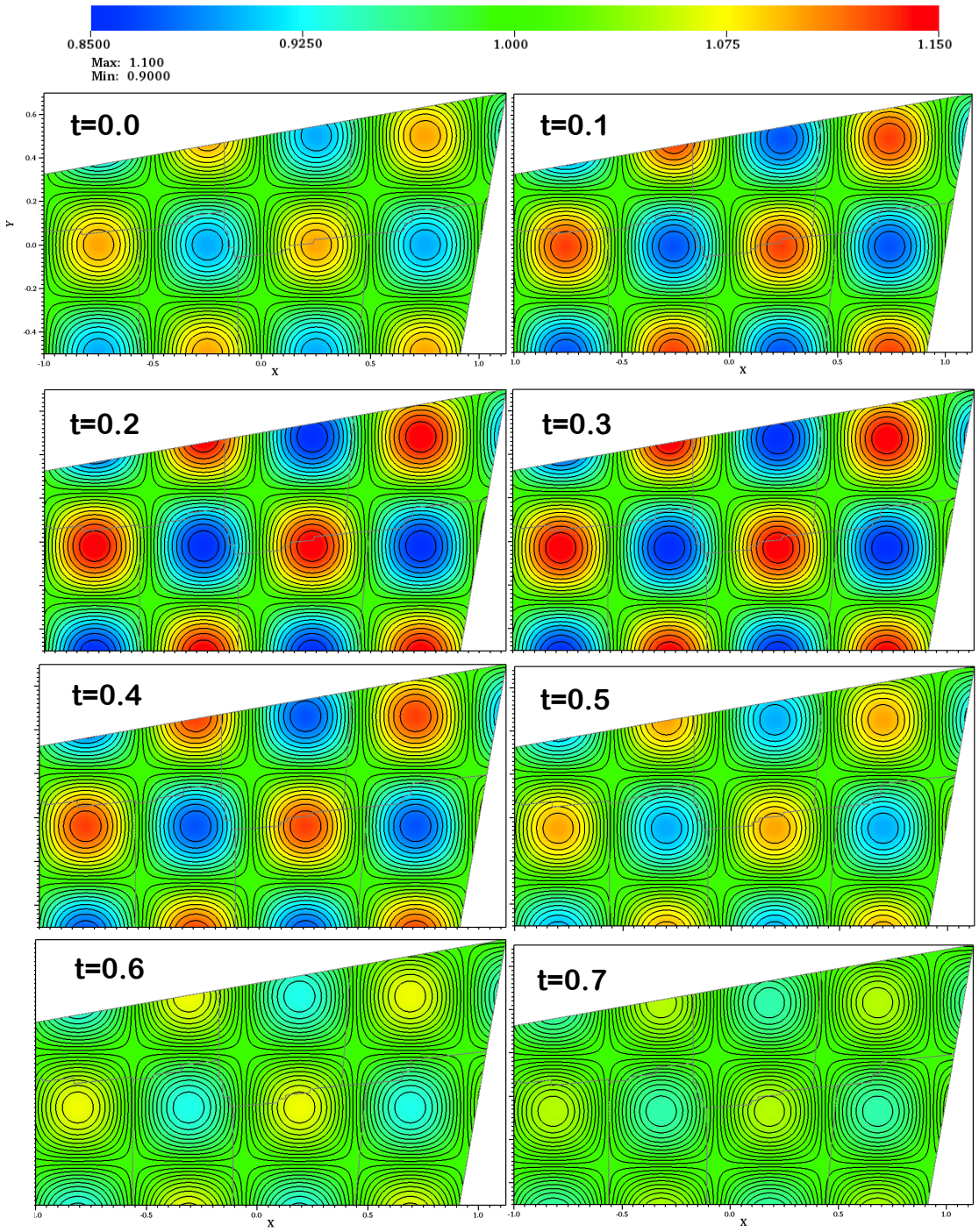


FIGURE 4.2. Dynamics of the pressure field for manufactured problem, using $\mathrm{rDG}_{\mathrm{P_2P_3}}$, ESDIRK_5, $\Delta t = 0.1$, 32,762 elements.

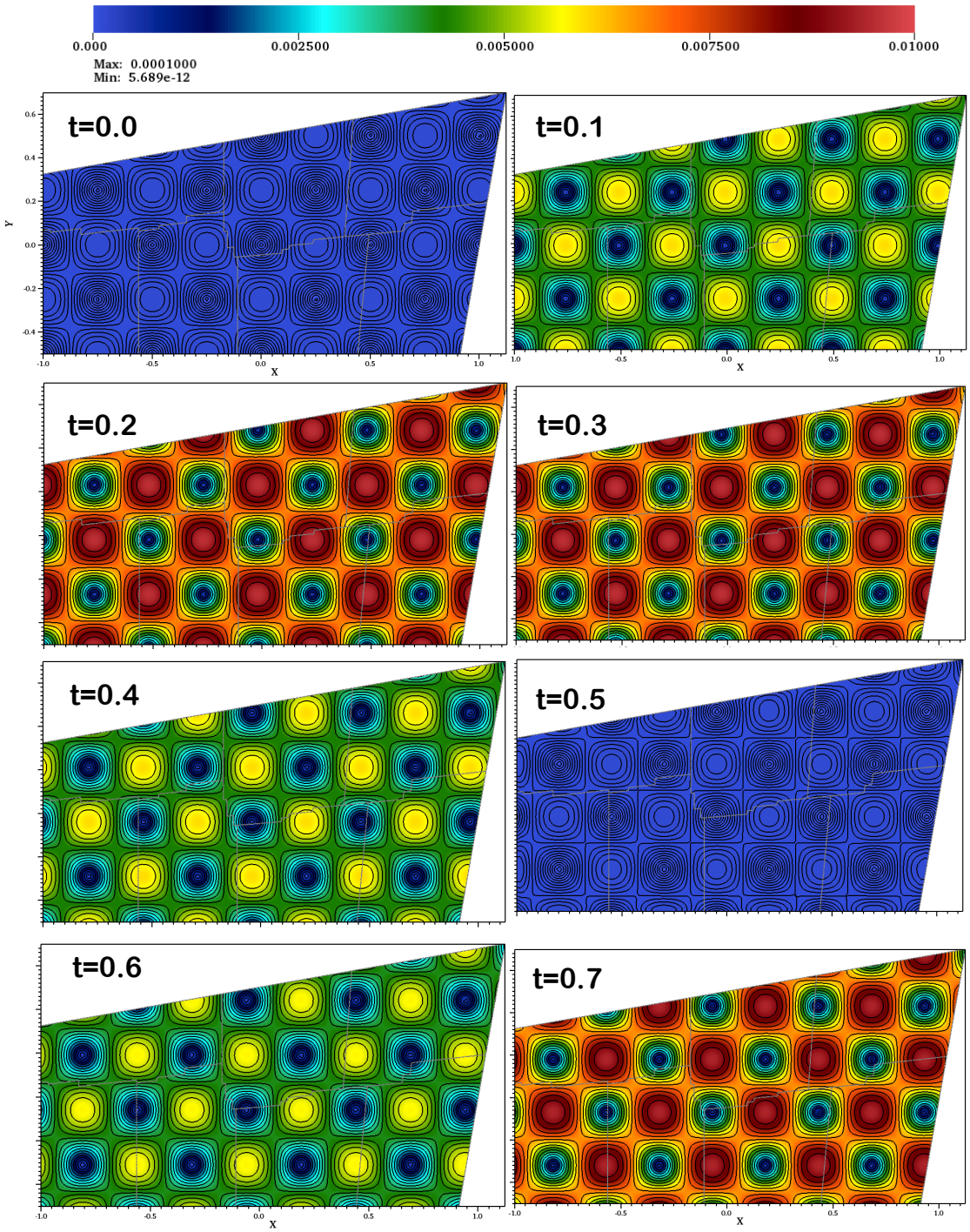


FIGURE 4.3. Dynamics of the velocity field for manufactured problem, using $\mathrm{rDG}_{\mathrm{P_2P_3}}$, ESDIRK_5, $\Delta t = 0.1$, 32,762 elements.

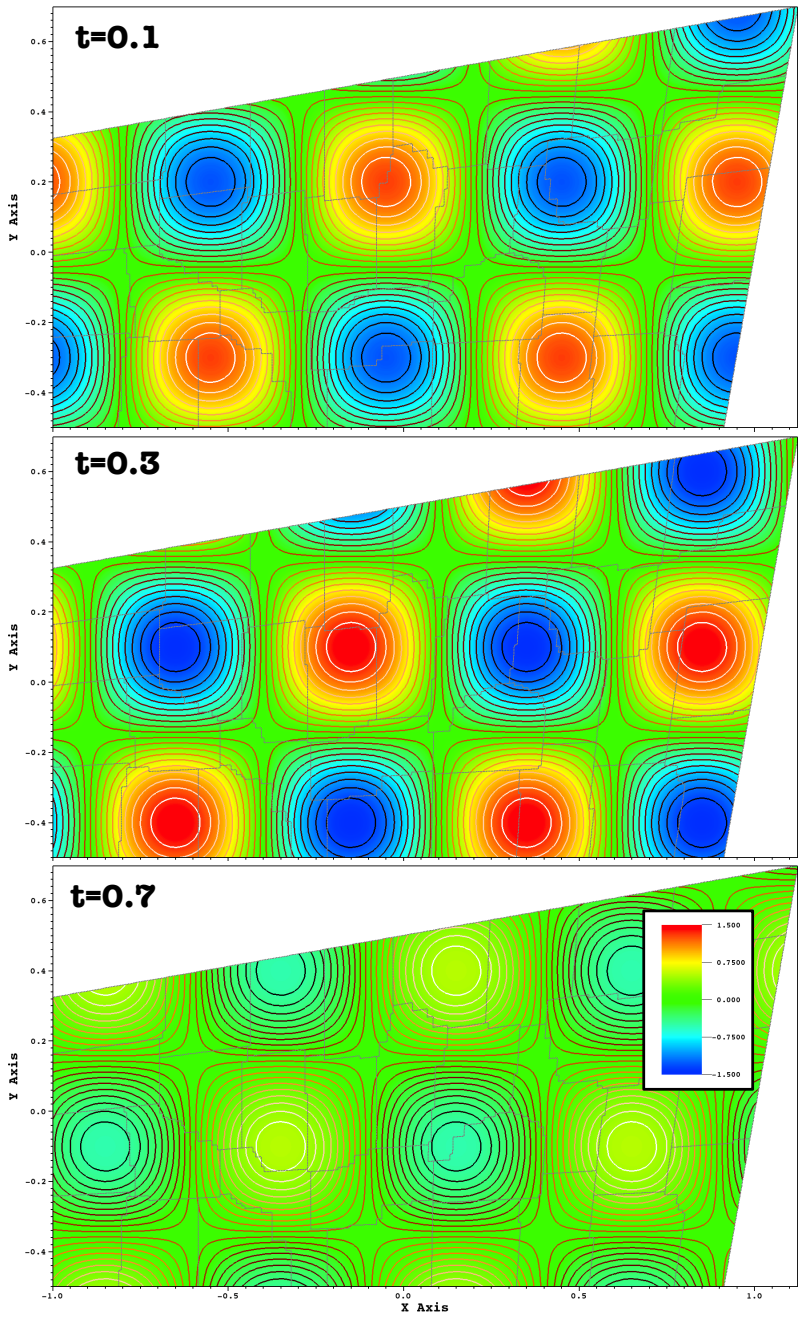


FIGURE 4.4. Dynamics of the temperature field for manufactured problem, using $\mathrm{rDG}_{\mathrm{P_2P_3}}$, ESDIRK_5, $\Delta t = 0.1$, 32,762 elements.

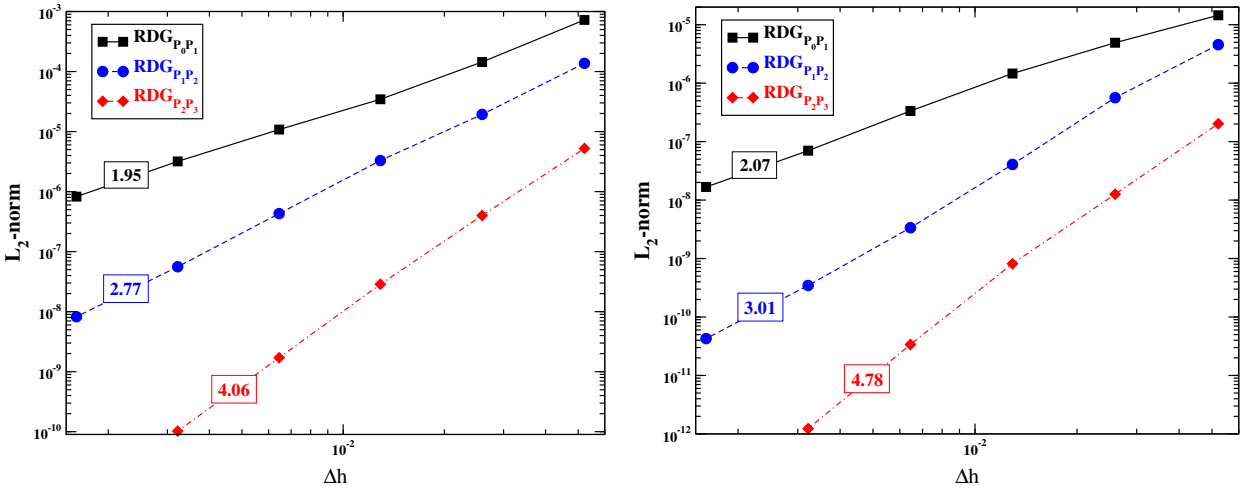


FIGURE 4.5. On convergence of the pressure (left) and velocity (right) fields, with mesh refinement and different space discretization schemes.

The γ-gas equation of state with $\gamma = 1.4$ was used. Both thermal conductivity and dynamic viscosity are set to be constant, $\kappa = 0.1$ and $\mu = 0.1$. Source terms generating this manufactured solution are computed using symbolic manipulation in *Mathematica*. Once the source terms are generated, the governing equations are solved on multiple grids with different mesh resolutions using the generated source terms.

The domain and mesh are shown in Figure 4.1. Using high-fidelity space and time resolution, the dynamics of pressure and velocity magnitude, and temperature fields are shown in Figures 4.2-4.4.

First, we measure space convergence rates, by using the 5^{th}-order-accurate time discretization ESDIRK$_5$ and setting time step to $2 \times \Delta t = 0.001$. This ensures that time discretization errors are smaller than space discretization errors. The results are shown in Figure 4.5. As one can see, all three of the rDG schemes converge consistently – i.e., with second order for rDG$_{P_0P_1}$, with third order for rDG$_{P_1P_2}$, and with fourth-order for rDG$_{P_2P_3}$.

In the simulations, we used rDG$_{P_2P_3}$ on the mesh with 32,762 elements, to ensure small spatial discretization errors. This space resolution is sufficient to measure nearly asymptotic convergence

rates for time discretization schemes up to the 3rd order accurate. The fourth- and the fifth-order accurate $\text{ESDIRK}_{4,5}$ schemes exhibit nearly 4^{th} order convergence rate, when time steps are large. The convergence is flatten for smaller time steps, when space discretization errors become dominant.

4.2. Mesh Convergence of the P_0P_1 and P_2P_3 rDG schemes

In this Section, we compare the computational efficiency between the high-order, $\text{rDG}_{\text{P}_2\text{P}_3}$, and the low-order, $\text{rDG}_{\text{P}_0\text{P}_1}$, schemes for an internally heated, laser-induced melt convection problem. Recall that the laser model is defined in Section 2.3. The figure of merit is to qualitatively resolve all four unstable eddies/vortices at a non-dimensional time of $\hat{t} = 35$.

In Figure 4.6, the material is initially solid steel at a non-dimensional temperature of $\hat{T} = 0$ on the same non-uniform mesh in Section 6.3. The laser spot of radius 0.1 is centered at (0,0), internally heating the material, inducing unsteady melt convection. The bottom wall has a Dirichlet boundary condition with temperature fixed at $\hat{T} = 0$, while all other walls have Neumann boundary conditions for temperature with zero heat flux. All four walls enforce a no-slip boundary condition on velocity. The melt temperatures are $T_{solidus} = 0.95$ and $T_{liquidus} = 1.05$, corresponding to a mushy region thickness of $\epsilon = 0.1$. The Rayleigh, Prandtl, and Stefan numbers are: $\text{Ra} = 10^5$, $\text{Pr} = 0.13$, and $\text{Ste} = 8$. In all runs, we used a time step, $\Delta t = 0.1$.

We refine the mesh, successively, until both the low-order and high-order schemes visually resolve all four eddies with streamlines, as seen in Figure 4.6. On the left column, we observe that the fourth eddie, V4, is finally captured with the 2^{nd} order accurate, $\text{rDG}_{\text{P}_0\text{P}_1}$ scheme, on a fine, 320×160 mesh. On the right column, however, the 4^{th}-order accurate, $\text{rDG}_{\text{P}_2\text{P}_3}$ scheme, captured the fourth eddie with a very coarse, 48×24, mesh. For both schemes, we further refine the mesh in order to verify that the solution has indeed converged.

In Figure 4.7, we compare both of the fully resolved cases, side by side. Comparing the two

meshes, the $\text{rDG}_{\text{P}_0\text{P}_1}$ scheme required 45 times more elements than the $\text{rDG}_{\text{P}_2\text{P}_3}$ scheme. To compare the number of degrees of freedom used, we note that in 2D, the 4^{th}-order scheme has 6 degrees of freedom per equation per element, while the 2^{nd}-order scheme has 1 degree of freedom per equation per element. Since the fourth order scheme has 6 times more degrees of freedom per element, 45 times less elements translates to a factor of 7.5 times less total degrees of freedom. The 4^{th}-order scheme, however, only converged in half of the CPU runtime compared to the 2^{nd}-order scheme, because high-order methods are more computationally expensive per degree of freedom. We also note that since the high-order method required only a coarse mesh, the CFL and Fourier numbers, defined in Appendix A.2, were much lower: $\text{CFL}_{\text{aco}} = 378$, $\text{CFL}_{\text{mat}} = 0.87$ and $\text{Fo}_{\nu} = 162$, $\text{Fo}_{\alpha} = 1.2$, as compared to much higher CFL and Fourier numbers with the low-order method: $\text{CFL}_{\text{aco}} = 2500$, $\text{CFL}_{\text{mat}} = 6$ and $\text{Fo}_{\nu} = 7000$, $\text{Fo}_{\alpha} = 55$. Since the low-order case had significantly higher CFL and Fourier numbers, the underlying linear systems are more ill-conditioned/stiff, severely hindering scalability.

When we lowered the Rayleigh number, $\text{Ra} \leq 10^3$, for the same test problem, the convergence study showed that the low order rDG-based scheme on a coarse mesh was sufficient to capture the dynamics. This convergence study demonstrates that for Rayleigh numbers above $\text{Ra} \geq 10^5$ or when the flow has several unstable vortices, the high-order rDG-based scheme is more computationally efficient than the low-order rDG-based scheme, for the same qualitative solution.

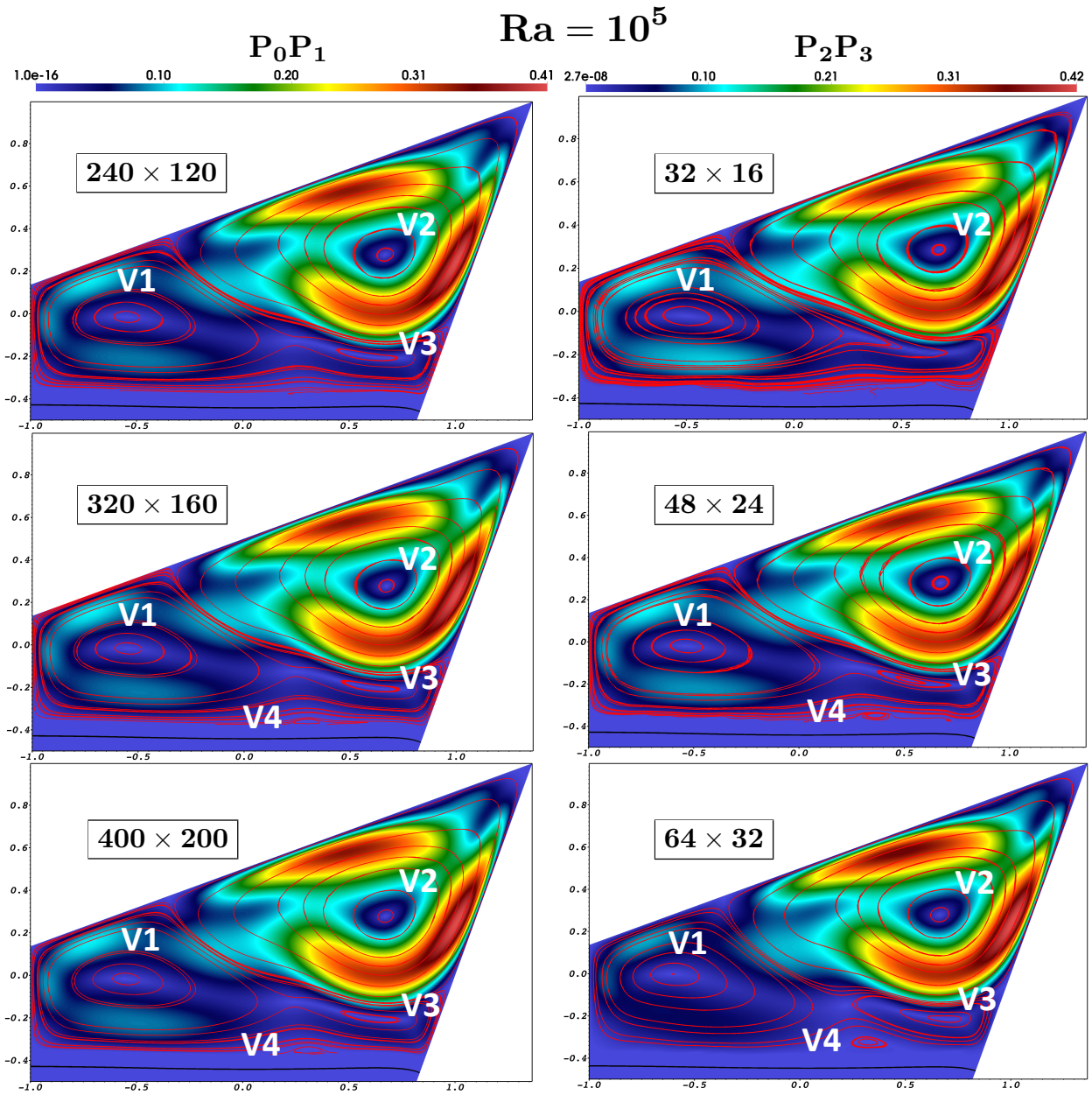
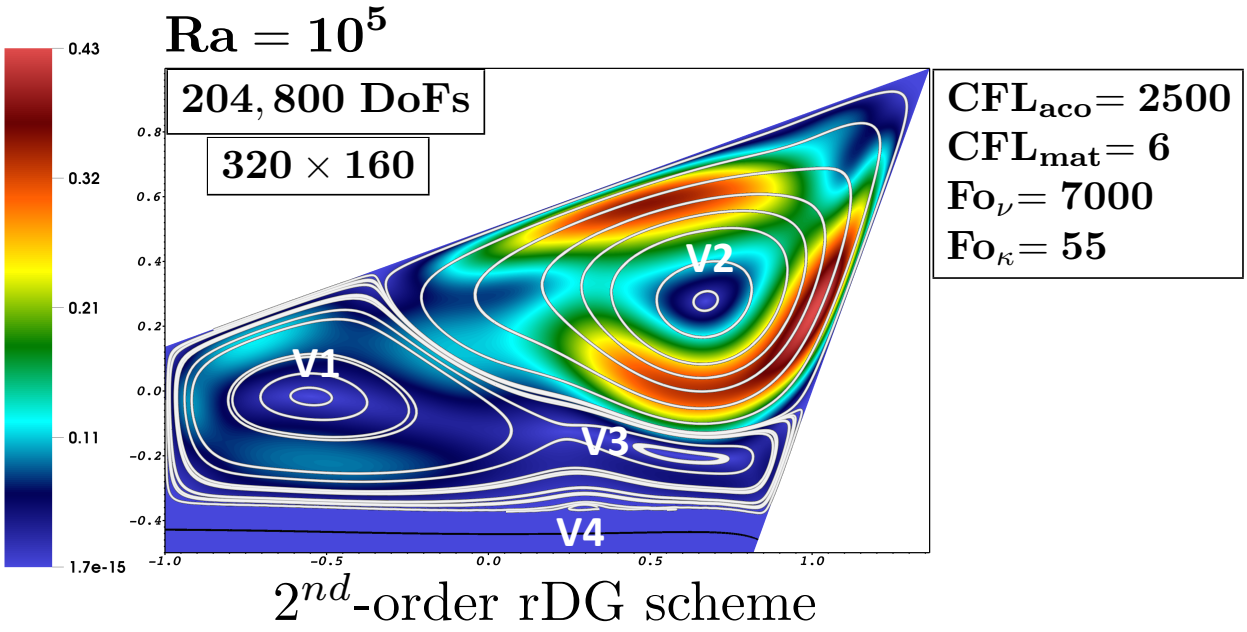


FIGURE 4.6. Qualitative convergence study of the low-order rDG scheme, left column, and the high-order rDG scheme, right column, for an internally heated melt-convection problem. The figures show velocity magnitude with streamlines in red at a non-dimensional time: $\hat{t} = 35$. All four vortices, V1-V4, indicate convergence.



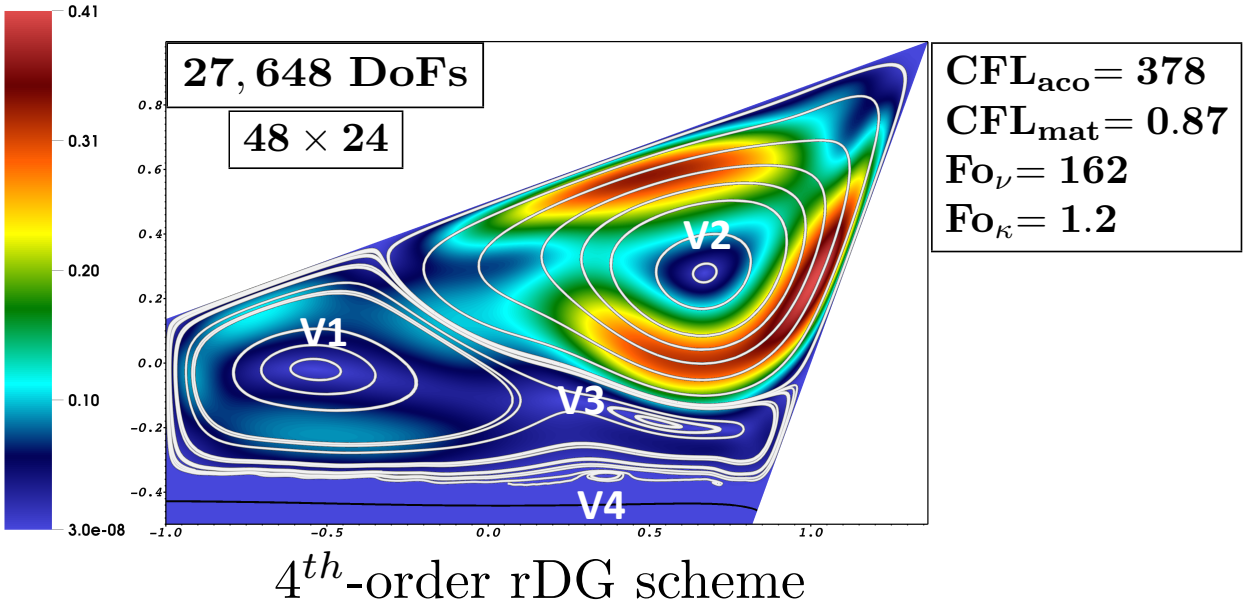


FIGURE 4.7. Side by side comparison showing the number of elements, number of degrees of freedom, and CFL/Fourier numbers for both the low-order (top figure) and high-order (bottom figure) rDG schemes.

CHAPTER 5

Solidification Model Results

In this Chapter, we analyze different combinations of the velocity suppression models, as described in Section 2.2.2. Since we are primarily interested in modeling melt convection, we only require a simple strength model that enforces the solid phase to be static. Therefore, to discriminate between the velocity suppression models, the primary figure of merit is the magnitude of velocity in the solid. Since the drag force model is a function of the drag coefficient, C, and the enhanced viscosity model is a function of the viscosity ratio, $\frac{\mu_s}{\mu_l}$, we test different combinations of the two parameters that inhibit the motion of the solid and mushy regions.

5.1. Problem Formulation

For a laser-induced melt convection problem, as seen in Figure 5.1, we discriminate between the various velocity suppression models by measuring the velocity magnitude as a function of position in the solid and mushy phases for each of the tested models. We use the material values defined in Appendix A.3. The laser has a power of 200W and moves at 2000 mm/s. For this problem, we choose to use a large mushy region to measure the velocity drop-off across many grid cells, $\mathrm{T_{solidus}} = 945\mathrm{K}$ and $\mathrm{T_{liquidus}} = 2355\mathrm{K}$. The test problem is run on a 64×64 mesh using the 2^{nd}-order accurate, $\mathrm{rDG_{P_0P_1}}$ spatial discretization scheme. The measurements were made after 15 time steps using a $\Delta t = 2.7\mu\mathrm{s}$ and using a 2^{nd}-order, $\mathrm{BDF_2}$ time discretization scheme.

5.2. Analysis of Velocity Suppression Models

The results for all of the velocity suppression models are shown in Figure 5.2. For the top four cases, we show results for varying drag coefficients **across** the four plots and varying viscosity ratio's **within** each plot. For the bottom four cases, we show results for varying viscosity ratio's

across the four plots and varying drag coefficients **within** each plot. For the lower two drag coefficient cases, $C = 0$ and $C = 10,000$, we observe that the models with a higher viscosity ratio correspond to smaller velocity magnitude values in the solid, as we would expect. For the higher drag coefficient cases, $C = 100,000$ and $C = 500,000$, we observe that the higher viscosity ratio's, however, correspond to larger velocity magnitudes in the solid. This trend is expected because as both parameters become very large, the viscous term begins to compete with the drag term, reducing the efficacy of the drag model. Note that velocity magnitudes below 10^{-13} are in the range of floating point roundoff and are negligible.

With the exception of the pure viscosity models, all of the tested velocity suppression models have sufficient levels of velocity suppression in the solid. Thus we further discriminate between the models by considering the computational efficiency (CPU wall-time) of the various models, which is ultimately dependent on the underlying non-linear/linear solvers and choice of preconditioner. Recall that the enhanced viscosity models introduce strong nonlinearities in the parabolic (viscous) operator while the drag force models introduce strong nonlinearities in the source term operator. As a result, we select a phase change model based on the choice of preconditioner, which is selected based on the stiffness of the particular problem (large or small CFL numbers). For the externally induced phase change problems (melting/freezing from the wall) with thick mushy regions, we use a viscosity ratio of $\frac{\mu_s}{\mu_l} = 10,000$ and $C = 500,000$. For the laser-induced melt convection problems with thin mushy regions, we use a pure drag model $\left(\frac{\mu_s}{\mu_l} = 1\right)$ with $C = 500,000$, in order to avoid large viscosity jumps across the thin interface.

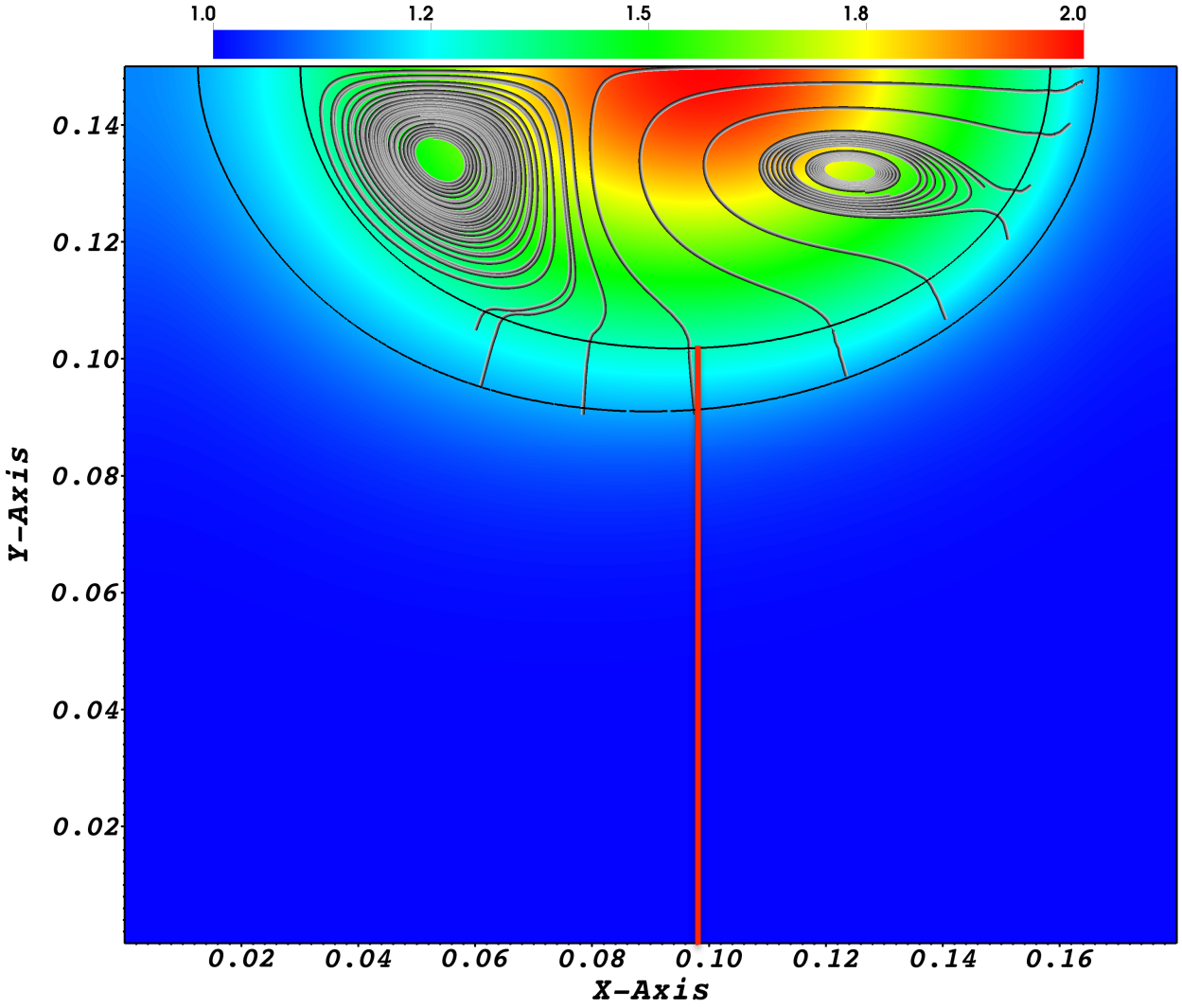


FIGURE 5.1. Laser-induced melt pool showing temperature and streamlines at non-dimensional time, $\hat{t} = 1.5 \times 10^{-3}$. The velocity magnitude is measured on the the vertical (red) line and used in Figure 5.2.

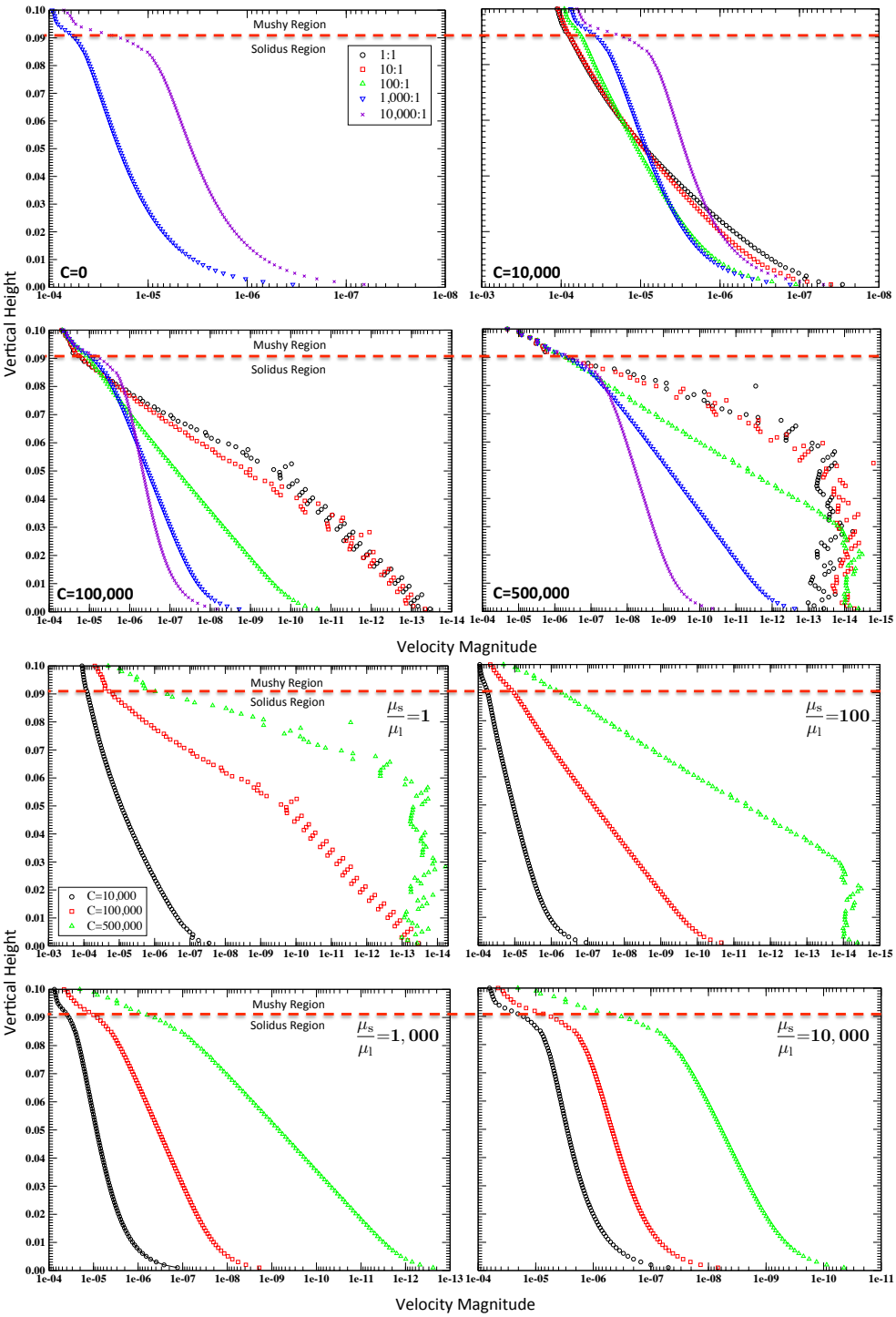


FIGURE 5.2. Plots of velocity magnitude vs. vertical height. The top four figures vary the drag coefficients across the plots and vary the viscosity ratio's within each plot. The bottom four figures vary the viscosity ratio's across the plots and vary the drag coefficients within each plot.

CHAPTER 6

Parametric Studies

In this Chapter, we demonstrate that our numerical framework can robustly handle a wide range of melt convection problems. To accomplish this, we independently scale each dimensionless number, analyzing the sensitivity of the melt convection dynamics to the non-dimensional parameters. Different melting and solidification configurations are tested, verifying the ability to simulate both melting and solidification problems. As described in Section 2.4, our non-dimensionalized equations for melt convection problems depend on three dimensionless numbers: the Rayleigh number, Ra, the Prandtl number, Pr, and the Stefan number, Ste.

In each of the following Sections, we vary one of the non-dimensional numbers while holding the other two constant. When held constant, the dimensionless numbers are: $Ste = 5$, $Pr = 0.1$, and $Ra = 10^6$.

6.1. Prandtl Number Effects

In this Section, we vary the Prandtl number for three cases: $Pr = 0.01$ (mercury), 0.1 (steel), 7 (water), while holding the Rayleigh number, $Ra = 10^6$, and Stefan number, $Ste = 5$, constant.

In all of the cases, the material is initially held at a non-dimensional temperature of $\hat{T} = 2$. The temperature boundary condition at the left and right walls have Dirichlet boundary conditions, set to a temperature of $\hat{T} = 2$. The temperature boundary condition on the top and bottom walls have a Neumann boundary condition with zero heat flux. All four walls enforce a no-slip boundary condition on velocity. The melt temperatures are $T_{solidus} = 1.45$ and $T_{liquidus} = 1.55$, corresponding to a mushy region thickness of $\epsilon = 0.1$. At $\hat{t} = 0$, the left wall temperature drops to $\hat{T} = 1.6$, inducing natural convection. Once steady-state circulation is developed at $\hat{t} = 1000$, the left wall

temperature drops further to $\hat{T} = 1$, forming a solid crust that advances from the left wall.

The simulations are run on a 64×64 mesh using a 4^{th}-order accurate $\text{rDG}_{\text{P}_2\text{P}_3}$ spatial discretization scheme. We simulate $\hat{t} = 2000$ units of dimensionless time using a 2^{nd}-order, BDF_2 time discretization scheme. Since our simulations are fully-implicit and we want to step over acoustic and material timescales (dynamics evolve slowly), we pick a time step, $\hat{\Delta}t = 2.0$, corresponding to large CFL and Fourier numbers: $\text{CFL}_{\text{aco}} = 3{,}200$, $\text{CFL}_{\text{mat}} = 30$, $\text{Fo}_\alpha = 200$, $\text{Fo}_\nu = 200$.

In Figure 6.1, the plots on the left column show the temperature and velocity vectors of natural convection at steady-state for increasing Prandtl numbers (from top to bottom), while the plots on the right column show velocity magnitudes of the corresponding cases at the same time. For the top case, $Pr = 0.01$ (mercury), we observe that the natural convection pattern has a ring-shaped, annular structure, as deduced from the velocity magnitude and velocity vectors. This low Prandtl number case corresponds to a thin viscous boundary layer, allowing for easier flow penetration and enhanced natural convection heat transfer near the vicinity of the wall. Our results are in agreement with [66], which show a similar trend of increased heat transfer at the wall for decreasing fluid Prandtl numbers. For the higher Prandtl number case, $Pr = 7$ (water), we observe that the flow velocity begins to diminish near the top and bottom walls, which is due to a thicker viscous boundary layer inhibiting flow penetration. As expected, the intermediate case, $Pr = 0.1$ (steel), resembles a flow structure between the two extreme cases.

After natural convection at steady-state is developed, we drop the left wall temperature from $\hat{T} = 1.6$ to $\hat{T} = 1$, which is below the freezing temperature. As the liquid freezes, the solid crust slowly advances to the right, until steady-state is reached again, as seen in Figure 6.2. As expected, the natural convection patterns developed in Figure 6.1 are displaced to the right from the advancing solid crust from the left wall. The solid-liquid interface is denoted by the two thin black contour lines, where the left contour corresponds to the solidus temperature and the right contour corresponds to the liquidus temperature, and the mushy region exists in-between. In the

top case, $Pr = 0.01$, the solid-liquid interface is curved around the circular convection pattern, while in the bottom case, $Pr = 7$, the solid-liquid interface is less curved and more straight.

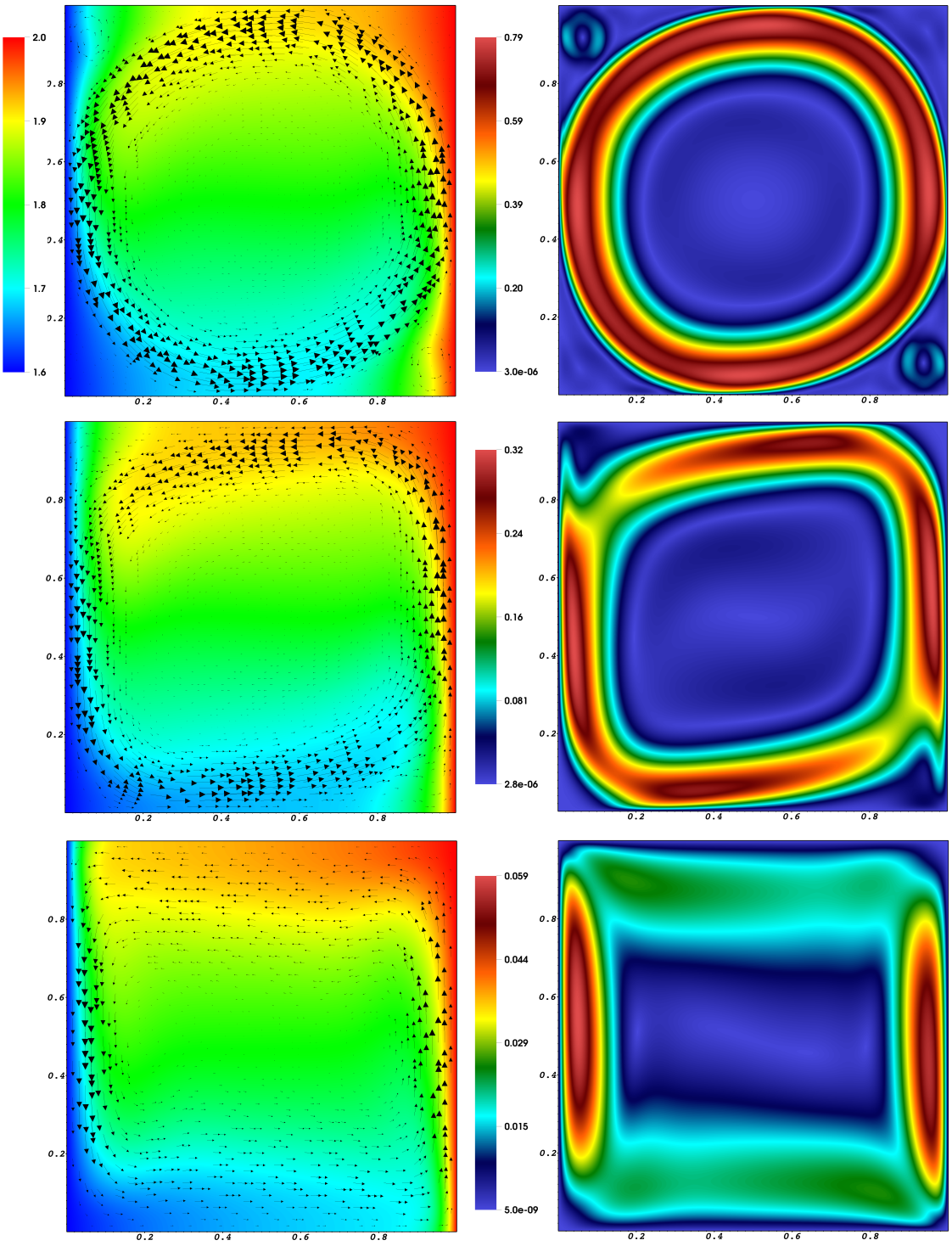


FIGURE 6.1. Natural convection without phase change at steady-state. Left column shows temperature with velocity vectors and right column shows velocity magnitude. The Prandtl number increases from the top row to the bottom row: $Pr = 0.01, 0.1, 7$.

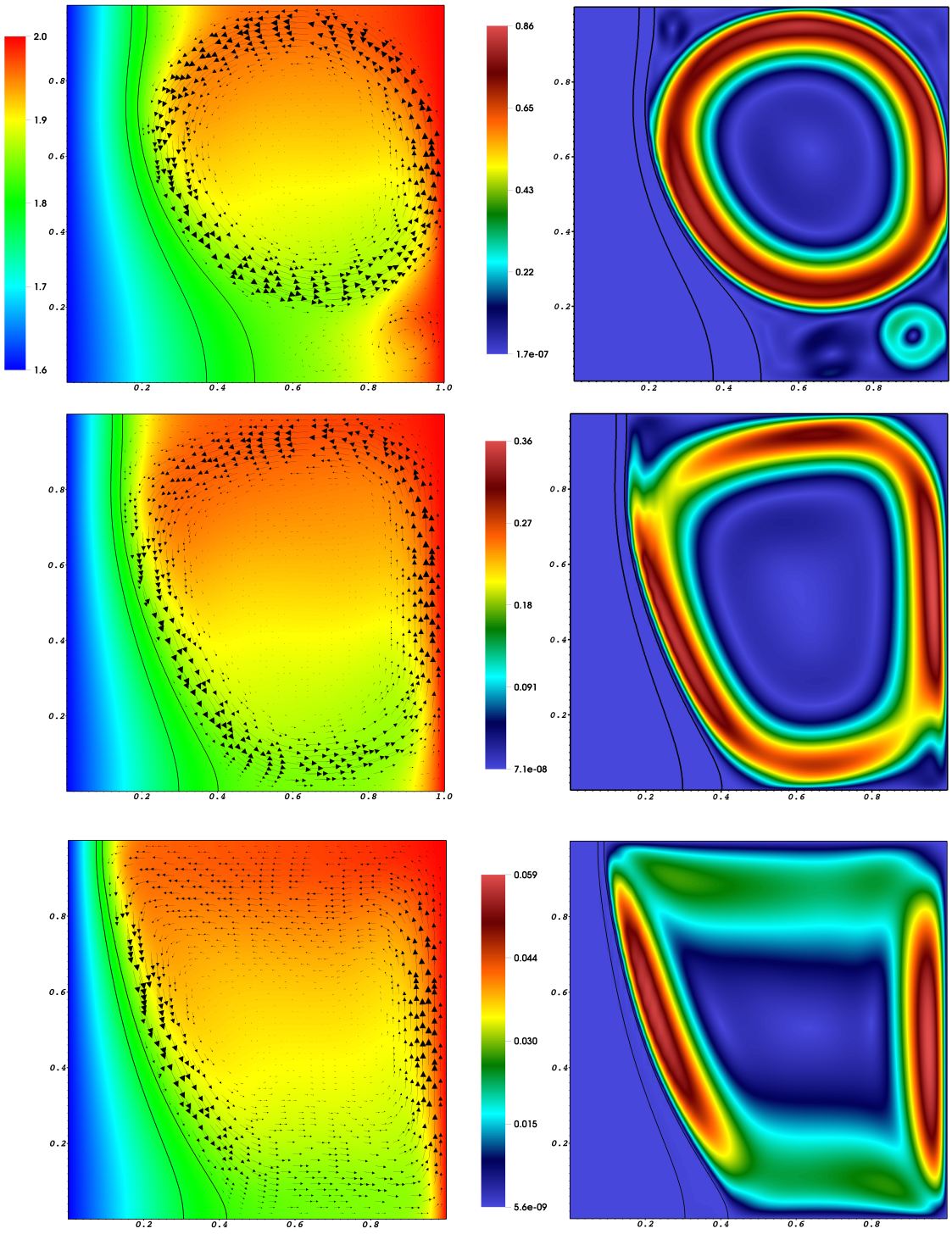


FIGURE 6.2. Natural convection with phase change at steady-state. Left column shows temperature with velocity vectors and right column shows velocity magnitude. The Prandtl number increases from the top row to the bottom row: $Pr = 0.01, 0.1, 7$.

6.2. Stefan Number Effects

In this Section, we vary the Stefan number for two cases: $Ste = 5.0$ and $Ste = 0.5$, while holding the Prandtl number, $Pr = 0.1$, and Rayleigh number, $Ra = 10^6$, constant and using the same problem setup as in the last Section 6.1.

In Figure 6.3, we show the temperature and streamlines for both the low and high Stefan numbers cases, on the top and bottom rows, respectively. The plots on the left column correspond to a snapshot in time when the solid crust just begins to appear and advance to the right. Comparing both Stefan number cases, we observe that the solid-liquid interface for the higher Stefan number case has advanced further to the right, for the same snapshot in time. This behavior is expected if one recalls that the Stefan number, as defined in Eq. (2.32), is the ratio of sensible heat to latent heat. Larger Stefan numbers correspond to relatively smaller latent heats, which means less energy is required for a change of phase. Thus in problems with larger Stefan numbers, for a given temperature gradient, the solid-liquid interface will propagate faster, as verified in the left column of Figure 6.3. The plots on the right column correspond to a snapshot at steady-state. Comparing the two cases, we observe only minor differences in the final position of the solid-liquid interface. These differences are expected since natural convection occurs simultaneously as the interface is propagating, thus changing the final position of the solid-liquid interface. Since the velocity magnitudes are quite small, $|\hat{\mathbf{v}}| < 1$, there are only minor differences in the final interface position. If the velocity magnitudes were very large, however, we would expect larger differences in the final position of the solid-liquid interface due to the non-linear coupling of the governing equations. In contrast, for a pure heat conduction (Stefan) problem, all Stefan number cases would have the same final solid-liquid interface position.

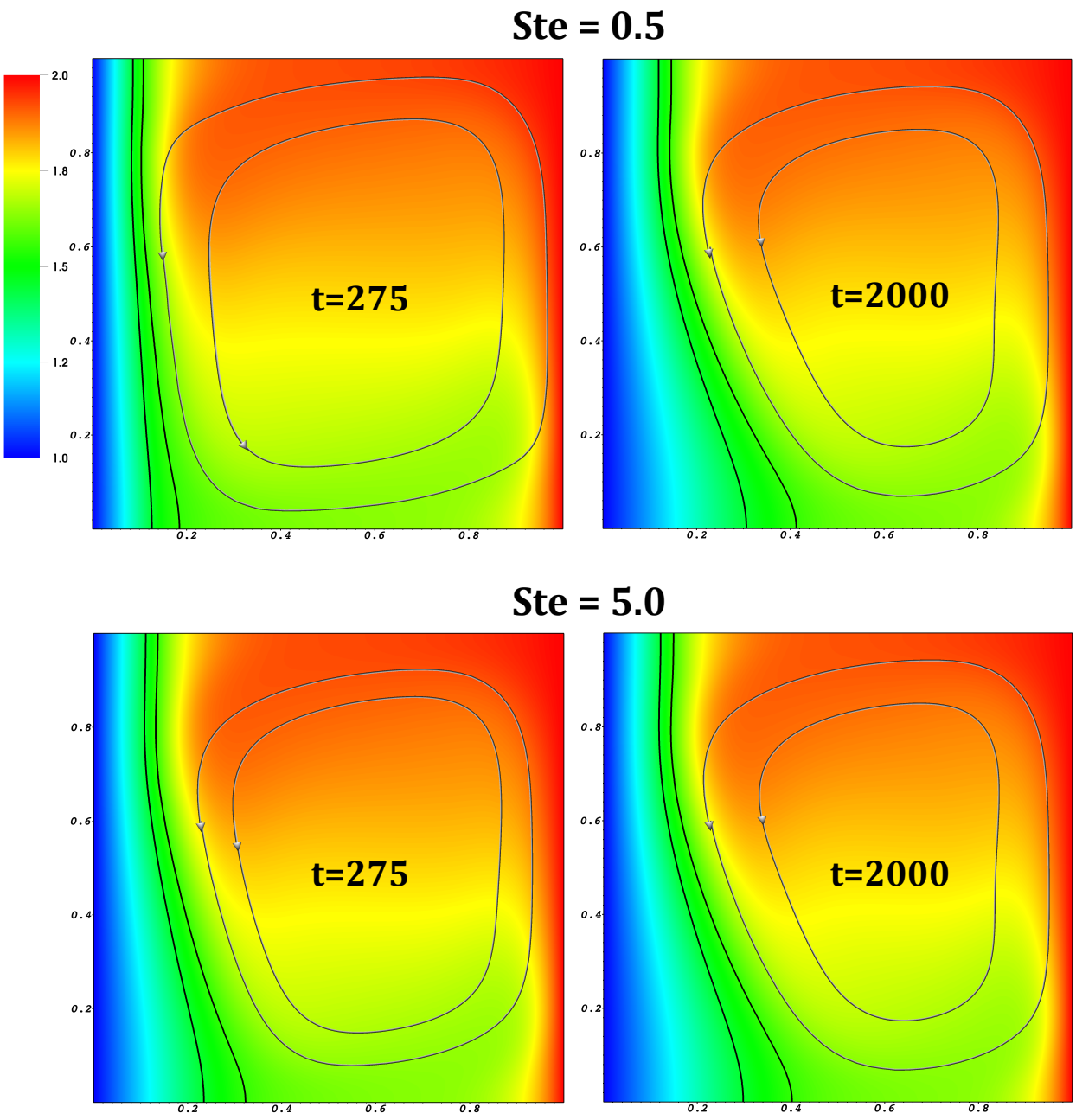


FIGURE 6.3. Temperature and streamlines are shown for two snapshots in time and two different Stefan numbers ($Ste = 0.5$ on the top row and $Ste = 5.0$ on the bottom row). The left column shows snapshots at $\hat{t} = 275$, when the solid crust just appears, while the right column shows snapshots at $\hat{t} = 2000$, corresponding to steady-state.

6.3. Rayleigh Number Effects

In this Section, we vary the Rayleigh number for three cases: $Ra = 10^4, 10^6, 10^8$, while holding the Prandtl number, $Pr = 0.1$, and Stefan number, $Ste = 5$, constant. In this study, we use a more challenging non-uniform, wedge-shaped domain.

In all three cases in Figure 6.4, the material is initially solid steel with a non-dimensional temperature of $\hat{T} = 1$ on the non-uniform mesh. The top and bottom walls have Dirichlet boundary conditions for temperature and both are initially set to a temperature of $\hat{T} = 1$. The bottom wall is quickly ramped to a higher temperature of $\hat{T} = 2.0$, inducing melt convection from the bottom. The left and right walls have Neumann boundary conditions for temperature with a zero heat flux. All four walls enforce a no-slip boundary condition on velocity. The melt temperatures are $T_{solidus} = 1.4$ and $T_{liquidus} = 1.6$, corresponding to a mushy region thickness of $\epsilon = 0.2$. The simulations are run on a 40×80 mesh using a 4^{th}-order accurate $\text{rDG}_{\text{P}_2\text{P}_3}$ spatial discretization scheme. We simulate $\hat{t} = 1000$ units of dimensionless time using a 2^{nd}-order, BDF_2 time discretization scheme. We can again afford to pick a large time step, $\hat{\Delta}t = 0.5$, corresponding to large CFL/Fourier numbers: $\text{CFL}_{\text{aco}} = 10{,}000$, $\text{CFL}_{\text{mat}} = 15$, $\text{Fo}_\alpha = 6$, and $\text{Fo}_\nu = 624$.

As seen in Figure 6.4, we increase the Rayleigh number by two orders of magnitude in each case. For all cases, we observe that heating from the bottom produces the well known Rayleigh-Benard thermal instability, due to thermal buoyancy. For the low Rayleigh number case, 10^4, steady-state develops at $\hat{t} = 1000$ with only two convection cells. As we increase the Rayleigh number to 10^6, more convection cells grow, and steady-state never develops since the eddies become unsteady with time, known as "soft turbulence". In the high Rayleigh number case, $Ra = 10^8$, the onset of "hard turbulence" begins and the simulation is completely unsteady with multiple large and small eddies. In Figure 6.5, a time sequence is shown for this high Rayleigh number case, where we see the formation and subsequent break-up of eddies with time.

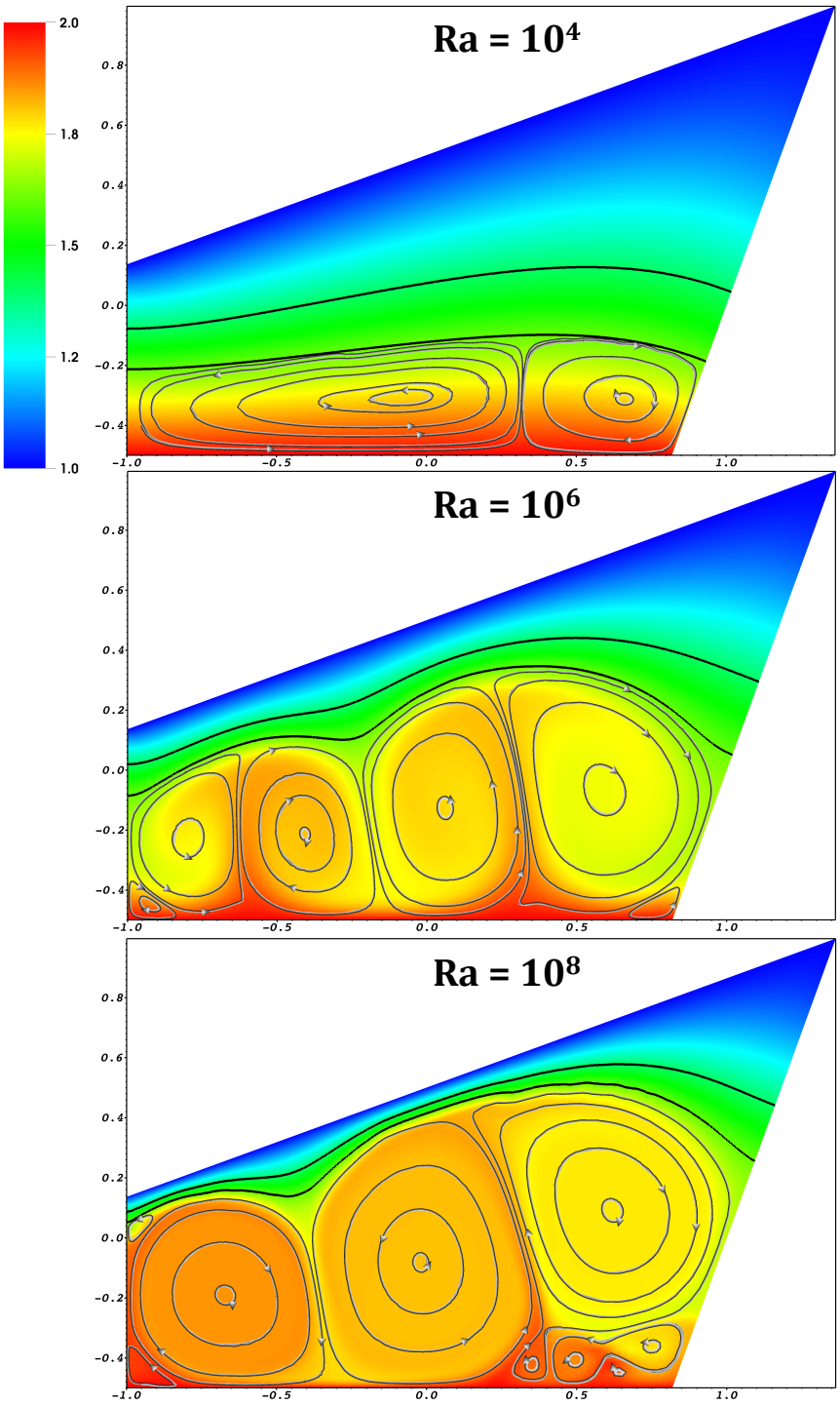


FIGURE 6.4. Temperature plots with streamlines at time, $\hat{t} = 870$, for increasing Rayleigh number.

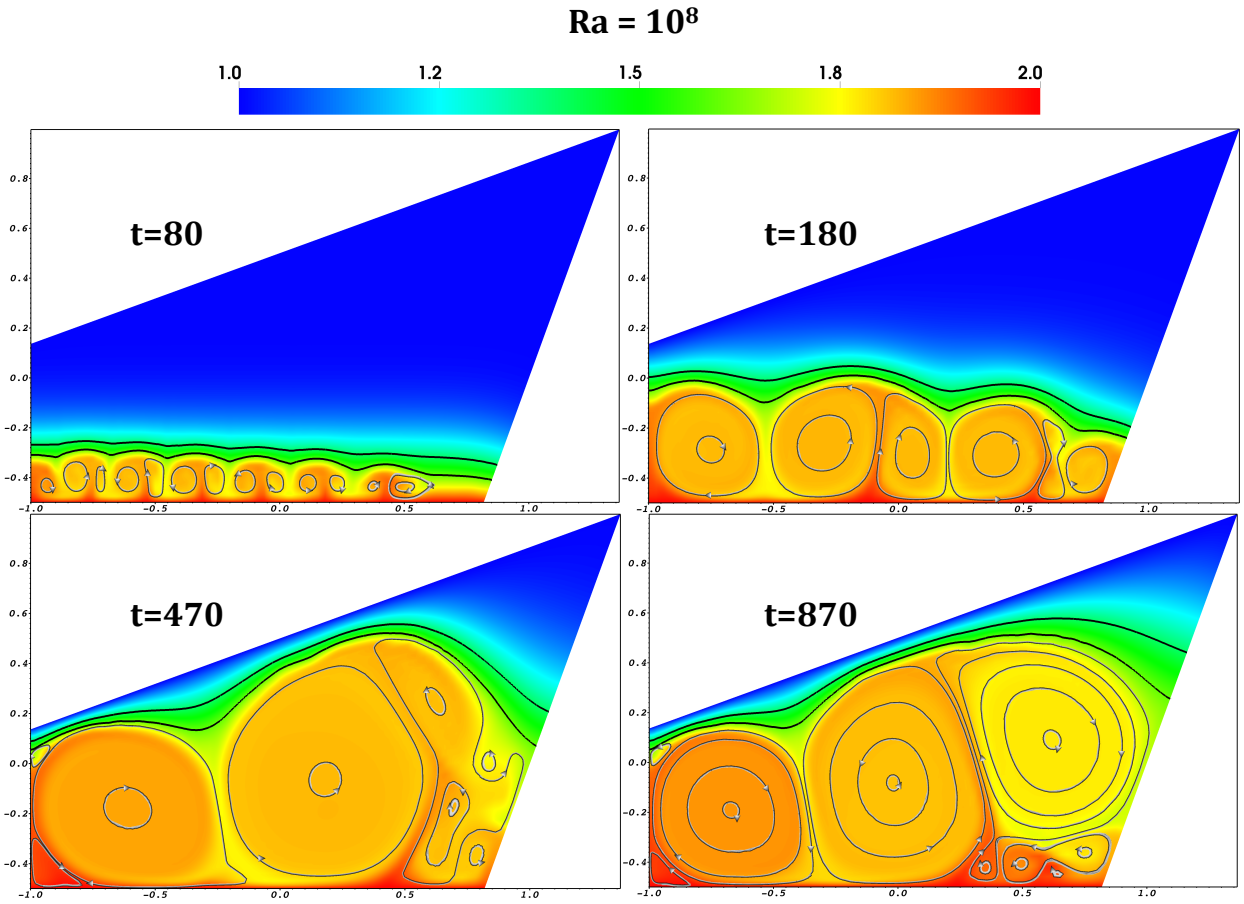


FIGURE 6.5. Temperature plots with streamlines for several time snapshots for the high Rayleigh number case, $Ra = 10^8$.

6.4. Melting/Solidification Configuration Effects

In this Section, we qualitatively verify that our numerical framework correctly produces the expected results in melt convection problems for different melting and solidification configurations. We accomplish this by analyzing the sensitivity of melting and freezing on four different configurations. In all cases, we use the same geometry and parametric values in Section 6.3.

For each of the four different configurations in Figures 6.6 and 6.7, we have two oppositely faced Dirichlet boundary conditions for temperature and two oppositely faced Neumann boundary conditions for temperature. All four walls have no-slip boundary conditions for velocity. In the top left case of Figure 8.10, the top wall is set to $\hat{T} = 1$ while the bottom wall is set to a higher temperature of $\hat{T} = 2$, inducing melt convection from the bottom. In the other configurations, the high temperature wall is set the top, left, and right walls, respectively.

As seen in the top left case of Figure 6.6, heating from the bottom produces the well known Rayleigh-Benard instability, while heating from the top produces a stable configuration as seen in the top right case. Heating from the left and right produce steady natural circulation cells for the given Rayleigh number.

In Figure 6.7, we use the same four configurations except we reverse the boundary conditions and instead cool from the bottom, top, right, and left, respectively. We observe similar flow dynamics as in the melting cases, but an asymmetry exists due to the different initial conditions. For the melting case, we start with an initial (solid) block of steel, where as in the freezing case, we start with an initial (liquid) melt pool. The melting case is computationally very challenging, especially for incompressible flow solvers, since $\nabla \cdot \mathbf{v} = 0$ must be enforced at the beginning of the melting process [46]. Since we use an artificial compressibility approach, as described in Section 2.1.2, we do not observe any such difficulty.

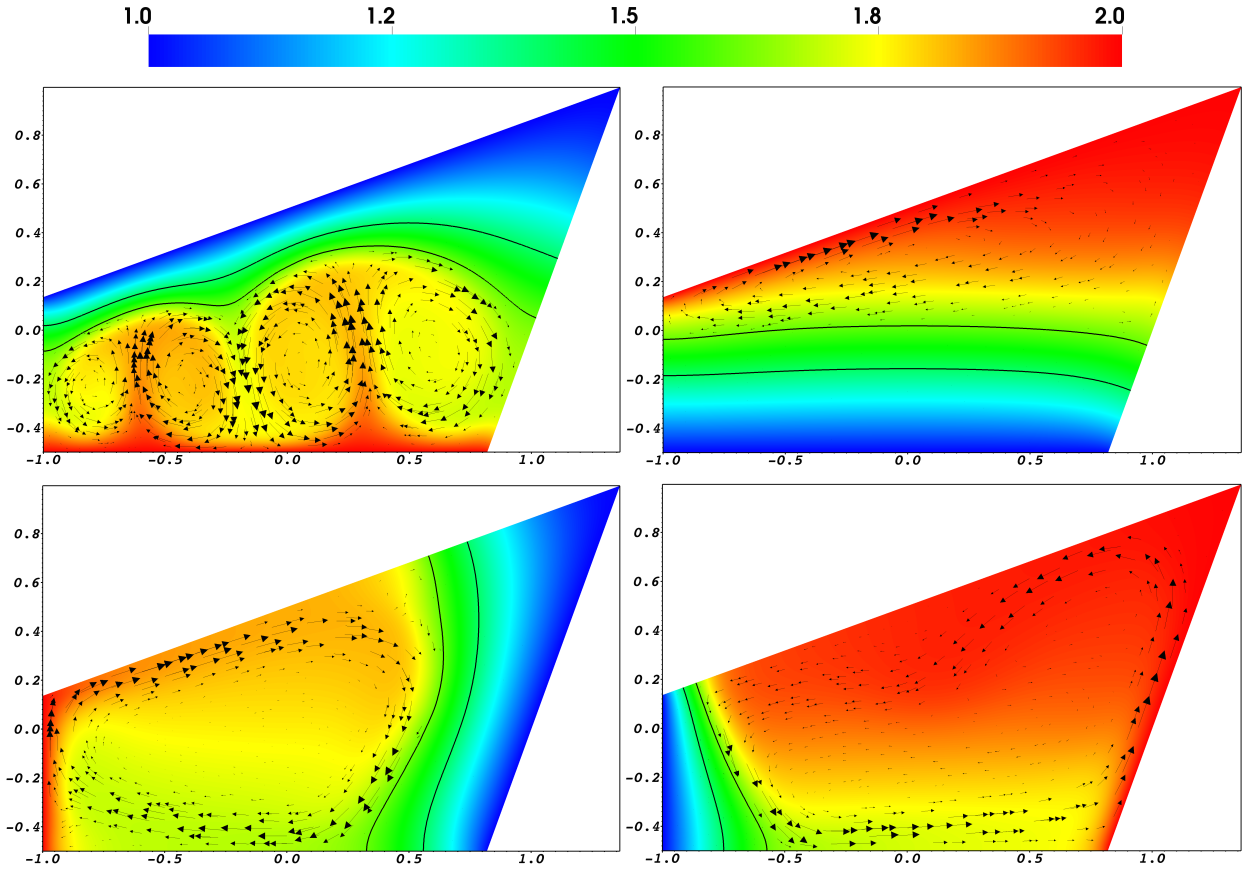


FIGURE 6.6. Temperature plots with velocity vectors for four heating configurations. Heating from the bottom, top, left, and right walls.

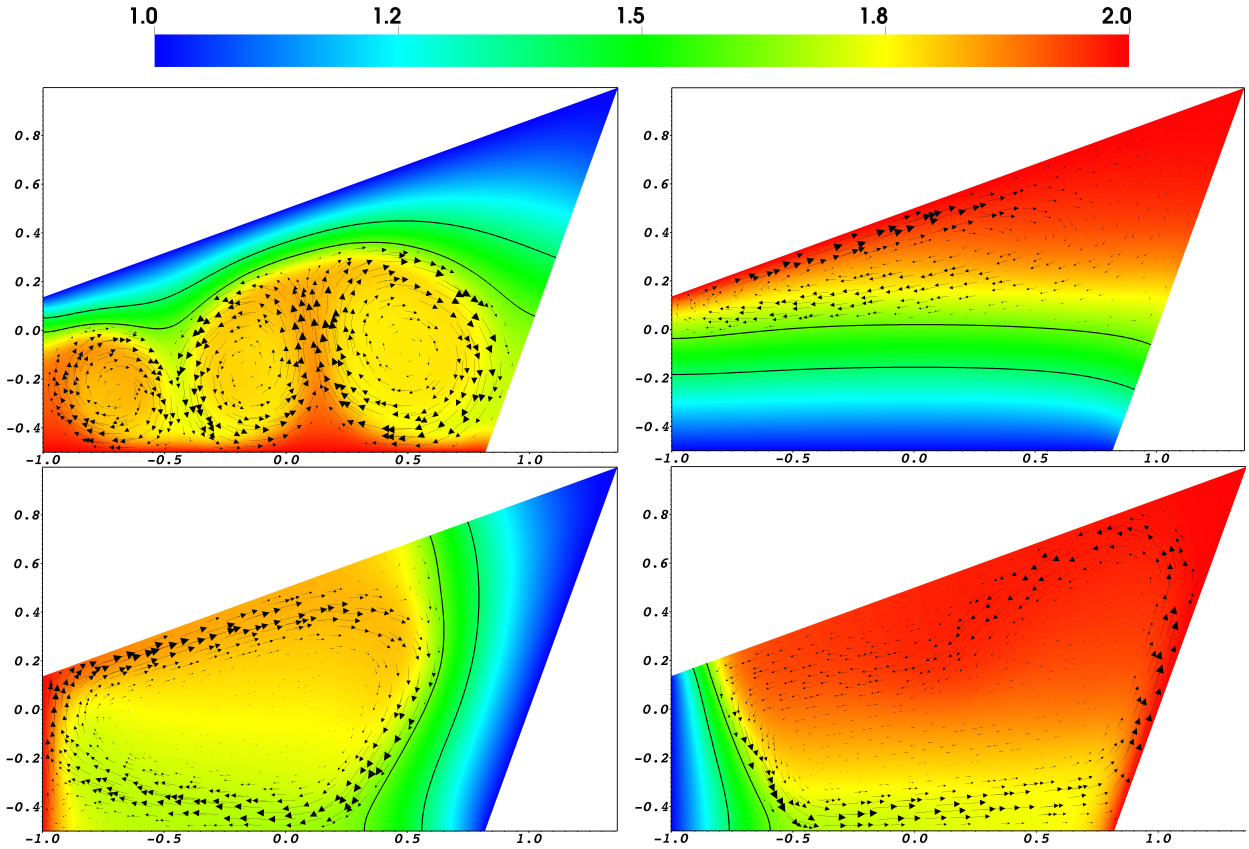


FIGURE 6.7. Temperature plots with velocity vectors for four cooling configurations. Cooling from the bottom, top, left, and right walls.

CHAPTER 7

Laser Processing Results: Moving Laser-Induced Melt Convection

Since we are interested in developing numerical simulations of laser-based additive manufacturing processes, such as SLM, we want to verify that our numerical framework can converge moving laser-induced melt convection. We show converged results for 2D single-track simulations in Section 7.1 and 3D single-track simulations in Section 7.2. The necessity of 3D simulations is demonstrated in Section 7.3, since 2D simulations have artificially elongated melt pools, an unphysical phenomenon.

7.1. 2D Single-Track Simulations

In this numerical example, we simulate a moving two-dimensional laser-induced melt convection problem. As seen in Figures 7.1 and 7.2, the single-track domain is 1 mm $\times$ 0.3 mm. Using the same material parameters defined in Appendix A.3 and the beam model described in Section 2.3, we scan a 200W laser at 2000 mm/s laser across the 1 mm track (centered at the top of the domain). The max penetration depth of the laser is 0.05 mm.

The material is initially an idealized metal with properties similar to those of stainless steel at room temperature, $T_0 = 300$ K, with an energy absorption coefficient of 0.3. The bottom wall has a Dirichlet boundary condition for temperature, fixed at $T = 300\,\mathrm{K}$, while the other walls have Neumann temperature boundary conditions with zero heat flux. The top wall has a slip-boundary condition while the remaining walls enforce a no-slip boundary condition on velocity. The simulation has 40,000 elements in the melt pool (50,000 elements total) using the 2^{nd}-order accurate, $\mathrm{rDG}_{\mathrm{P_0P_1}}$, spatial discretization scheme. Since we want to resolve the time scale associated with the laser's velocity, we pick a time step, $\Delta t = 0.275\mu$s, which is below the dynamic time scale of the laser. These time steps correspond to $\mathrm{CFL}_{\mathrm{aco}} = 1$ and $\mathrm{CFL}_{\mathrm{mat}} = 10^{-5}$, and $\mathrm{Fo}_\alpha = 1.6$ and $\mathrm{Fo}_\nu = 0.2$. Using a 2^{nd}-order, BDF_2 time discretization scheme, we simulate a total of $500\,\mu s$,

which is the time it takes for the laser to scan across the domain.

As the laser moves across the domain and heats the steel beyond its melt temperature, melt pools are formed and subsequently solidified. The temperature field is shown in Figure 7.1 while the velocity field and streamlines are shown in Figure 7.2. Both figures show plots for four snapshots of the simulation in time. Peak temperatures of the steel are observed to be upwards of $5,800$ K. Similar laser parameters were used in a single-track simulation in [36], but the peak temperatures were found to be $3,100$ K, corresponding to the boiling temperature of steel. As mentioned in Section 2.1, since the simulations do not account for the latent heat of vaporization of steel, which is on the order of megajoules per kilogram, radiative effects, or Marangoni convection, the significant over-prediction in temperature is expected. We note that since the Rayleigh number is very low, $\text{Ra} \sim 10^2$, and there are only two stable vortices, the low-order, $\text{rDG}_{\text{P}_0\text{P}_1}$, scheme is sufficient to capture the dynamics in the melt pool.

7.2. 3D Single-Track Simulations

In our final numerical example, we consider a 3D laser-induced melt convection problem on a single-track. The domain is 1 mm $\times$ 0.1 mm $\times$ 0.1 mm as shown in Figure 7.3. We use the same parameters as in the 2D laser-induced melt convection problem in Section 7.1.

As in the last example, the material is initially solid steel at room temperature, $T_0 = 300$ K, with an energy absorption coefficient of 0.3. The bottom wall has a Dirichlet boundary condition for temperature, fixed at $T = 300\,\text{K}$, while the other walls have Neumann temperature boundary conditions with zero heat flux. Both the top wall and the closest facing wall (at $Z = 0.1$ mm) have slip-boundary conditions for velocity, while the remaining walls enforce a no-slip boundary condition. For computational efficiency, we only simulate half the domain since the problem is symmetrical across the Z-axis.

We simulated the 3D laser melting problem for two different cases: one with a coarse mesh with a total of 150,000 elements and the other with a fine mesh, with a total of 700,000 elements. The

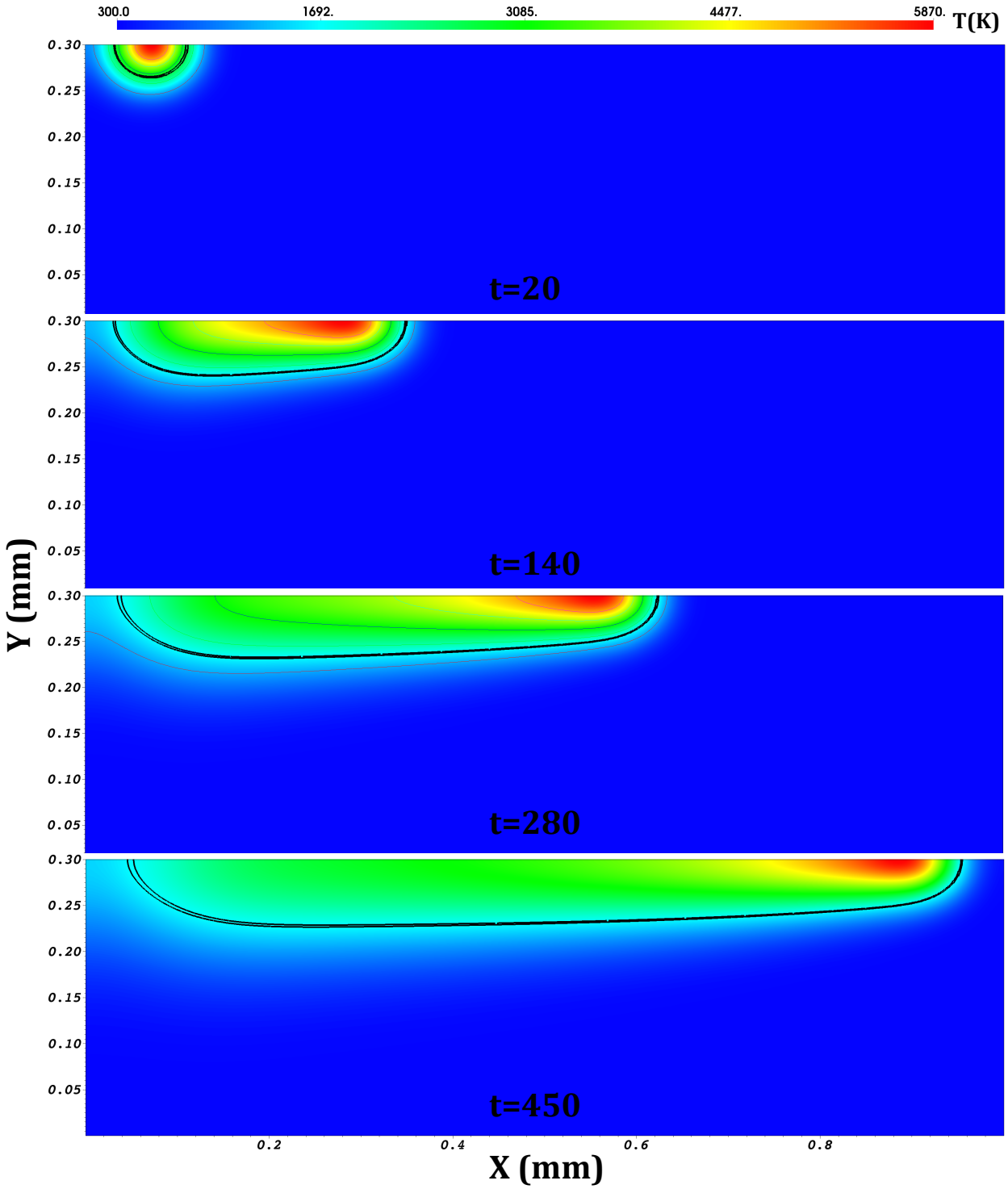


FIGURE 7.1. Temperature plots with temperature contours of a 2D single-track for different time snapshots (in microseconds). Black contour line corresponds to the solid-liquid interface.

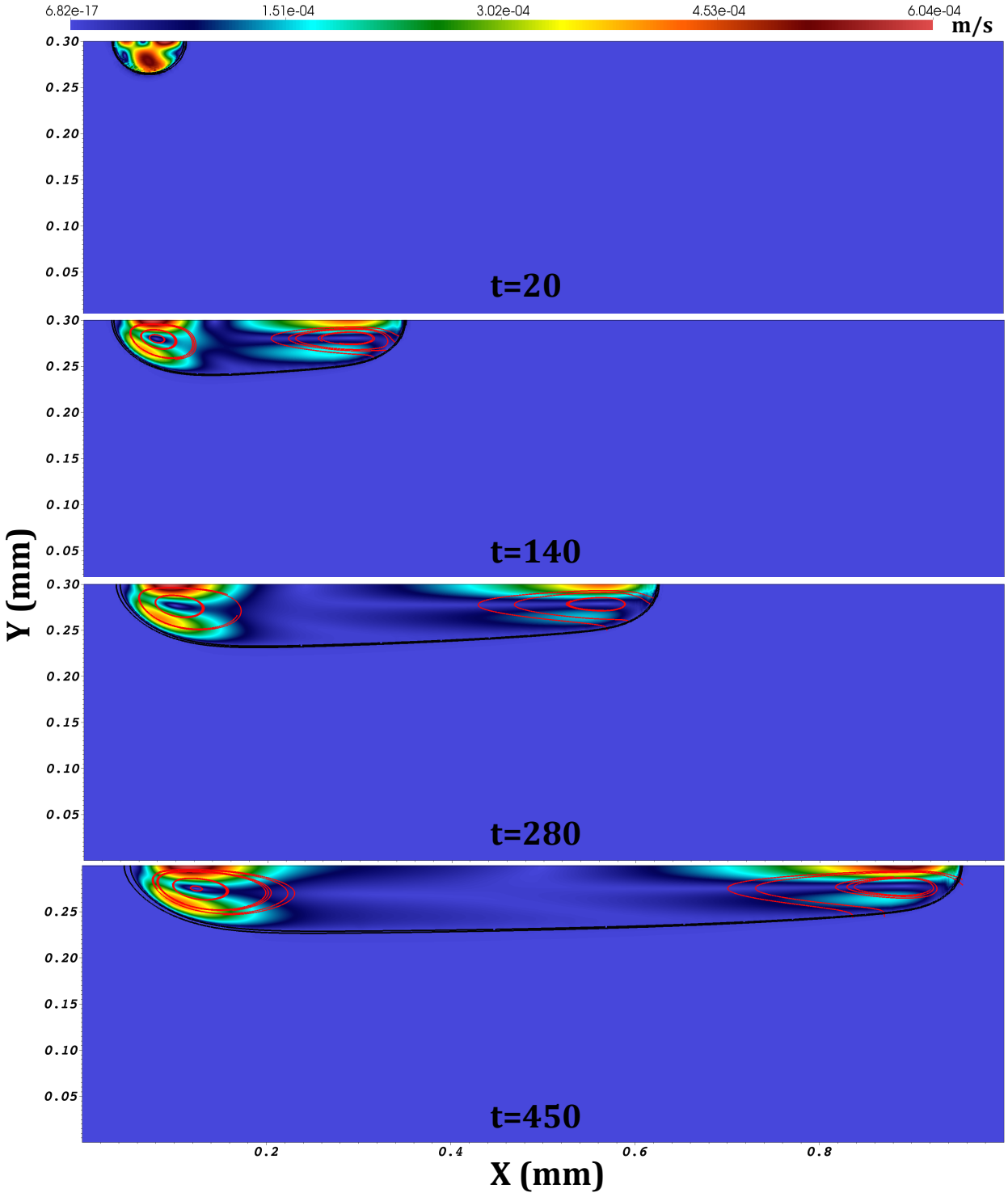


FIGURE 7.2. Velocity magnitude plots with streamlines of a 2D single-track for different time snapshots (in microseconds). Black contour line corresponds to the solid-liquid interface.

coarse mesh case had 100,000 elements in the melt pool and required 6.5 hours on 120 processors using a combined phase change model, $\frac{\mu_s}{\mu_l} = 10,000$ and $C = 500,000$, with a $\Delta t = 2.75\,\mu$s, corresponding to: $\text{CFL}_{\text{aco}} = 3$ and $\text{CFL}_{\text{mat}} = 10^{-6}$, and $\text{Fo}_\alpha = 1.43$ and $\text{Fo}_\nu = 34$. The fine mesh case, as seen in Figure 7.3, had 450,000 elements in the melt pool and required 60 hours to complete on 320 processors using a pure drag force phase change model, $C = 500,000$, with a $\Delta t = 0.25\,\mu$s, corresponding to a $\text{CFL}_{\text{aco}} = 0.5$ and $\text{CFL}_{\text{mat}} = 10^{-7}$, and $\text{Fo}_\alpha = 0.4$ and $\text{Fo}_\nu = 0.05$. In both cases, the simulation are run using the $\text{rDG}_{\text{P}_0\text{P}_1}$ spatial discretization scheme and the BDF_2 time discretization scheme with time steps that resolve the dynamic time scale of the laser.

Similar to the 2D single-track example, both of the 3D cases significantly over predict the peak temperature, as expected. Simulations of the multi-material SLM process will require over a million elements to capture high-resolution dynamics in the melt pool. Scaling up to simulations with more than a million elements, however, requires a more computationally efficient solver and preconditioner, such as developed in Chapter 8.

7.3. Single-Track Length: 2D vs. 3D

To verify the necessity of 3D laser melting simulations (instead of 2D approximations), we compare the size of the melt pool from the 2D and 3D simulations on the (X-Y) plane at $t = 500\mu s$. As seen in Figure 7.4, the melt pool in the 2D simulation is roughly twice as long as the melt pool in the 3D simulation. This difference is expected, since in 3D there is an additional dimension for energy to be transported, while in the 2D case, the energy is confined to the 2D plane, artificially elongating the melt pool. The extra dimension also explains why the peak temperature in the 2D case is slightly higher than the 3D case. Thus, single-track laser melting experiments are an inherently 3D process and require 3D simulations. The melt pool track length shown here qualitatively compares to the simulation results in [37].

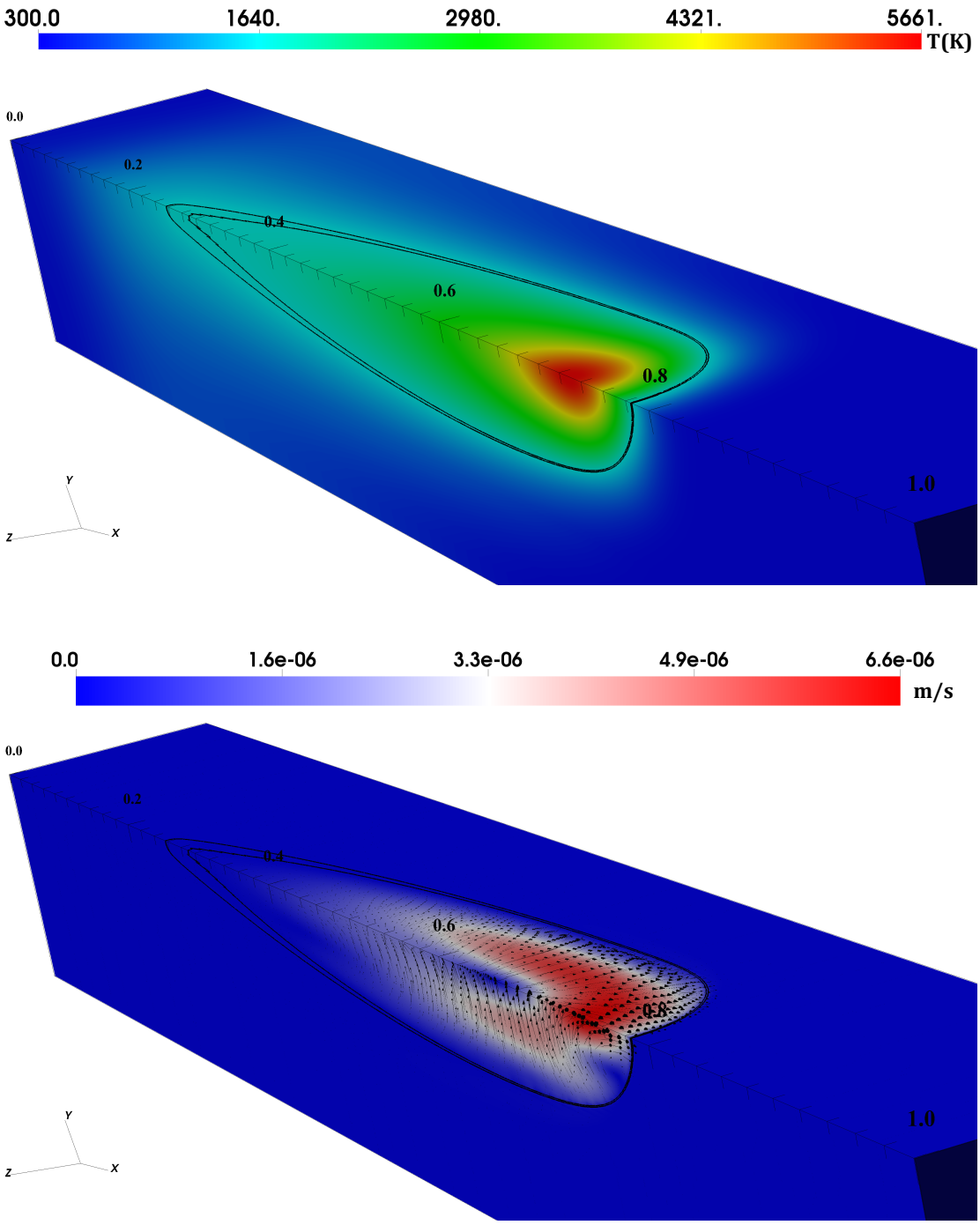


FIGURE 7.3. 3D single-track showing half the domain on a 700,000 element mesh. Temperature is shown on the top and the velocity magnitude with vectors is shown on the bottom, at $t = 325\,\mu s$. The black contour lines represent the solidus and liquidus temperatures, respectively.

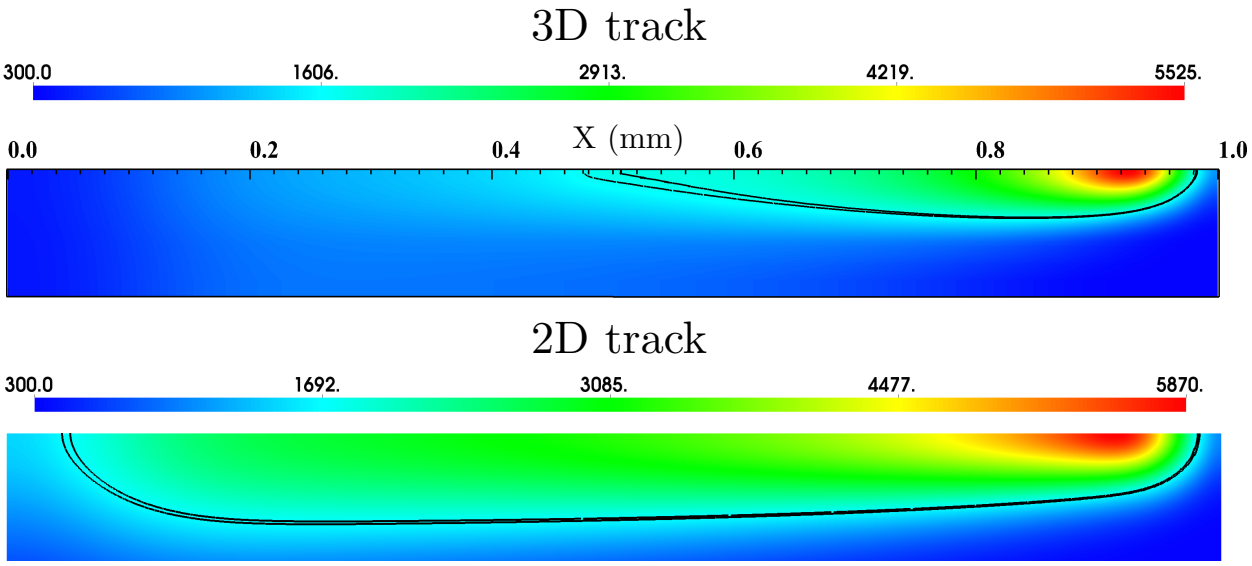


FIGURE 7.4. Length and height of the melt pool on the (X-Y) plane for the 2D and 3D single-track domains using a $\text{rDG}_{\text{P}_0\text{P}_1}$ scheme.

CHAPTER 8

Exploring Preconditioning Strategies

In this Chapter, we review and compare common preconditioning strategies in the literature. Numerical results (strong and weak scaling studies) are shown for three problems of increasing complexity: low-Mach lid-driven cavity flow, low-Mach melt convection on a non-uniform mesh, and 3D laser-induced melt convection on single and multiple melt tracks. The lid-driven cavity flow problem exhibits the numerical challenges associated with stepping over acoustic time-scales in the low-Mach regime. In the melting brick problem, the acoustic time-scale as well as the viscous time-scale associated with solid-liquid phase change are stepped over. Finally, the 3D laser melting problems incorporate the previous challenges with an additional nonlinearity associated with rapid heating due to a moving laser source. We find that the most robust and efficient preconditioner is a multigrid block reduction technique, developed in Section 8.2.4, which uses the the velocity-pressure ($\mathbf{v}P$) and velocity-temperature ($\mathbf{v}T$) Schur complement systems.

8.1. Need for a Scalable and Efficient Preconditioner

The results up until now have been using an LU factorization as a preconditioner for the outer JFNK solver, which is an unscalable approach since the largest problem that can be fit into memory is 700,000 elements in 3D, as seen in Figure 7.3. To simulate larger problems, a scalable and computationally efficient preconditioner needs to be developed with the ability to converge solutions resulting from highly-ill conditioned systems at high CFL/Fourier numbers.

Performance of all preconditioners are compared by analyzing outer GMRES iteration counts and CPU-times. All calculations requiring over 500 processors were run on the Jade supercomputer at the Lawrence Livermore National Laboratory. The machine has 1,302 compute nodes (36 processors per node) with 2.1 GHz Intel Xeon E5-2695 processors and 128 GB of memory per node. All

other runs were performed on the RZMerl supercomputer with 154 compute nodes (16 processors per node) using 2.6 GHz Sandy Bridge processors with 32 GB of memory per node.

8.2. Survey of Preconditioning Techniques

In this section, we survey common preconditioning techniques in the literature for JFNK solvers. An extensive survey of JFNK methods and preconditioners can be found in [40].

8.2.1. One-Level Additive Schwarz Preconditioner. We consider an element-block additive Schwarz method as a preconditioner for the outer JFNK solver. In this study, the degrees of freedom in each element-block consist of all three cell-averaged fields: pressure, velocity, and temperature. In this one-level approach, we couple all degrees of freedom within an element block to all degrees of freedom within neighboring elements in the stencil, using an element-block Schur complement matrix. To form the Schur complement matrix, the element block matrix (a 4×4 system in 2D) is exactly inverted and a diagonal approximation is used for the overlapping degrees of freedom.

8.2.2. Monolithic Algebraic Multigrid Preconditioner. We apply Algebraic Multigrid (AMG) on the fully-coupled monolithic system as a preconditioner to the outer GMRES-Newton solver. We use HYPRE's BoomerAMG, a parallel implementation of AMG developed at the Lawrence Livermore National Laboratory [27]. In this study, we use the default settings for BoomerAMG, which utilizes a symmetric SOR smoother on a single V-Cycle with Falgout coarsening.

8.2.3. Physics-Block Gauss-Seidel. This preconditioner uses the lower triangular portion of the approximate Jacobian matrix, ordered by physics blocks,

$$\begin{bmatrix} \mathbf{M}_{PP} & 0 & 0 \\ \mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{\mathbf{vv}} & 0 \\ \mathbf{M}_{TP} & \mathbf{M}_{T\mathbf{v}} & \mathbf{M}_{TT} \end{bmatrix} \begin{bmatrix} \mathbf{x}_P \\ \mathbf{x}_{\mathbf{v}} \\ \mathbf{x}_T \end{bmatrix} = \begin{bmatrix} \mathbf{b}_P \\ \mathbf{b}_{\mathbf{v}} \\ \mathbf{b}_T \end{bmatrix}. \tag{8.1}$$

The outgoing vector is then solved for with one iteration of a block Gauss-Seidel method,

$$\mathbf{x}_P = \mathbf{M}_{PP}^{-1}\mathbf{b}_P \tag{8.2}$$

$$\mathbf{x}_V = \mathbf{M}_{\mathbf{vv}}^{-1}(\mathbf{b}_\mathbf{v} - \mathbf{M}_{\mathbf{v}P}\mathbf{x}_P) \tag{8.3}$$

$$\mathbf{x}_T = \mathbf{M}_{TT}^{-1}(\mathbf{b}_T - \mathbf{M}_{TP}\mathbf{x}_P - \mathbf{M}_{T\mathbf{v}}\mathbf{x}_\mathbf{v}). \tag{8.4}$$

To approximate the action of the inverse for the three block solves in Eq.'s (8.2-8.4), we use BoomerAMG as a preconditioner to FGMRES. A similar physics-block Gauss-Seidel strategy was developed and implemented in [65, 70].

8.2.4. Physics-Block Schur Complement (vP-vT). Since our *2-parameter* EOS has no pressure-temperature dependance, the pressure and temperature coupling is weak. Therefore, the $\mathbf{M}_{PT}$ and $\mathbf{M}_{TP}$ blocks can be neglected without a loss in robustness. Thus, the 3×3 block system reduces to

$$\begin{bmatrix} \mathbf{M}_{\mathbf{vv}} & \mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{\mathbf{v}T} \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{M}_{PP} & 0 \\ \mathbf{M}_{TV} & 0 & \mathbf{M}_{TT} \end{bmatrix} \begin{bmatrix} \mathbf{x}_\mathbf{v} \\ \mathbf{x}_P \\ \mathbf{x}_T \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\mathbf{v} \\ \mathbf{b}_P \\ \mathbf{b}_T \end{bmatrix}. \tag{8.5}$$

The block LU decomposition of this 3×3 system is

$$\begin{bmatrix} \mathbf{M}_{\mathbf{vv}} & \mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{\mathbf{v}T} \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{M}_{PP} & 0 \\ \mathbf{M}_{TV} & 0 & \mathbf{M}_{TT} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ \mathbf{M}_{P\mathbf{v}}\mathbf{M}_{\mathbf{vv}}^{-1} & \mathbf{I} & 0 \\ \mathbf{M}_{T\mathbf{v}}\mathbf{M}_{\mathbf{vv}}^{-1} & -\mathbf{M}_{T\mathbf{v}}\mathbf{M}_{\mathbf{vv}}^{-1}\mathbf{M}_{\mathbf{v}P}\mathbf{S}_{\mathbf{v}P}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{M}_{\mathbf{vv}} & \mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{\mathbf{v}T} \\ 0 & \mathbf{S}_{\mathbf{v}P} & -\mathbf{M}_{P\mathbf{v}}\mathbf{M}_{\mathbf{vv}}^{-1}\mathbf{M}_{\mathbf{v}T} \\ 0 & 0 & \mathbf{Z} \end{bmatrix}. \tag{8.6}$$

where $\mathbf{S}_{\mathbf{v}P}$ is the velocity-pressure Schur complement

$$\mathbf{S}_{\mathbf{v}P} = \mathbf{M}_{PP} - \mathbf{M}_{PV}\mathbf{M}_{\mathbf{vv}}^{-1}\mathbf{M}_{\mathbf{v}P}, \tag{8.7}$$

and $\mathbf{Z}$ is a nested Schur complement matrix, coupling velocity, pressure, and temperature

$$\mathbf{Z} = \mathbf{M}_{TT} - \mathbf{M}_{T\mathbf{v}}\mathbf{M}_{\mathbf{vv}}^{-1}(\mathbf{I} + \mathbf{M}_{\mathbf{v}P}\mathbf{S}_{\mathbf{v}P}^{-1}\mathbf{M}_{P\mathbf{v}}\mathbf{M}_{\mathbf{vv}}^{-1})\mathbf{M}_{\mathbf{v}T}. \tag{8.8}$$

As noted in [18], solving the fully-coupled 3×3 nested Schur complement system is formidable. Instead, we reduce the 3×3 system to a sequence of reduced 2×2 block systems, the velocity-pressure ($\mathbf{v}P$) and velocity-temperature ($\mathbf{v}T$) systems. The block LU decomposition of the 2×2 $\mathbf{v}P$ system is given by

$$\begin{bmatrix} \mathbf{M_{vv}} & \mathbf{M}_{\mathbf{v}P} \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{M}_{PP} \end{bmatrix} = \begin{bmatrix} \mathbf{M_{vv}} & 0 \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{S}_{\mathbf{v}P} \end{bmatrix} \begin{bmatrix} I & \mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}P} \\ 0 & I \end{bmatrix}. \tag{8.9}$$

Similarly, the block LU decomposition of the 2×2 $\mathbf{v}T$ system is given by,

$$\begin{bmatrix} \mathbf{M_{vv}} & \mathbf{M}_{\mathbf{v}T} \\ \mathbf{M}_{T\mathbf{v}} & \mathbf{M}_{TT} \end{bmatrix} = \begin{bmatrix} \mathbf{M_{vv}} & 0 \\ \mathbf{M}_{T\mathbf{v}} & \mathbf{S}_{\mathbf{v}T} \end{bmatrix} \begin{bmatrix} I & \mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}T} \\ 0 & I \end{bmatrix}, \tag{8.10}$$

where the $\mathbf{v}P$ and $\mathbf{v}T$ Schur complement matrices are defined as

$$\mathbf{S}_{\mathbf{v}P} = \mathbf{M}_{PP} - \mathbf{M}_{PV}\mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}P} \tag{8.11}$$

$$\mathbf{S}_{\mathbf{v}T} = \mathbf{M}_{TT} - \mathbf{M}_{TV}\mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}T}. \tag{8.12}$$

Using these physics-block factorizations, the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioning matrix is

(8.13)

$$M_{\mathbf{v}P-\mathbf{v}T} = \begin{bmatrix} \mathbf{M_{vv}} & \mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{\mathbf{v}T} \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{M}_{PP} & \mathbf{M}_{P\mathbf{v}}\mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}T} \\ \mathbf{M}_{T\mathbf{v}} & \mathbf{M}_{T\mathbf{v}}\mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}P} & \mathbf{M}_{TT} \end{bmatrix} = \begin{bmatrix} \mathbf{M_{vv}} & 0 & 0 \\ \mathbf{M}_{P\mathbf{v}} & \mathbf{S}_{\mathbf{v}P} & 0 \\ \mathbf{M}_{T\mathbf{v}} & 0 & \mathbf{S}_{\mathbf{v}T} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}P} & \mathbf{M_{vv}^{-1}}\mathbf{M}_{\mathbf{v}T} \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix}.$$

With the above preconditioning matrix, the solution procedure proceeds in two steps. First, the intermediate velocity, pressure, and temperature solutions are solved using forward substitution

$$\mathbf{x}_{\mathbf{v}}^{*} = \mathbf{M_{vv}^{-1}}\mathbf{b_{vv}} \tag{8.14}$$

$$\mathbf{x}_P = \mathbf{S}_{\mathbf{v}P}^{-1}(\mathbf{b}_P - \mathbf{M}_{P\mathbf{v}}\mathbf{x}_{\mathbf{v}}^{*}) \tag{8.15}$$

$$\mathbf{x}_T = \mathbf{S}_{\mathbf{v}T}^{-1}(\mathbf{b}_T - \mathbf{M}_{T\mathbf{v}}\mathbf{x}_{\mathbf{v}}^{*}). \tag{8.16}$$

Finally, the velocity solution is corrected using backward substitution

$$\mathbf{x_v} = \mathbf{M_{vv}^{-1}}(\mathbf{b_v} - \mathbf{M_{v}}_P\mathbf{x}_P - \mathbf{M_{v}}_T\mathbf{x}_T). \tag{8.17}$$

Note that this procedure is analogous to Chorin's projection algorithm and operator-splitting methods, such as SIMPLE, in incompressible flow solvers [16, 71].

Eq.'s (8.15) and (8.16) require approximating the action of the inverse of a Schur complement system. Explicitly forming the exact Schur complement matrices is prohibitively expensive, since the inverse of $\mathbf{M_{vv}}$ would be required. We implement three different strategies for solving the Schur complement system.

(1) Form an approximate Schur complement matrix by using a diagonal approximation, $\mathbf{D_{vv}} = \text{diag}(\mathbf{M_{vv}})$. In this case, $\mathbf{S}_{\mathbf{v}P}$ and $\mathbf{S}_{\mathbf{v}T}$ are now replaced by $\tilde{\mathbf{S}}_{\mathbf{v}P}$ and $\tilde{\mathbf{S}}_{\mathbf{v}T}$, respectively

$$\tilde{\mathbf{S}}_{\mathbf{v}P} = \mathbf{M}_{PP} - \mathbf{M}_{P\mathbf{v}}\mathbf{D}_{\mathbf{vv}}^{-1}\mathbf{M}_{\mathbf{v}P} \tag{8.18}$$

$$\tilde{\mathbf{S}}_{\mathbf{v}T} = \mathbf{M}_{TT} - \mathbf{M}_{T\mathbf{v}}\mathbf{D}_{\mathbf{vv}}^{-1}\mathbf{M}_{\mathbf{v}T}. \tag{8.19}$$

$\mathbf{D_{vv}}$ is a good approximation of $\mathbf{M_{vv}}$ when the off-diagonal contributions are small relative to the diagonal, such as the case for small time steps or small viscosity ratio's between the solid and liquid phase.

(2) Solve the exact Schur complement systems in a matrix-free fashion, without explicitly forming the Schur complement matrices. In this strategy, $M_{\mathbf{VV}}^{-1}$ is approximated with a single V-cycle.

(3) (Most robust case) Solve the exact matrix-free, $\mathbf{S}_{\mathbf{v}P}$ and $\mathbf{S}_{\mathbf{v}T}$ Schur complement systems, but now preconditioned with the explicitly formed Schur complement matrices, $\tilde{\mathbf{S}}_{\mathbf{v}P}$ and $\tilde{\mathbf{S}}_{\mathbf{v}T}$, as defined in Eq.'s (8.18) and (8.19).

These three strategies are compared in Section 8.4.2. To approximate the action of the inverse of the block solves in Eq.'s (8.14-8.17), we use AMG preconditioned GMRES to solve each block system to a relative linear tolerance of 10^{-1}. Various combinations of preconditioners and solvers (SOR, AMG, GMRES), as well as linear tolerances are explored in Section 8.3.4.

8.3. Low-Mach Lid-Driven Cavity Flow Results

The velocity magnitude with streamlines and temperature fields for the 2D lid-driven cavity flow problem is shown in Figures 8.1 and 8.2, respectively. For performance analysis, the initial temperature of the entire domain is set to a constant temperature $T = 1$ and all four walls have isothermal boundary conditions for temperature at $T = 1$. The top wall is moving to the right with a prescribed velocity, $v_x = 1$, and all other walls enforce a no-velocity condition. For all of the studies, we started at $t \approx 2$ and counted the number of outer FGMRES iterations and CPU time per time step, averaged over 50 time steps. A square domain with a 512×512 mesh resolution was used, except for the strong and weak scaling results. The Mach number varied in the range of 10^{-2} to 10^{-5}. The $\mathrm{rDG}_{\mathrm{P_0P_1}}$ scheme and the BDF_2 integrator were used for performance analysis of different preconditioning techniques.

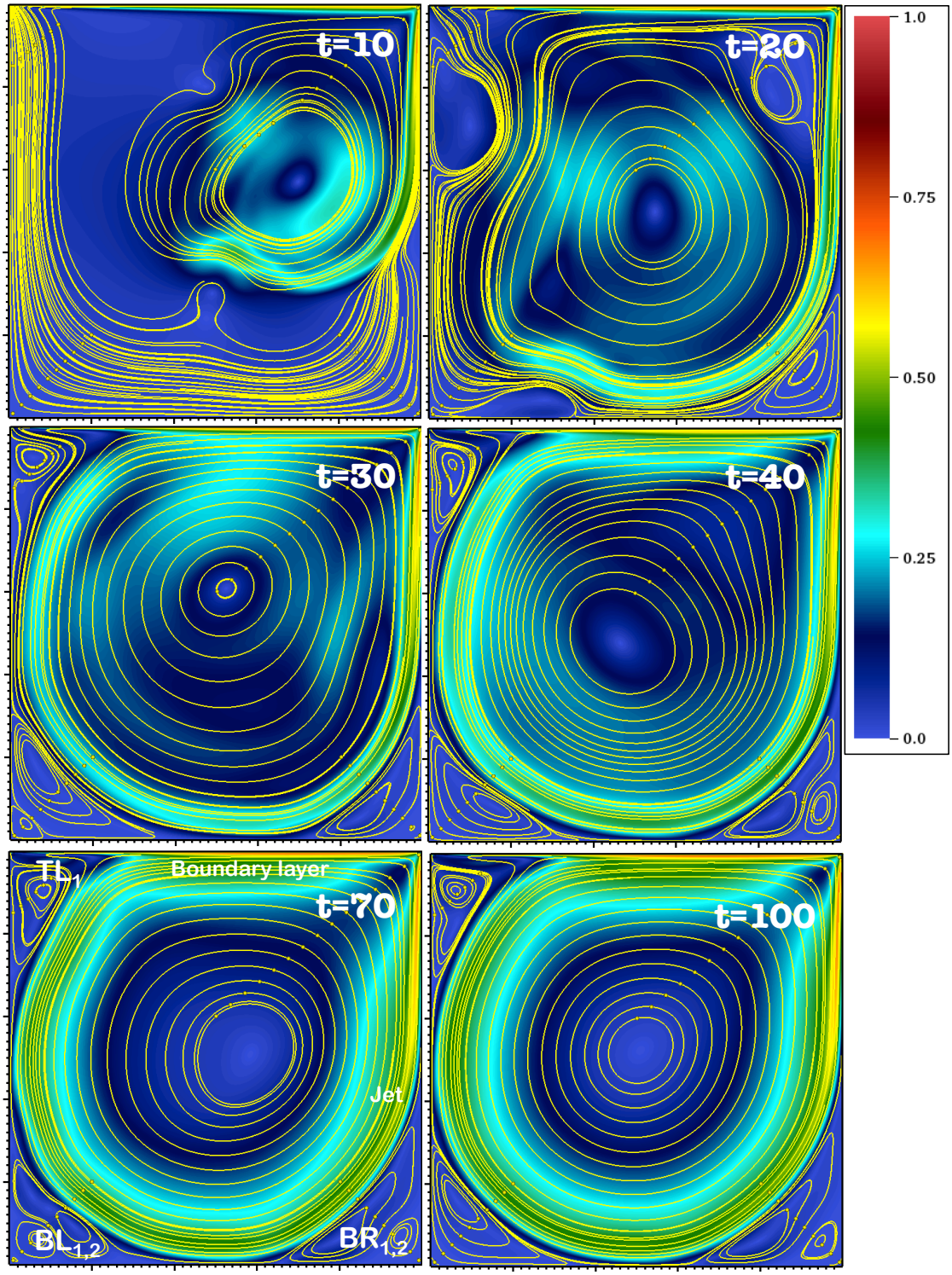


FIGURE 8.1. Dynamics of the velocity magnitude and streamlines for $\mathrm{rDG}_{\mathrm{P_2P_3}}$, mesh resolution 128×128, ESDIRK_5 , $\Delta t = 1$, $\mathrm{Re} = 10^4$.

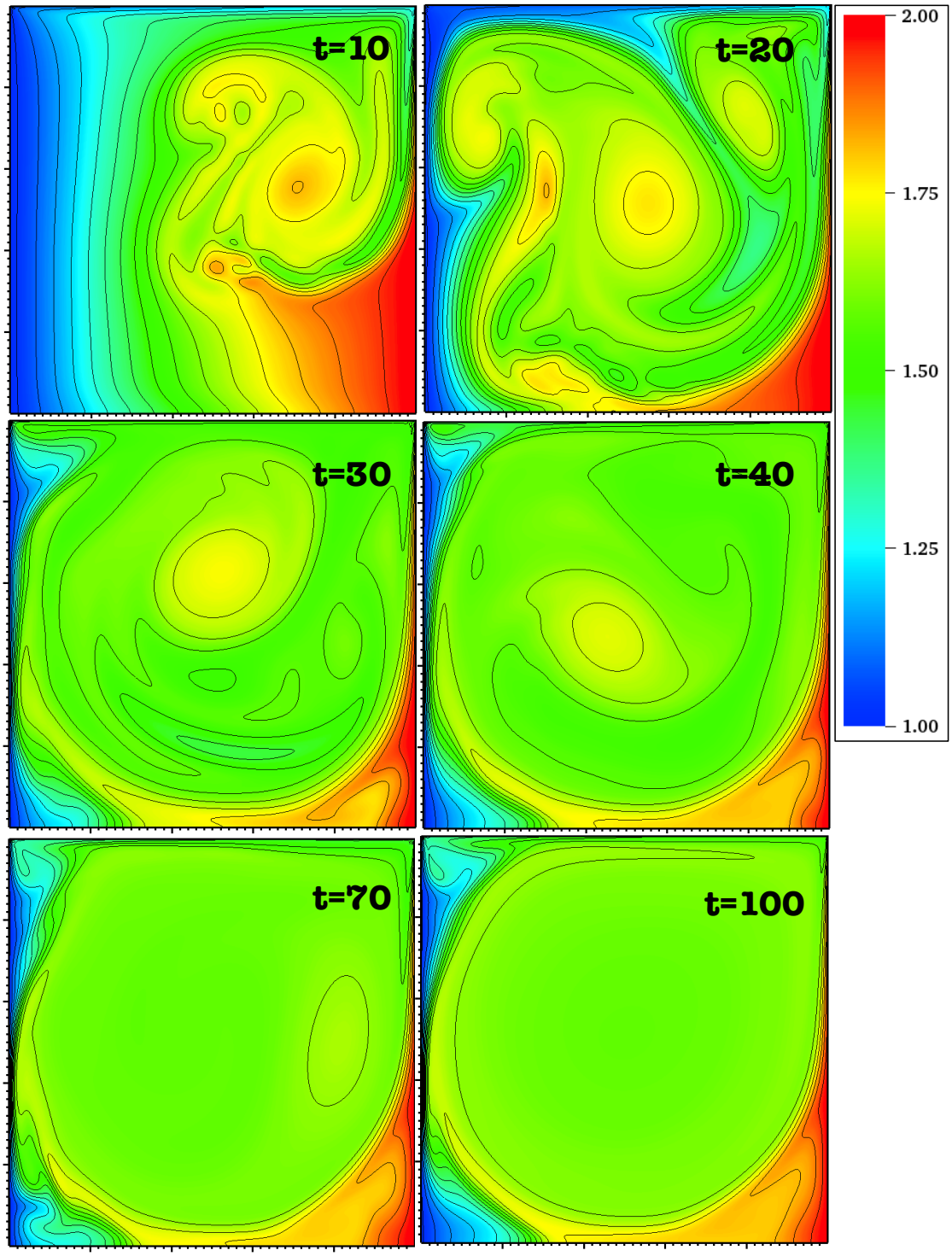


FIGURE 8.2. Dynamics of the temperature field for $\mathrm{rDG}_{\mathrm{P_2P_3}}$, mesh resolution 128×128, ESDIRK_5, $\Delta t = 1$, $\mathrm{Re} = 10^4$. Left and right walls are initially held at temperatures of $T = 1$ and $T = 2$, respectively, with a linear temperature gradient across the domain.

8.3.1. Eigenvalues for a Lid-Driven Cavity Flow Matrix. To illustrate the complexity of solving the Jacobian system in Eq. (3.5), the eigenvalues of an unpreconditioned Jacobian matrix is plotted for the lid-driven cavity flow problem in Figure 8.3. In this system, the condition number is on the order of 10 million, corresponding to an acoustic CFL number of 10,000 and an advection CFL number of 30. An effective preconditioner that clusters the eigenvalues is thus necessary for rapid convergence.

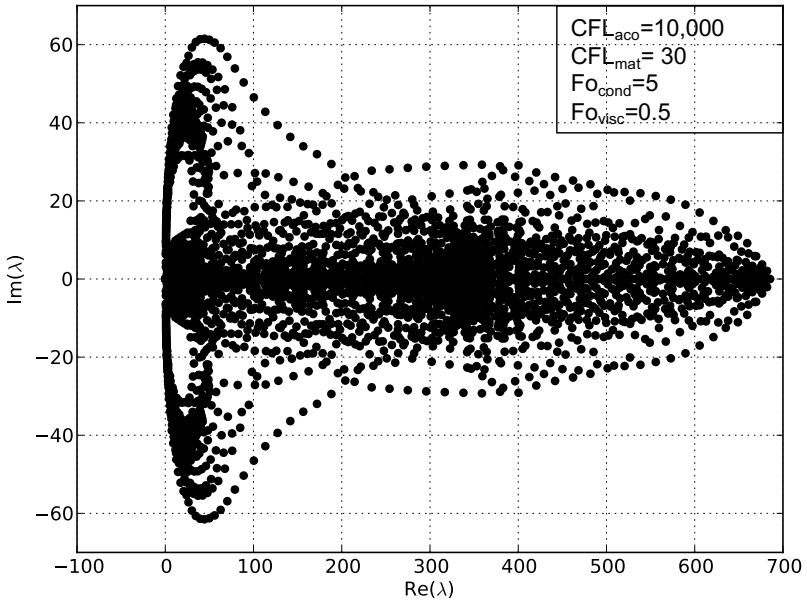


FIGURE 8.3. Eigenvalues of an unpreconditioned Jacobian matrix for the low-Mach lid-driven cavity flow problem. The condition number is large, $\kappa = 10^7$.

8.3.2. Weak Scaling Study. In Figure 8.4, we conduct a (fixed CFL number) weak scaling study to analyze the algorithmic scalability of the different preconditioners for the low-Mach lid-driven cavity flow problem. In all of the cases, the number of degrees of freedom per processor is fixed, to ensure a constant workload. The mesh varied from 362×362 up to 2048×2048, while the number of processors varied from 16 to 512. The acoustic CFL number was fixed at 15 and all other time-scales were resolved. We note that as the number of degrees of freedom (DoF) increases in each case, the mesh width decreases, which increases the Fourier number.

As seen in Figure 8.4, all of the preconditioners scale well in the weak sense. The one-level

additive Schwarz preconditioner converges with the most iterations and CPU time for the given CFL number. The physics-block Gauss-Seidel preconditioner fares better, but the LU factorization and the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioners converge in the least number of iterations[1]. The LU factorization is competitive up to 10^5 DoFs, but performance significantly degrades above 10^6. Furthermore, the LU factorization preconditioner cannot load problems larger than 10^6 DoFs in memory. As a result, the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner had the best performance and is algorithmically scalable.

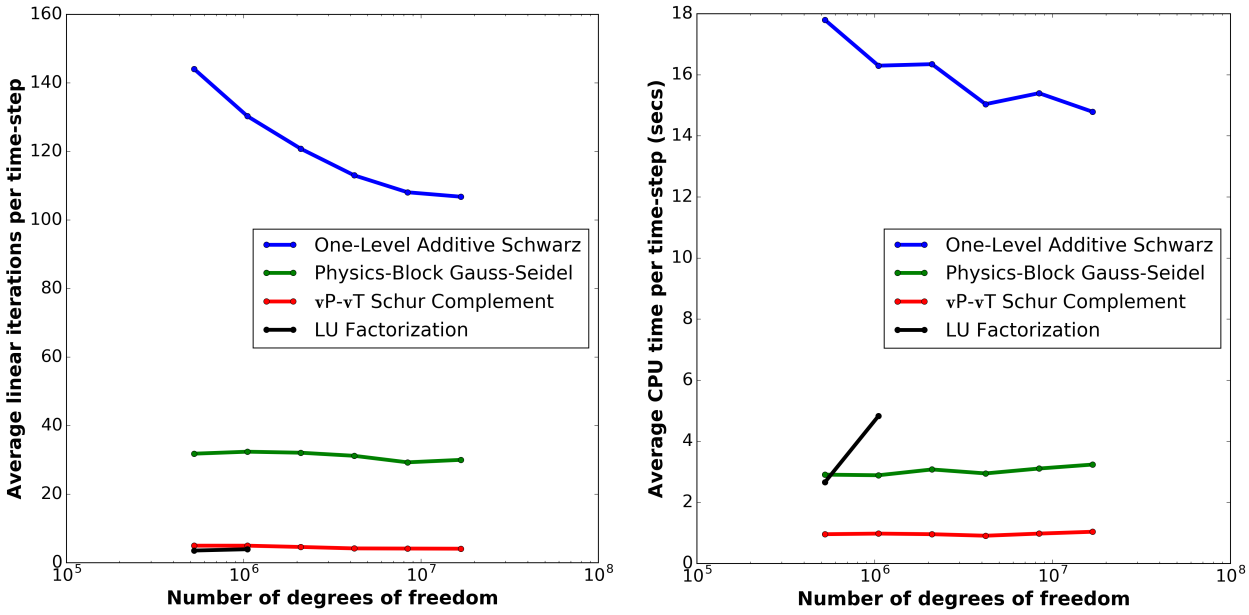


FIGURE 8.4. Weak scaling (fixed CFL) study for the low-Mach lid-driven cavity flow problem.

8.3.3. Acoustic CFL Number Study. Next, in Figure 8.5, we study the effect of the acoustic CFL number on the performance of the different preconditioners. In each case, we increase the sound speed by a factor of 10, which proportionally increases the acoustic CFL number from 1 to $1,280$. Since the Mach number is a function of the sound speed, it decreases from 10^{-2} to 10^{-5}. All other time-scales were resolved.

[1]Linear iterations are measured over the full time step, which may include several Newton steps. Additionally, since the Jacobian matrix is only approximate, an LU factorization preconditioner will still require more than one GMRES iteration.

AMG on the fully-coupled system and the one level-additive Schwarz preconditioners were unable to converge for the problems above acoustic CFL numbers of 1 and 10, respectively. This result is not surprising as the high CFL numbers create a strong global stiffness, leading to non-diagonal dominance [13]. The physics-block Gauss-Seidel preconditioner was more effective in capturing the global coupling and as a result was very efficient when the acoustic CFL number was less than 10. For larger CFL numbers, the physics-block Gauss-Seidel preconditioner became ineffective and failed to converge above CFL numbers of 100. The only preconditioners that were moderately independent of the acoustic CFL number were the LU factorization and the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioners. Both methods had a constant number of outer FGMRES iterations per time step, due to the robust coupling of the velocity-pressure system. We observe that the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner had the best performance as it converged with the least amount of CPU time.

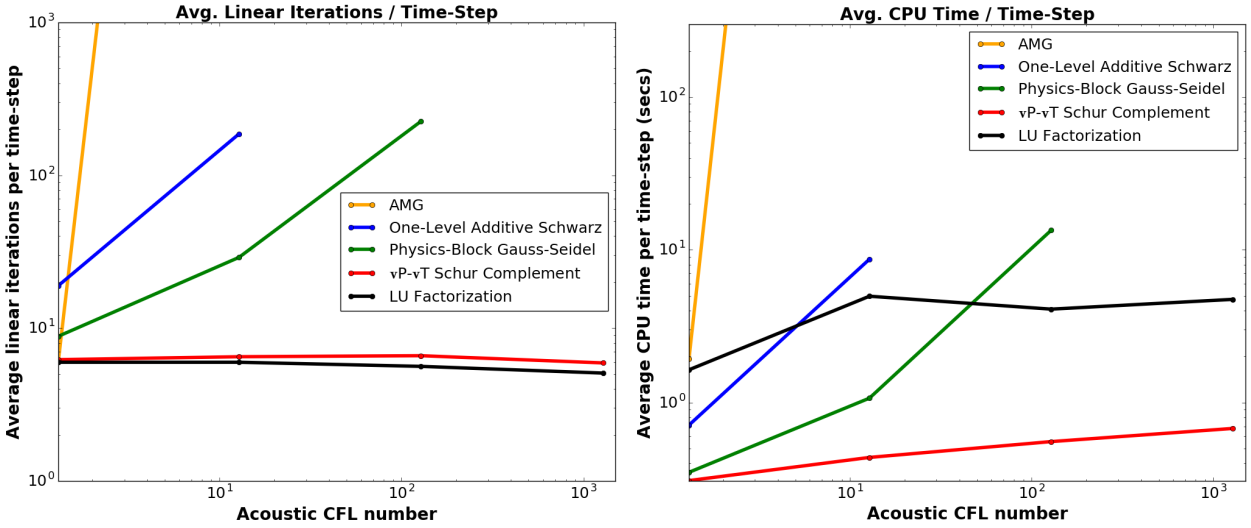


FIGURE 8.5. Acoustic CFL number study for the low-Mach driven cavity flow problem.

8.3.4. $\mathbf{v}P$-$\mathbf{v}T$ Schur Complement Preconditioner: Block Solver/Tolerance Study. In Figures 8.6 and 8.7, we compare the performance of different relative block-tolerances and various combinations of solvers and preconditioners, respectively, as a function of time step for the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner. As the time step increases in each case for both studies, all of the CFL and Fourier numbers increase, as seen in Table 8.1. In both studies, we observe that

as the time step increases, the average number of iterations and CPU time per time step increases for all cases, which is expected since the condition number of the underlying systems is a function of the CFL/Fourier numbers.

For the relative block tolerance study in Figure 8.6, each block is solved with AMG preconditioned FGMRES (AMG-FGMRES). We observe that all three cases require a similar number of outer GMRES iterations. A relative block tolerance of 10^{-1}, however, converges with the shortest CPU-time. To maximize computational efficiency, the block-solves should have a loose relative tolerance in the preconditioner.

For the solver/preconditioner study in Figure 8.7, a fixed relative block tolerance of 10^{-1} is used. GMRES as a standalone block-solver was the least robust and failed to converge above an acoustic CFL number of 300. AMG as a standalone block-solver and SOR-FGMRES were more robust solvers, but failed to converge past an acoustic CFL number of 3000. An LU factorization and AMG-FGMRES were the most robust block-solvers and were able to converge the most ill-conditioned case, corresponding to an acoustic CFL number greater than 30,000 and an advection CFL number of 100.

Time Step	CFL_c	CFL_{mat}	Fo_α	Fo_ν
2×10^{-4}	38	0.1	0.17	0.017
2×10^{-3}	380	1	1.7	0.17
2×10^{-2}	3800	10	17	1.7
2×10^{-1}	30800	100	170	17

TABLE 8.1. Time step correspondence to CFL and Fourier numbers.

8.3.5. vP-vT Schur Complement Preconditioner: Strong Scaling. To demonstrate parallel scalability, we perform a strong scaling study, shown in Figure 8.8. In this study, we counted the total CPU-time over 50 time steps, starting from $t = 0$. A square domain with a very fine, 4047×4047, mesh resolution was used. The number of processors varied from 144 to 9216. Since the problem size was fixed, the number of elements per processor varied from 113,737 to

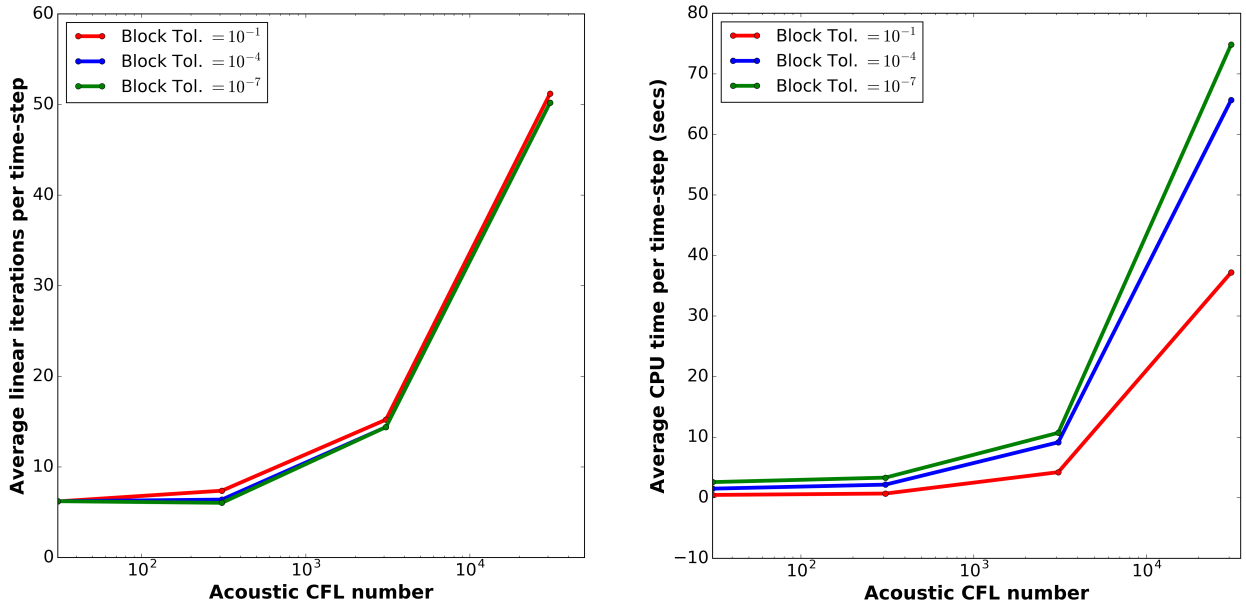


FIGURE 8.6. Relative block tolerance study for the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner as a function of time step.

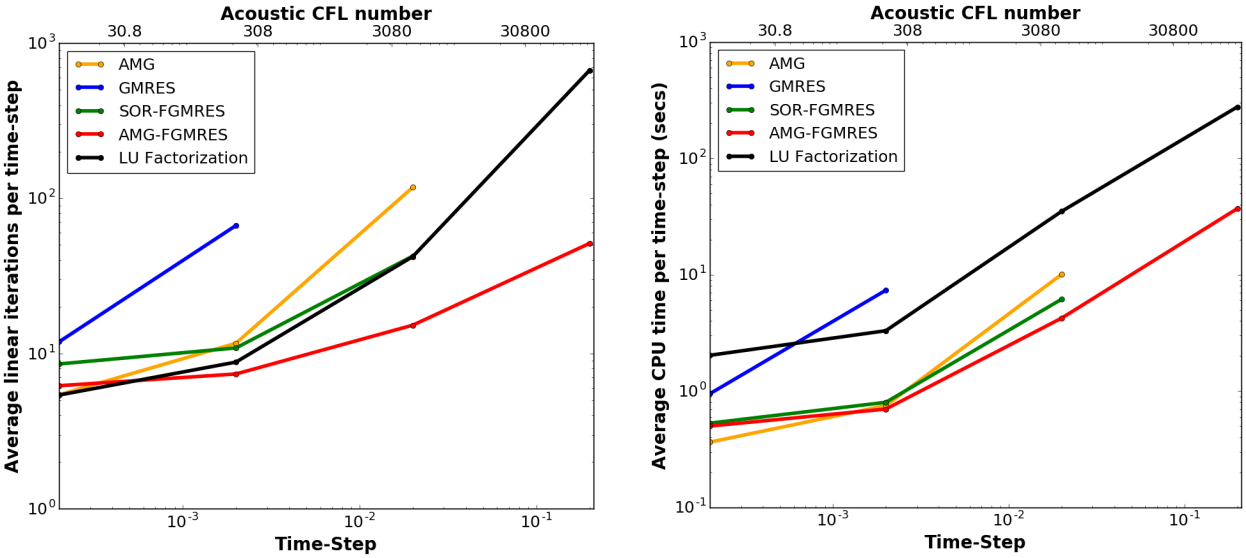


FIGURE 8.7. Preconditioner/solver study for the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner as a function of time step.

1,777. The acoustic CFL number was 120 and all other time-scales were resolved.

For parallel domain decomposition, we use the ParMETIS library [35]. The processor domain decomposition is visualized in Figure 8.9 for the case with 9216 processors. In Figure 8.8, we observe ideal scaling from 113,737 elements per processor down to 1,777 elements per processor. At 1,777 elements per processor, the ratio of ghost zones to the total number of zones is 20% and thus we see that the scaling flattens out, as expected, since the communication between processor zones begins to dominate.

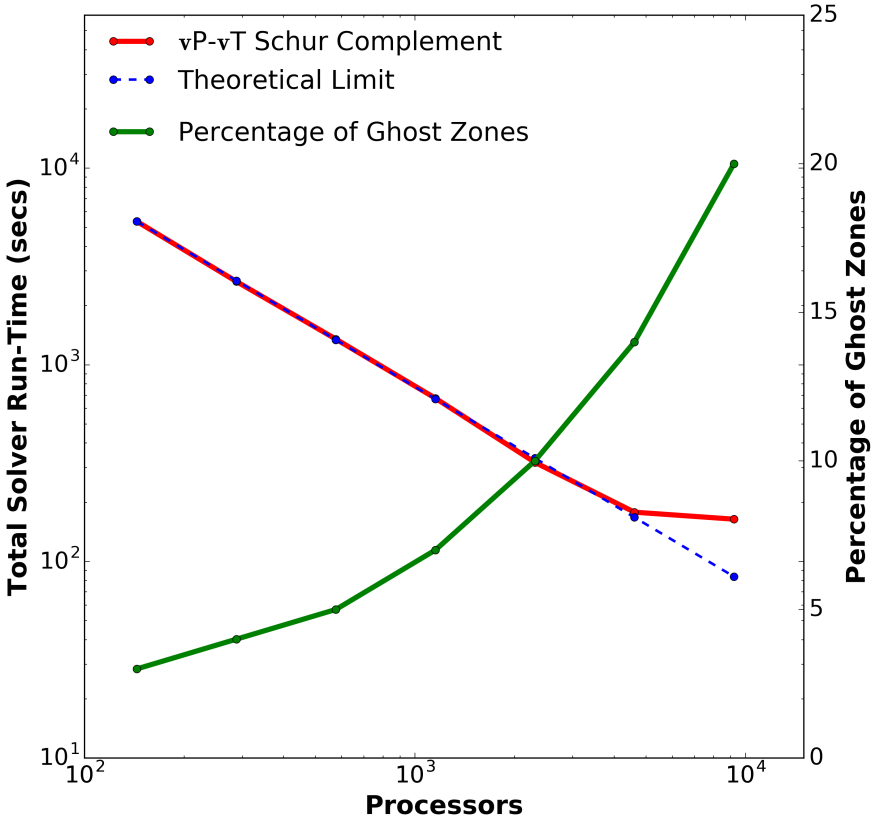


FIGURE 8.8. Strong scaling study for 2D lid-driven cavity flow problem with 64 million degrees of freedom. For the given mesh resolution, the preconditioner/solver exhibits ideal scaling up to 9216 processor cores, demonstrating excellent parallel scalability for the given mesh resolution.

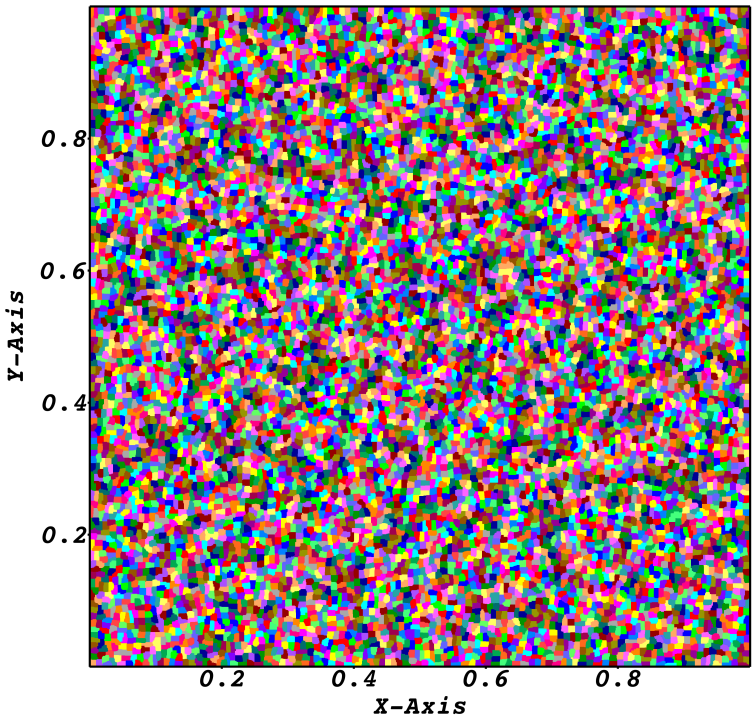


FIGURE 8.9. Processor domain decomposition across 9,216 processor cores for the 2D lid-driven cavity flow problem. Each colored island represents a particular processor's domain.

8.4. Low-Mach Melt Convection of a Steel Brick Results

In this numerical example, shown in Figure 8.10, the material is initially solid steel with a temperature of $T = 1$ on a wedged-shape, non-uniform mesh. The top and bottom walls have Dirichlet boundary conditions for temperature and are both initially set to a temperature of $T = 1$. The bottom wall is quickly ramped to a higher temperature of $T = 2$, inducing melt convection from the bottom. The left and right walls have adiabatic (zero heat flux) temperature boundary conditions. All four walls enforce a no-slip boundary condition on velocity. The melt temperatures are $T_S = 1.4$ and $T_L = 1.6$, corresponding to a mushy region thickness of $\epsilon = 0.2$. For all of the studies, we staredt at $t \approx 2$ and counted the number of outer FGMRES iterations and CPU time per time step, averaged over 50 time steps.

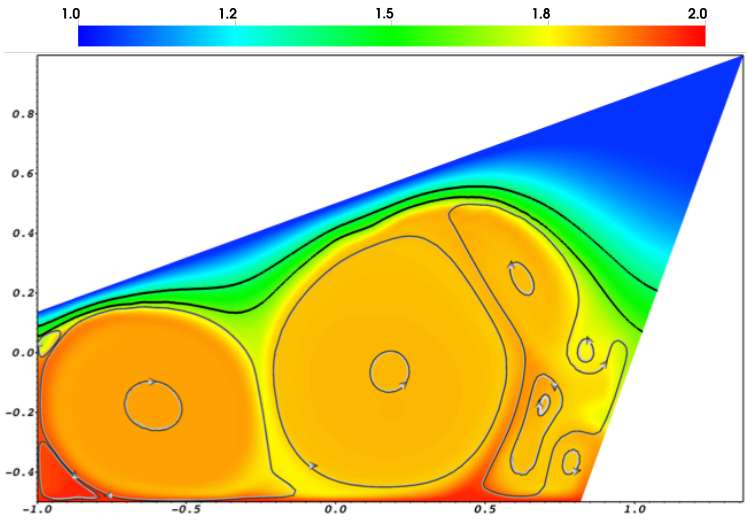


FIGURE 8.10. Snapshot of temperature and streamlines for melting of a steel brick on a non-uniform mesh. Black contour lines represent the solid-liquid interface.

8.4.1. Eigenvalues for a Solidification Phase Change Problem. The eigenvalues of an unpreconditioned Jacobian matrix is plotted for a phase change problem in Figure 8.11. In the phase change case, the condition number is on the order of a billion and the eigenvalues have now spread across the imaginary axis, presenting significant challenges for the solvability of the underlying linear system.

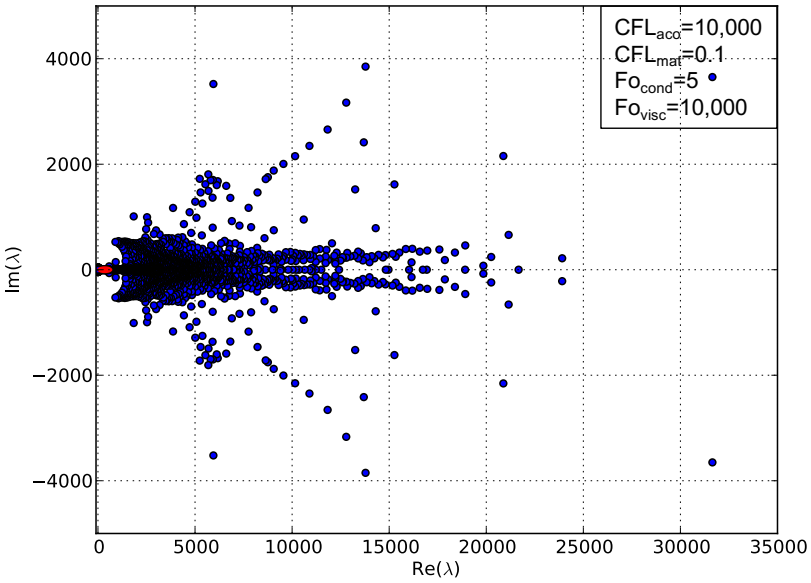


FIGURE 8.11. Eigenvalues of an unpreconditioned Jacobian matrix for a solidification phase change problem. For comparison, the eigenvalues from Figure 8.3 are shown in red. For phase change, the condition number is very large, $\kappa = 10^9$.

8.4.2. vP Schur Complement Preconditioner for Varying Viscosity Ratios. To illustrate the numerical challenges associated with the viscosity-strength model in melting and solidification problems, we compare the performance between three variations of the $\mathbf{v}P$ Schur complement preconditioner, outlined in Section 8.2.4, for increasing viscosity ratios, as shown in Figure 8.12. A 800×1600 mesh resolution was used for this study.

We observe that for large viscosity ratios $\left(\frac{\mu_s}{\mu_l} > 1000\right)$, the case which solves the approximate Schur complement system, $\tilde{\mathbf{S}}_{\mathbf{v}P}$, performs poorly compared to solving the full (matrix-free) Schur complement system. This result is not surprising since $\mathbf{M}_{\mathbf{vv}}$ becomes less diagonally dominant as the viscosity ratio increases, making $\mathbf{D}_{\mathbf{vv}}$ a poor approximation. The two cases that iteratively solve the exact Schur complement system, however, have excellent performance and converge with a similar number of outer GMRES iterations. Performance in run-time is further improved in the final case where the full Schur complement system is preconditioned with the approximate $\tilde{\mathbf{S}}_{\mathbf{v}P}$ Schur complement matrix as a preconditioner to the full Schur complement system.

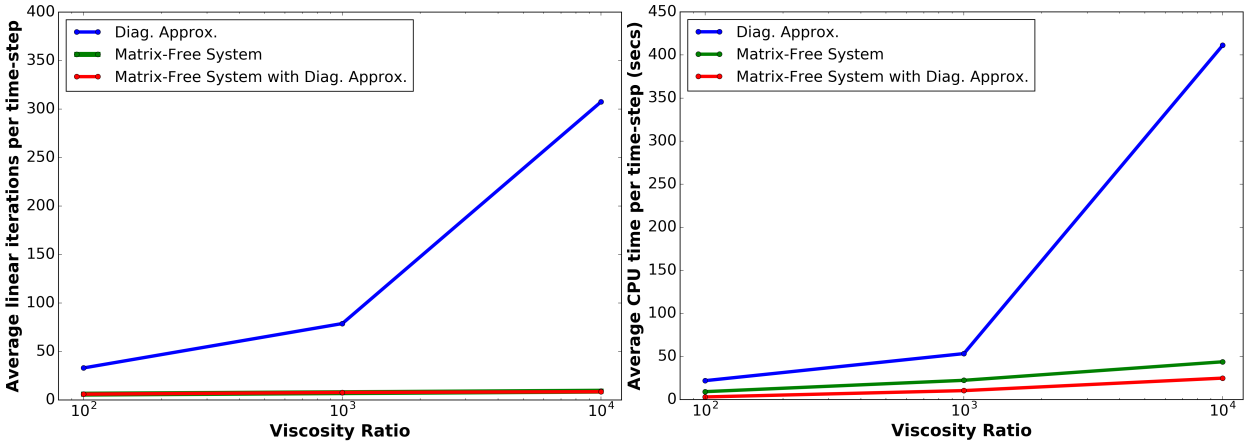


FIGURE 8.12. Viscosity ratio study of the $\mathbf{v}P$ Schur complement preconditioner.

8.4.3. $\mathbf{v}P$ Schur Complement Preconditioner Block Tolerance Study. In Figure 8.13, we compare the performance of different relative block-tolerances on a 400×800 mesh. As the time step increases in each case, all of the CFL and Fourier numbers increase. We generally observe that as the time step increases, the average number of iterations and CPU time per time step increase, which is expected since the condition number of the underlying systems is a function of the CFL/Fourier numbers.

For the relative block tolerance study in Figure 8.13, each block is solved with AMG preconditioned FGMRES. We observe that the loosest relative block tolerance of 10^{-1} requires the most outer GMRES iterations to converge, followed by the 10^{-3} case. The CPU-time shown in the the right graph of Figure 8.13, shows the opposite trend, which indicates a tradeoff between the relative block tolerance and the number of outer GMRES iterations. Therefore, a relative block tolerance of 10^{-1} requires the most outer GMRES iterations but is the most computationally efficient case (in CPU-time).

8.4.4. Weak Scaling for $\mathbf{v}P - \mathbf{v}T$ Schur Complement Preconditioner: High-Order rDG. In Figure 8.14, we compare the performance of three rDG schemes by conducting a weak scaling study (fixed CFL). Since the higher-order schemes have more degrees of freedom per element, the total number of degrees of freedom is kept constant between the schemes, instead of the number of elements. Table 8.2 shows the range of mesh sizes, range of Fourier numbers, and

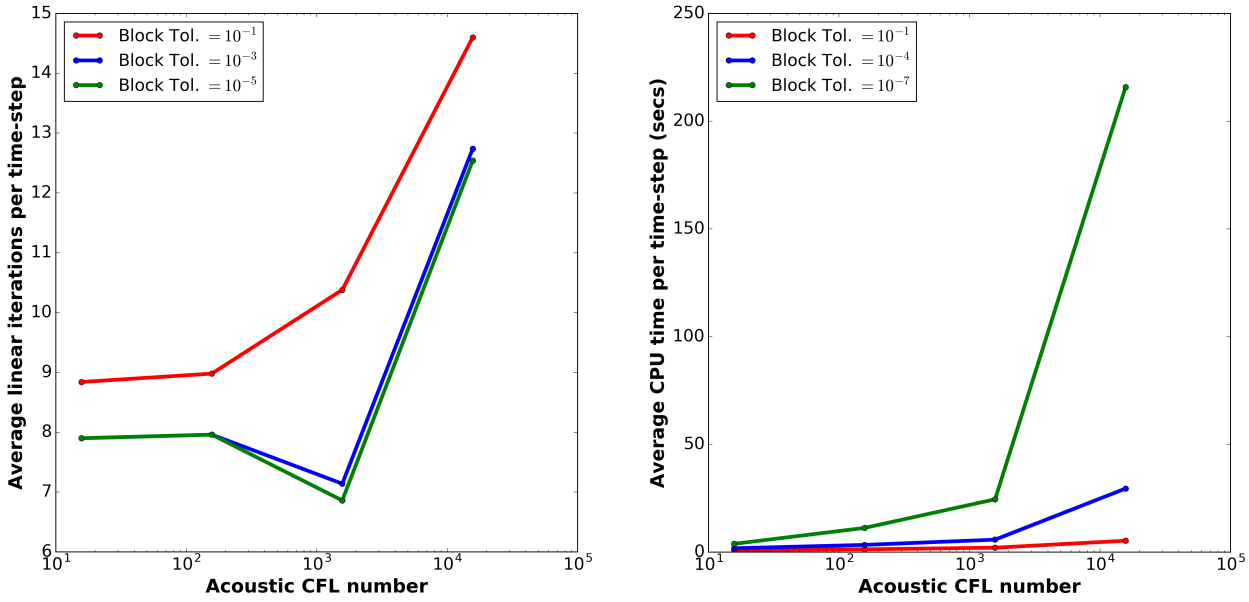


FIGURE 8.13. Relative block tolerance study for the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner on the 2D melting brick problem.

acoustic CFL number for three different rDG schemes. The total number of degrees of freedom for each scheme ranges from 288,000 to 2.3 million DoFs. For this study, we start at $t \approx 10$ and count the number of outer FGMRES iterations and CPU time per time step, averaged over 50 time steps. The number of processors varied from 16 to 128 for each scheme.

In Figure 8.14 on the left, we observe that all three rDG schemes have good algorithmic scalability and have the roughly the same number of outer GMRES iterations. On the right, we see that the low-order, P_0P_1, scheme converges in the least CPU-time. As seen in Table 8.2, even though the acoustic CFL and viscous Fourier number are the highest for P_0P_1 (since the mesh is the finest), the smaller matrix bandwidth more than compensates for the higher CFL/Fourier numbers, leading to a faster run-time. We also note that the P_2P_3 scheme converges moderately faster than the P_1P_3 scheme. Even though P_2P_3's matrix bandwidth is slightly larger, it has a cheaper cost of reconstruction, leading to a slightly faster run-time. We note that since the number of degrees of freedom were kept constant between each of the schemes, we would generally expect the low-order methods to be more efficient (in CPU-time) than the higher-order methods, due to the smaller matrix bandwidth. An efficiency comparison between the P_0P_1 and P_2P_3 schemes was

conducted in Section 4.2, which demonstrated that for the same resolving power, the P_2P_3 scheme needed to solve for less total degrees of freedom.

Scheme	Smallest Mesh	Largest Mesh	Min. Fo_ν	Max. Fo_ν	CFL_c
P_0P_1	380×760	1075×2150	88	248	597
P_1P_3	219×438	620×1240	29	82	344
P_2P_3	155×310	438×876	15	41	244

TABLE 8.2. Required mesh resolution and corresponding CFL/Fourier numbers for all three rDG schemes to have the same solution vector size.

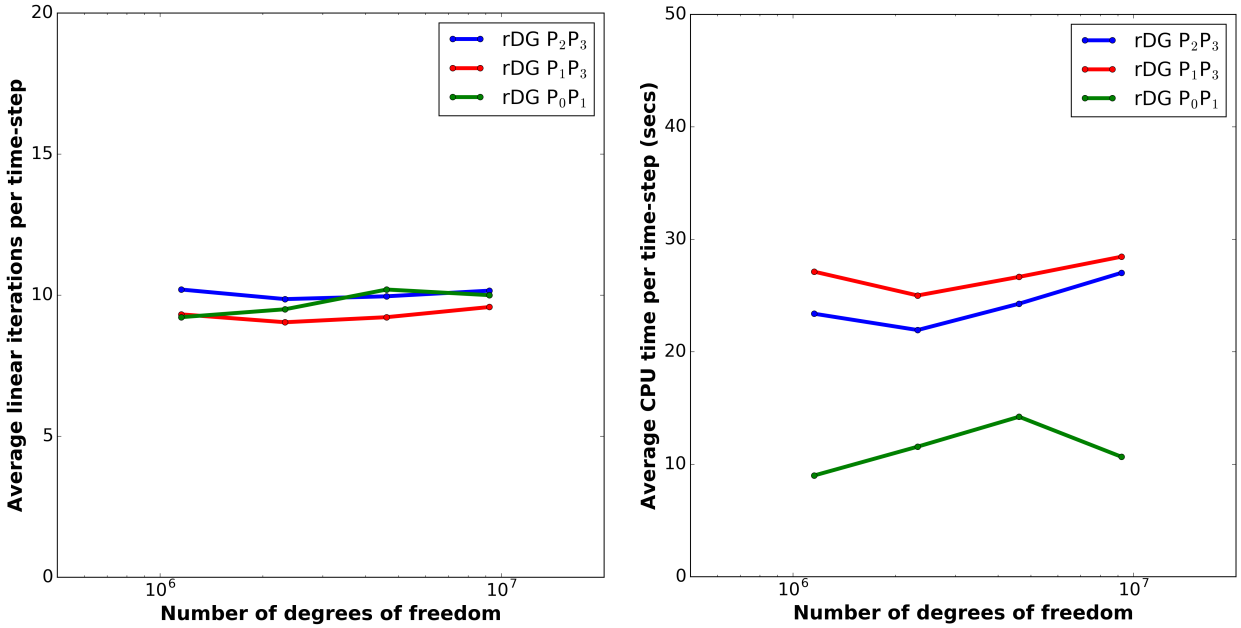


FIGURE 8.14. Weak scaling (fixed CFL) study comparing P_0P_1, P_1P_3, and P_2P_3 schemes for the same solution vector size.

8.5. 3D Selective Laser Melting Results: Single Track

We consider a moving 3D laser-induced melt convection problem on a single-track. The domain is 1 mm long $\times$ 0.3 mm wide $\times$ 0.15 mm high. A similar domain is shown in Figure 7.3. The material is initially solid steel at room temperature, $T_0 = 300$ K. The bottom wall has a Dirichlet boundary condition for temperature, fixed at $T = 300\,\mathrm{K}$, while the other walls have adiabatic (zero heat flux) temperature boundary conditions. A no-slip velocity boundary condition is enforced at all walls. The laser power is 200W and has a scan speed of 2000 mm/s. For computational efficiency, we only simulate half the domain since the problem is symmetrical across the Z-axis.

8.5.1. Weak Scaling. In Table 8.3, we conduct a (fixed CFL) weak scaling study for the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner. Starting at $t = 0$, we count the number of outer FGMRES iterations and CPU time per time step, averaged over 50 time steps. In all of the cases, the number of degrees of freedom per processor is fixed, to keep the workload constant. The number of degrees of freedom is varied from 5 million up to 40 million, while the number of processors varied from 64 to 512. The acoustic CFL number is 50.

As we increase the total problem size, the number of outer FGMRES iterations per Newton step and number of Newton steps per time step are roughly constant. This example demonstrates that the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner has excellent algorithmic scalability in 3D.

DoF	FGMRES/Newton	Newton/Cycle	CPU-Time/Cycle (Secs)
5 million	16.2	1.0	10.8
10 million	17.0	1.0	15.4
20 million	18.0	1.0	19.0
40 million	18.1	1.0	19.74

TABLE 8.3. Weak scaling (fixed time step) study for the 3D laser melting problem.

8.6. 3D Selective Laser Melting Results: Multiple Tracks

In our final numerical example, we consider a computationally challenging, moving 3D laser-induced melt convection problem on multiple melt tracks. The domain is 1 mm long $\times$ 1 mm

wide × 0.3 mm high, as shown in Figures 8.15. The material is initially solid steel at room temperature, $T_0 = 300$ K. The bottom wall has a Dirichlet boundary condition for temperature, fixed at $T = 300\,\mathrm{K}$, while the other walls have zero heat flux temperature boundary conditions. A no-slip velocity boundary condition is enforced at all walls. The laser power is 500W and has a scan speed of 4000 mm/s. As discussed in Section 7.1, the large temperature values are due to neglecting the evaporation, radiation, and Marangoni convection.

8.6.1. Weak Scaling. In Table 8.4, we conduct a (fixed CFL) weak scaling study for the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner. Starting at $t = 0$, we count the number of outer FGMRES iterations and CPU time per time step, averaged over 300 time steps. In all of the cases, the number of degrees of freedom (DoFs) per processor is fixed, to keep the workload constant. The number of degrees of freedom is varied from 5 million up to 40 million, while the number of processors varied from 108 to 864[2].

As we increase the total problem size, the number of outer FGMRES iterations per Newton step and number of Newton steps per time step is roughly constant. This demonstrates that the $\mathbf{v}P$-$\mathbf{v}T$ Schur complement preconditioner has excellent algorithmic scalability in 3D.

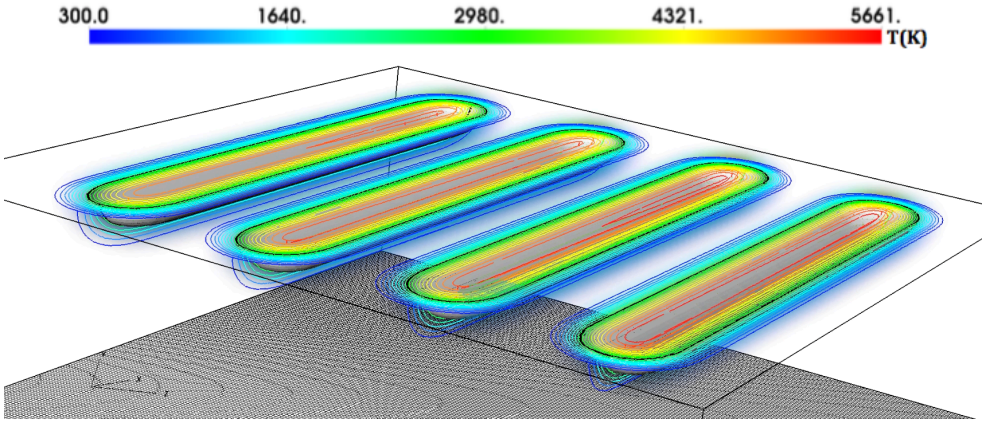


FIGURE 8.15. Temperature and melting front dynamics from 3D laser-induced phase change on multiple melt-tracks.

[2]The 3D SLM simulation on multiple-tracks also converged on 25 million elements (125 million DoFs) with 4096 processors using the Jade supercomputer.

DoF	FGMRES/Newton	Newton/Cycle	CPU-Time/Cycle (Sec)
5 million	20.3	6.3	70.1
10 million	20.5	6.0	71.7
20 million	19.3	5.9	72.54
40 million	20.8	5.3	63.46

TABLE 8.4. Weak scaling (fixed time step) study for the 3D laser melting problem on multiple melt-tracks.

CHAPTER 9

Conclusion and Future Directions

9.1. Concluding Remarks

A scalable block-based preconditioner was developed for a Newton-Krylov solver of a fully-implicit rDG-based discretization of the all-speed compressible flow equations with phase change. The primary challenge was to robustly converge time-accurate solutions at high CFL/Fourier numbers representing an ill-conditioned system of discrete equations, corresponding to simulations of fluid flow with laser-induced phase change. To address this challenge, I developed a robust approximate block factorization preconditioner, which is a multigrid block reduction technique that reduces a fully-coupled 3×3 block system to a sequence of two 2×2 block systems: the velocity-pressure ($\mathbf{v}P$) and velocity-temperature ($\mathbf{v}T$) Schur complement systems. For all tested problems, I found that the monolithic algebraic multigrid and one-level additive Schwarz preconditioners were ineffective at high CFL/Fourier numbers. The physics-block Gauss-Seidel preconditioner fared better, but the proposed $\mathbf{v}P$-$\mathbf{v}T$ Schur complement based preconditioner converged the fastest and required the least number of iterations. The $\mathbf{v}P$-$\mathbf{v}T$ preconditioner, which uses AMG as a preconditioner to GMRES for the inner block solves, exhibited excellent algorithmic and parallel scalability.

Furthermore, the framework was shown to be robust for a wide range of non-dimensional numbers and phase change configurations. To enforce the attenuation of velocity in the mushy and solid phases, I developed a velocity suppression model, which combines the variable viscosity and Darcy source term model. I have demonstrated that this framework is robust and capable of solving phase change problems for a wide range of parameters, including both melting and freezing in various configurations and moving laser-induced melt-convection problems at large CFL/Fourier numbers. The high-order rDG-based schemes were shown to produce highly accurate solutions and resolve flow features on very coarse meshes. Solutions converged at low-Mach numbers ($Ma \leq 10^{-2}$)

without explicit acoustic filtering, demonstrating all-speed flow capabilities, which is necessary for modeling tightly-coupled rapid phase change processes, such as evaporation/condensation, in metal additive manufacturing processes such as SLM.

9.2. Future Work

9.2.1. Physics Enhancement. Currently, large viscosity ratios between the solid and liquid phases $\left(\frac{\mu_s}{\mu_l} \geq 1000\right)$ are challenging for the solvability of the underlying linear algebra, since the velocity matrix $\mathbf{M_{vv}}$ becomes non-diagonally dominant due to the large off-diagonal entries. The non-diagonal dominant rows, however, are confined to a thin region near the solid-liquid interface. Future work will involve detecting the non-diagonally dominant rows at the interface and solving the interface elements with a more robust LU factorization.

Additionally, a novel Marker Re-Distancing (MRD) and sharp mix-cell reconstruction method was recently developed for evolving multi-material interfaces at high-order [67]. With multi-phase capabilities, future work involves incorporating more realistic physics, such as radiation, evaporation/condensation, surface tension, Marangoni convection, and recoil forces due to material evaporation. With a suitable laser model based on ray tracing, we will be able to simulate 3D multi-material metal additive manufacturing processes such as SLM [36, 37]. Since heating and cooling rates in the SLM process exceed 10^5 K/s, an equilibrium phase change model is not appropriate and solidification with undercooling must be accounted for [38]. With the interface tracking method, we can incorporate a kinetic model for rapid solidification, a non-equilibrium thermodynamics process [85].

9.2.2. '$N \times N'$ Multigrid Block Reduction. Future work involves solving larger block systems from higher-order (greater than 2^{nd}-order) schemes, binary and ternary systems, and coupling to solid and thermal mechanics solvers. Larger 4×4, 5×5, and in general $N \times N$ block systems will need to be preconditioned in a scalable and robust manner. I plan to extend the 2×2 Schur complement preconditioning technique used in this research to solve these larger block systems in

a nested $N \times N$ multigrid block reduction strategy.

Currently, our framework solves a 3×3 block system in the preconditioning stage with a reduced set of two 2×2 Schur complements: the velocity-pressure and velocity-temperature systems. This preconditioning strategy is effective when the velocity-pressure and velocity-temperature couplings are strong. This strategy neglects the temperature-pressure coupling, however, which is not appropriate when the temperature-pressure coupling is strong, as in the case of more sophisticated equations of state (Appendix A.1). The first step is to robustly solve the 3×3 block system using a single 2×2 Schur complement between the full [velocity-pressure] and temperature system

$$\begin{pmatrix} M_{[P\mathbf{v}]} & M_{[P\mathbf{v}]T} \\ M_{T[P\mathbf{v}]} & M_{TT} \end{pmatrix} \begin{pmatrix} x_{[P\mathbf{v}]} \\ x_T \end{pmatrix} = \begin{pmatrix} b_{[P\mathbf{v}]} \\ b_T \end{pmatrix}. \tag{9.1}$$

As with any 2×2 block system, there are always two distinct Schur complements. Using forward and backward substitution with the Schur complement of the $M_{[P\mathbf{v}]}$ block, leads to

$$\begin{aligned} x_T^* &= M_{TT}^{-1} b_T \\ x_{[P\mathbf{v}]} &= S_{[P\mathbf{v}]T}^{-1} \left(b_{[P\mathbf{v}]} - M_{[P\mathbf{v}]T} x_T^* \right) \\ x_T &= M_{TT}^{-1} \left(b_T - M_{T[P\mathbf{v}]} x_{[P\mathbf{v}]} \right), \end{aligned}$$

where

$$S_{[P\mathbf{v}]T} = M_{[P\mathbf{v}]} - M_{[P\mathbf{v}]T} M_{TT}^{-1} M_{T[P\mathbf{v}]}.$$

On the other hand, using forward and backward substitution with the Schur complement of the M_{TT} block leads to

$$\begin{aligned} x_{[P\mathbf{v}]}^* &= M_{[P\mathbf{v}]}^{-1} b_{[P\mathbf{v}]} \\ x_T &= S_{T[P\mathbf{v}]}^{-1} \left(b_T - M_{T[P\mathbf{v}]} x_{[P\mathbf{v}]}^* \right) \\ x_{[P\mathbf{v}]} &= M_{[P\mathbf{v}]}^{-1} \left(b_{[P\mathbf{v}]} - M_{[P\mathbf{v}]T} x_T \right), \end{aligned}$$

where

$$S_{T[P\mathbf{v}]} = M_{TT} - M_{T[P\mathbf{v}]} M^{-1}_{[P\mathbf{v}]} M_{[P\mathbf{v}]T}.$$

Both of these approaches are mathematically equivalent and have different pros and cons. In the first approach, only one 2×2 system, $S_{[P\mathbf{v}]T}$, needs to be solved. Solving this system with a nested Schur complement, however, would need to be approximated in a computationally efficient manner[1]. In the second approach, two 2×2 systems ($M_{[P\mathbf{v}]}$) would need to be solved, which doubles the current computational cost. Each of these 2×2 systems, however, can be solved with a nested Schur complement system (using the $\mathbf{v}$-P Schur complement), which is known to be efficient since it is already used in the present work. These nested Schur complement techniques can be generalized to an $N \times N$ level multigrid block reduction strategy and can be applied to arbitrarily large block systems, such as higher-order schemes. Currently the preconditioning technique for the low and high-order rDG schemes are identical and both cases involve the Schur complement systems of the full velocity-pressure and velocity-temperature systems. Instead of fully coupling the physics systems for the higher order schemes, we want to extend the approximate block factorization technique between the orders of the degrees of freedom, with a nested Schur complement (p-multigrid technique). The nested technique can also be applied to reduce the velocity system to its individual velocity components.

Lastly, I plan to tightly couple the rDG CFD fields (velocity, pressure, and temperature) to a solid-thermal mechanics Lagrangian/ALE solver (displacements, temperature). With the additional degrees of freedom, the size of the block system and solution vector size would increase. To precondition this system, we would like to apply the nested multigrid block reduction technique to these fully-coupled solid-thermal-fluid systems.

[1]Research would need to be done to develop effective approximations to this system

APPENDIX A

A.1. Equations of State (EOS)

A *3-parameter* $(\rho_0, c, \mathbf{e})$ EOS with an explicit pressure-temperature coupling is given by

$$P(\rho, \mathbf{e}) = P_0 + A_1 \left(\frac{\rho}{\rho_0} - 1 \right) + A_2 \mathbf{e}, \tag{A.1}$$

where the sound speed is a function of both the density and internal energy

$$c^2 = \frac{\partial P}{\partial \rho}\Big|_T + \frac{P}{\rho^2} \frac{\partial P}{\partial T}\Big|_\rho = \frac{A_1}{\rho} + \frac{P}{\rho^2} A_2 C_v, \tag{A.2}$$

where A_1 and A_2 are input coefficients.

The γ-law gas EOS is given by

$$P = \rho(\gamma - 1)\mathbf{e}, \tag{A.3}$$

where $\gamma = \frac{C_p}{C_v} = 1.4$ for air.

A.2. CFL Conditions and Fourier Numbers

The CFL (Courant-Friedrichs-Lewy) conditions and Fourier numbers are defined as follows

$$\mathrm{CFL}_{\mathrm{aco}} = \frac{c\Delta t}{\Delta x} \tag{A.4}$$

$$\mathrm{CFL}_{\mathrm{mat}} = \frac{\mathbf{v}\Delta t}{\Delta x} \tag{A.5}$$

$$\mathrm{Fo}_\alpha = \frac{\alpha \Delta t}{\Delta x^2} \tag{A.6}$$

$$\mathrm{Fo}_\nu = \frac{\nu \Delta t}{\Delta x^2}, \tag{A.7}$$

where c is the sound speed, $\mathbf{v}$ is the material velocity, α is the thermal diffusivity, ν is the kinematic viscosity, Δt is the current time-step of the simulation, and Δx is the length of the minimum mesh width. Note that the Mach number and the Prandtl number can be numerically calculated as, $Ma = \frac{\mathrm{CFL_{mat}}}{\mathrm{CFL_{aco}}}$, and $Pr = \frac{\mathrm{Fo}_\nu}{\mathrm{Fo}_\alpha}$, respectively.

A.3. Laser and Material Properties

We use the following material properties for stainless steel [2, 6]:

$\bar{\rho}$	Density	$7,800$	$\mathrm{kg/m^3}$
$\bar{C_v}$	Specific heat	800	$\mathrm{J/(kg \cdot K)}$
$\bar{k}$	Thermal conductivity	36	$\mathrm{W/(m \cdot }K)$
$\bar{\mu}$	Dynamic viscosity	6.1×10^{-3}	$\mathrm{N \cdot s/m^2}$
$\bar{\beta}$	Volumetric coefficient of expansion	5×10^{-5}	1/K
T_S	Solid melt temperature	$1,675$	K
T_L	Liquid melt temperature	$1,725$	K
$\overline{u_f}$	Latent heat of fusion	$285,000$	$\mathrm{J/(kg \cdot K)}$.

We define the last two scaling parameters:

$\bar{L}$	Characteristic length scale	1	mm
$\overline{\Delta T}$	Characteristic temperature difference	$2,700$	K.

$\bar{L}$ corresponds to a single-track length used in SLM simulations in [36]. $\overline{\Delta T}$ is the difference between the saturation temperature of stainless steel at atmospheric pressure and room temperature.

We use the following process parameters:

$$\begin{aligned}
\text{Laser power} &= 200 - 500\,\text{W} \\
\text{Material Absorption Coefficient} &= 0.3 \\
\text{Scan speed} &= 2000 - 4000\,\frac{\text{mm}}{\text{s}} \\
\text{Beam radius} &= 0.026\,\text{mm} \\
\text{Max. deposition length} &= 0.05\,\text{mm}.
\end{aligned}$$

Bibliography

[1] Additive manufacturing media. www.additivemanufacturing.media.

[2] ASM Aerospace Specification Metals Inc. http://asm.matweb.com/.

[3] ALE3D Web Page. https://wci.llnl.gov/simulation/computer-codes/ale3d, 2013.

[4] EH Amara, R Fabbro, and A Bendib. Modeling of the compressible vapor flow induced in a keyhole during laser welding. *Journal of Applied Physics*, 93(7):4289–4296, 2003.

[5] John David Anderson and J Wendt. *Computational Fluid Dynamics*, volume 206. Springer, 1995.

[6] European Stainless Steel Development Association et al. Stainless steel: tables of technical properties. *Materials and Applications Series*, 5, 2000.

[7] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[8] George Keith Batchelor. *An Introduction to Fluid Dynamics*. Cambridge university press, 2000.

[9] Alberto Beccantini, Etienne Studer, S Gounand, J-P Magnaud, Thibaud Kloczko, Christophe Corre, and Sergueï Kudriakov. Numerical simulations of a transient injection flow at low mach number regime. *International journal for numerical methods in engineering*, 76(5):662–696, 2008.

[10] H. Bijl, M.H. Carpenter, V.N. Vatsa, and C.A. Kennedy. Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow. *Journal of Computational Physics*, 179:313–329, 2002.

[11] PC Carman. Fluid flow through granular beds. *Chemical Engineering Research and Design*, 75:S32–S48, 1997.

[12] Mark H Carpenter, CA Kennedy, Hester Bijl, SA Viken, and Veer N Vatsa. Fourth-order runge-kutta schemes for fluid mechanics applications. *Journal of Scientific Computing*, 25(1-2):157–194, 2005.

[13] L. Chacón and A. Stanier. A scalable, fully implicit algorithm for the reduced two-field low-extended MHD model. *Journal of Computational Physics*, 326:763–772, 12 2016.

[14] Todd T. Chisholm and David W. Zingg. A Jacobian-free Newton–Krylov algorithm for compressible turbulent fluid flows. *Journal of Computational Physics*, 228(9):3490 – 3507, 2009.

[15] Y-H Choi and Charles L Merkle. The application of preconditioning in viscous flows. *Journal of Computational Physics*, 105(2):207–223, 1993.

[16] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135(2):118–125, 1997.

[17] Eric C Cyr, John N Shadid, and Raymond S Tuminaro. Stabilization and scalable block preconditioning for the Navier–Stokes equations. *Journal of Computational Physics*, 231(2):345–363, 2012.

[18] Eric C Cyr, John N Shadid, Raymond S Tuminaro, Roger P Pawlowski, and Luis Chacón. A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD. *SIAM Journal on Scientific Computing*, 35(3):B701–B730, 2013.

[19] Ionut Danaila, Raluca Moglan, Frédéric Hecht, and Stéphane Le Masson. A Newton method with adaptive finite elements for solving phase-change problems with natural convection. *Journal of Computational Physics*, 274:826 – 840, 2014.

[20] Jonathan A Dantzig. Modelling liquid–solid phase changes with melt convection. *International Journal for Numerical Methods in Engineering*, 28(8):1769–1785, 1989.

[21] Georg Ehlen, Andreas Ludwig, and Peter R Sahm. Simulation of time-dependent pool shape during laser spot welding: transient effects. *Metallurgical and Materials Transactions A*, 34(12):2947–2961, 2003.

[22] Howard Elman, Victoria E Howle, John Shadid, Robert Shuttleworth, and Ray Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 227(3):1790–1808, 2008.

[23] Howard C Elman, Victoria E Howle, John N Shadid, and Ray S Tuminaro. A parallel block multi-level preconditioner for the 3d incompressible Navier–Stokes equations. *Journal of Computational Physics*, 187(2):504–523, 2003.

[24] M Elmo and O Cioni. Low mach number model for compressible flows and application to htr. *Nuclear engineering and design*, 222(2):117–124, 2003.

[25] Félix A España, Vamsi Krishna Balla, and Amit Bandyopadhyay. Laser processing of bulk al–12si alloy: influence of microstructure on thermal properties. *Philosophical Magazine*, 91(4):574–588, 2011.

[26] Katherine J Evans, Dana A Knoll, and Michael Pernice. Development of a 2-D algorithm to simulate convection and phase transition efficiently. *Journal of Computational Physics*, 219(1):404–417, 2006.

[27] Robert D Falgout and Ulrike Meier Yang. HYPRE: A library of high performance preconditioners. In *International Conference on Computational Science*, pages 632–641. Springer, 2002.

[28] H.G. Fan and R. Kovacevic. Droplet formation, detachment, and impingement on the molten pool in gas metal arc welding. *Metallurgical and Materials Transactions B*, 30(4):791–801, 1999.

[29] W Scott Futral, Evi Dube, J Robert Neely, and Tim G Pierce. Performance of ALE3D on the ASCI Machines. Technical Report UCRL-JC-132166, Lawrence Livermore National Laboratory, 1999.

[30] Hervé Guillard and Cécile Viozat. On the behaviour of upwind schemes in the low mach number limit. *Computers & Fluids*, 28(1):63–86, 1999.

[31] Nayanee Gupta, Christopher Weber, and Sherrica Newsome. Additive manufacturing: Status and opportunities. 2012.

[32] L. Han and F.W. Liou. Numerical investigation of the influence of laser beam mode on melt pool. *International Journal of Heat and Mass Transfer*, 47(19–20):4385 – 4402, 2004.

[33] Zhiyu Han and Rolf D Reitz. Turbulence modeling of internal combustion engines using rng κ-ε models. *Combustion science and technology*, 106(4-6):267–295, 1995.

[34] Jeffrey Housman, Cetin Kiris, and Mohamed Hafez. Preconditioned methods for simulations of low speed compressible flows. *Computers & Fluids*, 38(7):1411–1423, 2009.

[35] George Karypis. Metis and parmetis. In *Encyclopedia of Parallel Computing*, pages 1117–1124. Springer, 2011.

[36] Saad A. Khairallah and Andrew T Anderson. Mesoscopic simulation model of selective laser melting of stainless steel powder. *Journal of Materials Processing Technology*, 214:2627–2636, 2014.

[37] Saad A Khairallah, Andrew T Anderson, Alexander Rubenchik, and Wayne E King. Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. *Acta Materialia*, 108:36–45, 2016.

[38] W King, AT Anderson, RM Ferencz, NE Hodge, C Kamath, and SA Khairallah. Overview of modelling and simulation of metal powder bed fusion process at lawrence livermore national laboratory. *Materials Science and Technology*, 31(8):957–968, 2015.

[39] Wayne King. Accelerated certification of additively manufactured metals (acamm). https://manufacturing.llnl.gov/additive-manufacturing/accelerated-certification.

[40] DA Knoll and DE Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[41] DA Knoll, DB Kothe, and B Lally. A New Nonlinear Solution Method for Phase-Change Problems. *Numerical Heat Transfer: Part B: Fundamentals*, 35(4):439–459, 1999.

[42] DA Knoll, PR McHugh, and DE Keyes. Newton-Krylov methods for low-mach-number compressible combustion. *AIAA journal*, 34(5):961–967, 1996.

[43] D.A. Knoll and V.A. Mousseau. On Newton–Krylov Multigrid Methods for the Incompressible Navier–Stokes Equations. *Journal of Computational Physics*, 163(1):262 – 267, 2000.

[44] DA Knoll, WB Vanderheyden, VA Mousseau, and DB Kothe. On Preconditioning Newton–Krylov Methods in Solidifying Flow Applications. *SIAM Journal on Scientific Computing*, 23(2):381–397, 2001.

[45] Carolin Körner, Elham Attar, and Peter Heinl. Mesoscopic simulation of selective beam melting processes. *Journal of Materials Processing Technology*, 211(6):978–987, 2011.

[46] DA Korzekwa. Truchas–a multi-physics tool for casting simulation. *International Journal of Cast Metals Research*, 22(1-4):187–191, 2009.

[47] Marcello Lappa. A mathematical and numerical framework for the analysis of compressible thermal convection in gases at very high temperatures. *Journal of Computational Physics*, 313:687–712, 2016.

[48] Patrick Le Quéré, Catherine Weisman, Henri Paillère, Jan Vierendeels, Erik Dick, Roland Becker, Malte Braack, and James Locke. Modelling of natural convection flows with large temperature differences: a benchmark problem for low mach number solvers. part 1. reference solutions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(03):609–616, 2005.

[49] Meng-Sing Liou. A Sequel to AUSM: AUSM+. *Journal of Computational Physics*, 129(2):364 – 382, 1996.

[50] Meng-Sing Liou. A sequel to AUSM, Part II: AUSM+-up for all speeds. *Journal of Computational Physics*, 214(1):137–170, 2006.

[51] Meng-Sing Liou and Christopher J Steffen. A new flux splitting scheme. *Journal of Computational physics*, 107(1):23–39, 1993.

[52] H. Luo, J.D. Baum, and R. Löhner. A Discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids. *Journal of Computational Physics*, 227(20):8875–8893, 2008.

[53] H. Luo, L. Luo, and R. Nourgaliev. A Reconstructed Discontinuous Galerkin method for the Euler equations on arbitrary grids. *Communication in Computational Physics*, 12(5):1495–1519, November 2012.

[54] H. Luo, L. Luo, R. Nourgaliev, and V. Mousseau. A Reconstructed Discontinuous Galerkin method for the compressible Euler equations on arbitrary grids. In *19th AIAA Computational Fluid Dynamics Conference*, San Antonio, Texas, USA, June 22-25 2009. AIAA 2009-3788.

[55] H. Luo, L. Luo, R. Nourgaliev, and V. Mousseau. A Reconstructed Discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids. *Journal of Computational Physics*, 229:6961–6978, 2010.

[56] H. Luo, Y. Xia, S. Li, R. Nourgaliev, and C. Cai. A Hermite WENO Reconstruction-based Discontinuous Galerkin method for the Euler equations on tetrahedral grids. *Journal of Computational Physics*, 231:5489–5502, 2012.

[57] H. Luo, Y. Xia, S. Spiegel, R. Nourgaliev, and Z. Jiang. A Reconstructed Discontinuous Galerkin method based on a Hierarchical WENO reconstruction for compressible flows on tetrahedral grids. *Journal of Computational Physics*, 236:477–492, 2013.

[58] Zhanhua Ma and Yuwen Zhang. Solid velocity correction schemes for a temperature transforming model for convection phase change. *International Journal of Numerical Methods for Heat & Fluid Flow*, 16(2):204–225, 2006.

[59] Mario J Martinez and David K Gartling. A finite element method for low-speed compressible flows. *Computer methods in applied mechanics and engineering*, 193(21):1959–1979, 2004.

[60] Materialgeeza. Wikipedia: Selective laser sintering.

[61] R.C. McCallen. ALE3D: Abitrary Lagrangian-Eulerian 2D and 3D Multiphysics Modeling and Simulation. Technical Report LLNL-MI-413853, Lawrence Livermore National Laboratory, 2009.

[62] Patrick A McMurtry, W-H Jou, J Riley, and RW Metcalfe. Direct numerical simulations of a reacting mixing layer with chemical heat release. *AIAA journal*, 24(6):962–970, 1986.

[63] VA Mousseau, DA Knoll, and WJ Rider. Physics-based preconditioning and the Newton–Krylov method for non-equilibrium radiation diffusion. *Journal of computational physics*, 160(2):743–765, 2000.

[64] C-D Munz, Sabine Roller, Rupert Klein, and Karl J Geratz. The extension of incompressible flow solvers to the weakly compressible regime. *Computers & Fluids*, 32(2):173–196, 2003.

[65] C Newman and Dana A Knoll. Physics-based preconditioners for ocean simulation. *SIAM Journal on Scientific Computing*, 35(5):S445–S464, 2013.

[66] R. Nourgaliev. Modeling and analysis of heat and mass transfer processes during in-vessel melt progression stage of Light Water Reactor (LWR) severe accidents. Technical Report ISBN 91-7170-235-0, Royal Institute of Technology, Department of Nuclear Power Safety, Stockholm, Sweden, April 27 1998. Doctoral Thesis.

[67] R. Nourgaliev, P. Greene, and S. Schofield. Marker Re-Distancing and Sharp Reconstruction for High-Order Multi-Material Interface Evolution. 69th Annual Meeting of the APS Division of Fluid Dynamics, 2016.

[68] R. Nourgaliev, H. Luo, B. Weston, A. Anderson, S. Schofield, T. Dunn, and J.-P. Delplanque. Fully-implicit orthogonal reconstructed Discontinuous Galerkin method for fluid dynamics with phase change. *Journal of Computational Physics*, 305:964 – 996, 2016.

[69] Shengyong Pang, Xin Chen, Jianxin Zhou, Xinyu Shao, and Chunming Wang. 3D transient multiphase model for keyhole, vapor plume, and weld pool dynamics in laser welding including the ambient pressure effect. *Optics and Lasers in Engineering*, 74:47–58, 2015.

[70] HyeongKae Park, Robert Nourgaliev, Richard C Martineau, and Dana A Knoll. On physics-based preconditioning of the Navier–Stokes equations. *Journal of Computational Physics*, 228(24):9131–9146, 2009.

[71] Suhas V Patankar and D Brian Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787–1806, 1972.

[72] M Pernice and MD Tocci. A Multigrid-Preconditioned Newton–Krylov Method for the Incompressible Navier–Stokes Equations. *SIAM Journal on Scientific Computing*, 23(2):398–418, 2001.

[73] P-O Persson and Jaime Peraire. Newton-GMRES Preconditioning for Discontinuous Galerkin Discretizations of the Navier-Stokes Equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008.

[74] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 2003.

[75] Y. Saad and M.H. Schultz. GMRES: a Generalized Minimal Residual algorithm for solving linear systems. 7:856, 1986.

[76] J.N. Shadid, R.S. Tuminaro, K.D. Devine, G.L. Hennigan, and P.T. Lin. Performance of fully coupled domain decomposition preconditioners for finite element transport/reaction simulations. *Journal of Computational Physics*, 205(1):24 – 47, 2005.

[77] Videographer Robert Sickles. ACAMM, Lawrence Livermore National Laboratory.

[78] Barry Smith, Petter Bjorstad, and William Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge university press, 2004.

[79] Moulay D Tidriri. Hybrid Newton-Krylov/domain decomposition methods for compressible flows. In *Proceedings of the Ninth International Conference on Domain Decomposition Methods in Sciences and Engineering*, pages 532–539, 1998.

[80] AG Tomboulides, JCY Lee, and SA Orszag. Numerical simulation of low mach number reactive flows. *Journal of Scientific Computing*, 12(2):139–167, 1997.

[81] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[82] RS Tuminaro, CH Tong, JN Shadid, KD Devine, and DM Day. On a multilevel preconditioning module for unstructured mesh Krylov solvers: two-level Schwarz. *Communications in numerical methods in engineering*, 18(6):383–389, 2002.

[83] Eli Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics*, 72(2):277–298, 1987.

[84] V.R. Voller and 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):1709–1719, 1987.

[85] G-X Wang and EF Matthys. Numerical modelling of phase change and heat transfer during rapid solidification processes: use of control volume integrals with element subdivision. *International Journal of Heat and Mass Transfer*, 35(1):141–153, 1992.

[86] F.M. White. *Viscous Fluid Flow*. McGraw-Hill series in mechanical engineering. McGraw-Hill, 1991.

[87] Y. Xia, X. Lio, H. Luo, and R. Nourgaliev. A third-order implicit Discontinuous Galerkin method based on Hermite WENO reconstruction for time-accurate solution of the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 79:416–435, 2015.

[88] Y. Xia, H. Luo, M. Frisbey, and R. Nourgaliev. A set of parallel, implicit methods for Reconstructed Discontinuous Galerkin method for compressible flows on 3D hybrid grids. *Computers & Fluids*, 96:406–421, 2014.

[89] Y. Xia, H. Luo, and R. Nourgaliev. An Implicit Hermite WENO Reconstruction-based Discontinuous Galerkin on tetrahedral grids. *Computers & Fluids*, 98:134–151, 2014.