Splitting schemes for compressible Navier-Stokes equations

by

Roman Frolov

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

Department of Mathematical and Statistical Sciences
University of Alberta

Abstract

The main goal of this work is the development of all-speed numerical methods for the compressible Navier-Stokes equations, i.e. methods that remain efficient in incompressible, weakly compressible, and compressible regimes. To achieve this goal we propose algorithms based on the direction splitting approach in Cartesian and spherical coordinates. First, we consider the case of the Cartesian coordinates and develop a Linearized-Block-Implicit (LBI) scheme suitable for computations of low- and high-Mach number flows. Next, we introduce a second-order direction splitting method for solving the incompressible Navier-Stokes-Boussinesq system in spherical geometries coupled with the artificial compressibility regularization of the incompressible Navier-Stokes system. Finally, we develop a numerical method for nearly incompressible and weakly compressible flows in spherical shells. Numerical experiments confirm that the scheme retains stability and convergence for extremely low values of the Mach number, preserves the incompressibility of the initial data, and has excellent parallel performance. Thus, we hope that it may serve as a foundation for the next generation of dynamical cores for weather and climate models, as well as be useful in other applications.

Preface

The first chapter of this thesis is based on the paper [35]: R. Frolov (2019), “An efficient algorithm for the multicomponent compressible Navier-Stokes equations in low- and high-Mach number regimes.”, Computers & Fluids, Vol. 178, pages 15-40. The second and third chapters are joint work with P. Minev and A. Takhirov.

Acknowledgements

I would like to express gratitude to my supervisor, Professor Peter Minev, for his guidance and support throughout my Ph.D. program. I learned a lot from him about mathematics, research, and life in general. I will never forget our lengthy discussions, brainstorms, and arguments, that helped to shape me as a researcher and as a person. I would like to thank Professor Aziz Takhirov, his contribution to the success of this project is hard to overestimate and our collaboration was both pleasant and productive. I would also like to thank the rest of my supervisory committee: Professor Xinwei Yu, and Professor Morris Flynn. Our meetings helped me to take a fresh look at the problems and challenges I faced during my work.

I would like to thank other professors from the Department of Mathematical and Statistical Sciences, University of Alberta I was fortunate to learn from: Professor Yau Shu Wong, Professor Vakhtang Putkaradze, Professor Volker Runde, and Professor Rouslan Krechetnikov.

I wish to acknowledge the contribution of the funding I obtained from the Department of Mathematical and Statistical Sciences, University of Alberta, the National Science and Engineering Research Council of Canada (NSERC), and the American Chemical Society Petroleum Research Fund. I also want to thank the ComputeCanada organization for providing computing hardware and technical support.

And most importantly, I want to express my gratitude to my family for their love, patience, and for supporting me through the hardest of times: my

wife Nadia, my son Eric, my brother Alexander, my parents Svetlana and Sergey, and my grandparents Vladimir, Galina, Vera, and Valery. Last but not least, I wish to thank my friends for making my life fun and worth living.

Contents

List of Tables

List of Figures

Introduction

Fluid flows are said to be in different compressibility regimes depending on the value of the Mach number (M) – the ratio of the flow velocity to the local speed of sound. When the Mach number is small (typically less than 0.3) compressibility effects are negligible and the flow may be assumed to be incompressible. Mathematically, it corresponds to a singular limit of the governing set of Partial Differential Equations (PDEs) in which the solution depends on two drastically different time-scales. Larger values of the Mach number indicate that compressibility plays a substantial role and cannot be ignored. Furthermore, compressibility mechanisms of different nature may have different effects on the physical process under consideration, e.g. a mechanically incompressible flow may be thermally compressible, which can be interpreted as another singular limit of the governing PDEs. Such limiting cases may allow one to derive a different set of governing equations as an approximation to the general Navier-Stokes (NSEs) model. Since designing numerical algorithms for solving the general compressible NSEs, efficient at low values of the Mach number, is known to be challenging, these sets (e.g. the incompressible Navier-Stokes equations, the Boussinesq approximation) are often used for numerical simulations if it is known that the flow of interest will remain in the corresponding regime. However, some important problems may feature both regions of low and high Mach numbers, transition between different compressibility regimes, or experience weak but not negligible compressibility effects. Examples of such flows include atmospheric and oceanic flows, combustion problems, some important astrophysical phenomena, among others. Hence, there is a need for methods that remain practical at any value of the Mach number. The main focus of this thesis is the development of such all-speed methods, i.e. numerical methods for compressible Navier-Stokes equations that remain efficient in incompressible, weakly compressible, and compressible regimes.

When solving complex initial-boundary value problems for systems of PDEs implicitly, it is often beneficial to split the original problem into a chain of simpler subproblems (see e.g. [124], [101]). These subproblems may be associated with different parts of the computational domain (domain decomposition/regionally additive schemes), different physical processes, or different spatial variables (direction splitting schemes), among other methods. Direction splitting schemes, also known as Alternating Directions Implicit (ADI) algorithms (see [28],[80], [114] for early results in this area, and [101] for a comprehensive review), are particularly appealing for the construction of implicit methods since solutions of one-dimensional problems can be easily achieved on parallel machines using direct solvers with low computational complexity and good scaling properties. This approach had been found successful in numerous applications but was believed to be unsuitable for low-Mach number flows (see e.g. remarks in [121] and [56]). In this work, we demonstrate that direction splitting methods can be successfully adapted for the use in both high- and low-Mach number regimes of compressible NSEs, and propose schemes based on the direction splitting methodology for Cartesian and spherical coordinate systems.

In particular, Chapter 1 of this thesis presents a direction splitting solver with an error reduction technique for the compressible NSEs in Cartesian coordinates. This approach, alongside carefully designed stabilization terms for highly compressible and almost incompressible regimes, allowed maintaining efficiency in a wide range of Mach numbers (as low as $M = 10^{-6}$, and as high as $M = 8.96$). The algorithm has been extended for multi-component flows with interfacial physics. Accuracy, stability, and parallel performance have been rigorously tested in several challenging numerical benchmarks.

Among the most interesting cases of low-Mach number weakly compressible flows are atmospheric and oceanic flows. An efficient and universal method is required here to solve a large-scale problem without significant simplifications of the system of governing equations by removing some of the physical effects from the consideration. This may improve existing dynamical cores of numerical weather prediction and climate simulation models. To simulate these flows using the ADI-type schemes, the direction splitting methodology has to be extended for the case of spherical shell domains. In Chapter 2, we combined the ideas of the direction splitting ([42]), artificial compressibility ([44]), and domain decomposition ([63]) to develop a promising algorithm for the Navier-Stokes-Boussinesq system in spherical shell geometries. We demonstrate theoretically (for the case of a linear parabolic equation) and numerically that the ADI schemes can be reformulated for spherical coordinates without losing their stability and accuracy. The artificial compressibility technique used in

Chapter 2 is inspired by the form of the mass conservation equation in the low Mach number limit of the compressible NSEs, and thus the method can be naturally extended to the compressible case. Such extension is presented in Chapter 3, where ideas of Chapters 1 and 2 are combined to propose a numerical method for weakly compressible flows in spherical geometries, i.e. for solving the compressible NSEs in spherical coordinates at small and moderate values of the Mach number. The method is accurate and preserves the correct scaling of the solution in the incompressible limit (at least for Mach numbers as low as 10^{-6}), and has excellent parallel scalability on large supercomputers.

Chapter 1

A numerical method for the compressible Navier–Stokes equations in low- and high-Mach number regimes

1.1 Introduction

The goal of this chapter is to develop an algorithm for solving the compressible Navier-Stokes equations (compressible NSEs) to be used in various applications, ranging from supersonic shock-interface or shock-boundary layer interactions to nearly incompressible cases, such as highly subsonic astrophysical flows. The compressible NSEs have been shown to converge to the incompressible NSEs as the Mach number (M) approaches zero (at least in some sense under certain conditions, see a review given in Section 1.2.1 for more details).

However, numerical methods designed for highly compressible flows are known to experience severe problems in the incompressible limit (see e.g. [50],[49], main results summarized in Section 1.2.2), such as strict limitations on the time-step (over-resolution in time) due to the dependence of the Courant-Friedrichs-Lewy (CFL) constant on the Mach number, and the need for unrealistically fine meshes (over-resolution in space) due to the incorrect scaling of the artificial viscosity term. A number of techniques have been proposed in the literature to overcome these issues, including but not limited to, preconditioners designed to rescale the artificial dissipation term (see e.g. [122],[121]), and explicit-implicit flux splitting (see e.g. [20],[27],[68], [126]), constructed to allow for an efficient numerical solution of time-dependent problems (see a summary in Section 1.2.2). Another way to construct an all-speed algorithm is to develop an efficient way of solving the fully-implicit Euler system/NSEs. Linearized block implicit (LBI) factored schemes, such as algorithms from [12], [9], [10], combine a linearization technique based on Taylor expansions with an approximate direction splitting factorization strategy. A related scheme is proposed in this chapter for the compressible NSEs based on a factorization strategy from [101], (p. 83). Although it had been reported that implicit algorithms of the ADI-type do not cure the stiffness problem at low Mach numbers (see [121]), Choi and Merkle demonstrated in [17] that the reason for the ineffectiveness of these schemes is the splitting error of the factorization, which dominates the physical fluxes as $M \to 0$. In this study, a reduction strategy, similar to the one used in [69] for the case of the two-dimensional Maxwell equations, is employed to solve the issue. It allows for the use of the method at extremely low Mach numbers without adding extra computational cost, as is demonstrated in Sections 1.4.1 and 1.4.3.

Centered-in-space discretization of physical fluxes in a system of hyperbolic conservation laws requires an introduction of artificial dissipation terms to avoid the high-frequency oscillations due to the odd-even decoupling at low Mach numbers, and spurious oscillations across flow discontinuities for highly compressible flows. In the proposed algorithm an artificial dissipation term similar to the one in [48] is added implicitly to the scheme at high Mach numbers. A nonlinear adaptive choice of artificial viscosity coefficient based on the maximum wave propagation speed in local one-dimensional Riemann problems (see [47] for details) guarantees robust behavior of the method for an arbitrary system of hyperbolic conservation laws, while being less dissipative than classical first-order schemes, such as Lax-Friedrichs (see Section 1.4.2 for details). For the low-Mach number regime this term needs to be rescaled to avoid over-dissipation, similar to other shock-capturing and artificial viscosity schemes (see [121], [87], [86]). Instead of using a preconditioning matrix to perform such rescaling, or introducing a fourth-order difference term, a different approach is proposed in this study. A novel artificial dissipation term is designed based on a second-order finite difference with artificial viscosity proportional to h^2. This difference operator is applied to the conservative variables and the product of conservative variables and the Jacobians. This combination allows one to maintain control of the kinetic energy and dump high-frequency oscillations, and thus to achieve an efficient solution for low-Mach number flows. The effectiveness of the proposed stabilization is shown in Section 1.4.1.

Finally, the algorithm allows for the simulation of multicomponent flows with surface tension. Among conventional methods of simulation of material interfaces are interface-capturing (see e.g. [6],[19],[110],[2],[3],[79],[102]), interface-tracking (see e.g. [15],[109]), and ghost fluid methods (see e.g. [78]). The interface-capturing technique based on the advection of the volume of fluid (VoF) function (similar to [19],[3],[102]) is chosen here alongside the interface sharpening technique from [110] for its numerical efficiency, easiness of implementation in higher dimensions, ability to automatically deal with topological changes, and possible implementations of interfacial physics (see e.g. [96]). Using a diffused interface model and conservative formulation of the governing equations requires special care to avoid spurious pressure oscillations across the interface and generation of artificial acoustic waves in the numerical mixture layer (see e.g. [102],[6]). The consistency of the Equation of State (EOS) in the mixture layer and preservation of contact discontinuity are guaranteed by the appropriate choice of advected flow variables and their incorporation into the computation of Jacobians and the linearization procedure, as well as special treatment of the VoF-advection equation that is consistent with the rest of the governing equations. These results are in line with the explicit case described in [102] and provide an extension of the LBI factored schemes to the multicomponent case. The dissipation terms proposed in this study were also found to preserve the pressure and velocity equilibrium at interfaces.

The rest of this chapter is organized as follows. Section 1.2 reviews the theoretical background and existing literature related to the proposed algorithm and the types of problems it is designed to tackle. Section 1.3 provides a detailed description of the proposed method, including a new version of the LBI factored scheme with splitting error reduction (Section 1.3.1), artificial dissipation terms at high and low Mach numbers (Section 1.3.2), and the extension of the method to the multicomponent case (Section 1.3.3). The summary of the algorithm can be found in Section 1.3.5. Numerical test cases are presented in Section 1.4. Section 1.5 provides some concluding remarks and discussions on the method and possible directions for future studies.

1.2 Theoretical background and literature review.

1.2.1 Low Mach number limit of compressible flows.

The topic of the low Mach number limit of the compressible Navier-Stokes and Euler equations has been extensively studied in recent decades. This section does not attempt to give a comprehensive review of the results and the literature in this area or explain technical details of the methods typically used to analyze the limit and prove existence and convergence results. Instead, it aims to give a brief introduction to the field and present a justification for the expected behavior of all-speed numerical methods for compressible flows at small values of the Mach number. For a more comprehensive review please see [5], [22], [26], [106], among other sources.

In general, the presence of a small parameter in a system of differential equations may indicate the dependence of its solution on different time-scales. The limit as the parameter goes to zero is referred to as a singular limit, since oscillations on the short scale may prevent the convergence of the solution (see [105]). In particular, the limit of the Mach number (ratio of a characteristic velocity of the flow to the sound speed in the fluid) approaching zero is a singular limit that involves two time-scales (see [5]). This limit is fundamental for fluid dynamics since it bridges the gap between the mathematical descriptions of the compressible and incompressible fluid flows ([106]).

Indeed, following the steps outlined in [77], we may consider the non-dimensionalized compressible Navier-Stokes equations in the isentropic regime, written in terms of density and velocity $(\rho, \mathbf{u})$), where μ, ξ, a, and γ are given parameters:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0,\ \ \rho \geq 0, \tag{1.1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \mu \nabla^2 \mathbf{u} - \xi \nabla \nabla \cdot \mathbf{u} + \frac{\nabla p}{M^2} = 0,\ \ \mu \geq 0, \mu + \xi > 0, \tag{1.2}$$

$$p = a\rho^\gamma,\ \ a > 0, \gamma > 1. \tag{1.3}$$

and formally obtain the incompressible Navier-Stokes as the limit of (1.1)-(1.3) when $M \to 0$. The equation of state (1.3) and the momentum equation (1.2) indicate that $\rho = \rho_0 + \mathcal{O}(M^2)$, where ρ_0 can be taken to be equal to 1 without loss of generality. As $M \to 0$, $\rho \to 1$, the mass conservation yields the divergence-free condition on velocity, and the momentum equation can be re-written using $\nabla \rho^\gamma = \nabla(\rho^\gamma - 1)$, transforming the system (1.1)-(1.3) into the incompressible Navier-Stokes system:

$$\nabla \cdot \mathbf{u} = 0, \tag{1.4}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \nabla \pi - \mu \nabla^2 \mathbf{u} = 0, \tag{1.5}$$

where hydrostatic pressure π is the “limit” of renormalized thermodynamic pressure $\frac{p-1}{M^2}$. The details of this limiting process, however, are more involved and may require consideration of additional oscillating terms, depending on what the initial conditions are (see [77] for further details).

The non-isentropic inviscid case is considered in [50] (note that the derivation will be essentially the same for the viscous case). The non-dimensionalized Euler equations can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1.6}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \frac{1}{M^2} \nabla p = 0 \tag{1.7}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{u} + p \mathbf{u}) = 0 \tag{1.8}$$

where E is the total energy, and the non-dimensional form of the ideal gas equation of state is given by:

$$p = (\gamma - 1) \left(\rho E - \frac{M^2}{2} \rho ||\mathbf{u}||^2 \right) \tag{1.9}$$

Next, following [50], we assume the existence of the solution to (1.6)-(1.8) in the form of an asymptotic expansion in powers of the Mach number:

$$\rho = \rho_0 + M\rho_1 + M^2\rho_2 + ... \tag{1.10}$$

$$\mathbf{u} = \mathbf{u}_0 + M\mathbf{u}_1 + M^2\mathbf{u}_2 + ... \tag{1.11}$$

$$p = p_0 + Mp_1 + M^2p_2 + ... \tag{1.12}$$

$$E = E_0 + ME_1 + M^2E_2 + ... \tag{1.13}$$

Substituting the expansions (1.10)-(1.13) into the equations (1.6)-(1.8) and collecting the terms with equal powers of M, we obtain:

- Order $\frac{1}{M^2}$

$$\nabla p_0 = 0, \tag{1.14}$$

- Order $\frac{1}{M}$

$$\nabla p_1 = 0, \tag{1.15}$$

- Order 0

$$\frac{\partial \rho_0}{\partial t} + \nabla \cdot (\rho \mathbf{u}_0) = 0 \tag{1.16}$$

$$\frac{\partial (\rho_0 \mathbf{u}_0)}{\partial t} + \nabla \cdot (\rho \mathbf{u}_0 \otimes \mathbf{u}_0) + \nabla p_2 = 0 \tag{1.17}$$

$$\frac{\partial (\rho_0 \mathbf{u}_0)}{\partial t} + \nabla \cdot (\rho E_0 \mathbf{u}_0 + p_0 \mathbf{u}_0) = 0 \tag{1.18}$$

The 0-order equation of state becomes:

$$p_0 = (\gamma - 1)\rho_0 E_0. \tag{1.19}$$

Thus, from (1.14) and (1.15) we conclude that

$$p(\mathbf{x}, t) = p_0(t) + M^2 p_2(\mathbf{x}, t) \tag{1.20}$$

(i.e. the pressure is constant in space up to fluctuations of order M^2). If no open boundaries are present, the thermodynamic pressure p_0 may be assumed to be constant in time. Next, from (1.19) $\dfrac{\partial \rho_0 E_0}{\partial t} = \nabla(\rho_0 E_0) = 0$, and the energy equation gives the divergence-free condition $\nabla \cdot \mathbf{u}_0 = 0$. Using this condition in the mass conservation equation, (1.16) gives zero material derivative of ρ_0, and the 0-order system reduces to the nonhomogeneous incompressible Euler equations (i.e. incompressible Euler equations for the case of non-constant density):

$$\frac{\partial \rho_0}{\partial t} + \mathbf{u} \cdot \nabla \rho_0 = 0, \tag{1.21}$$

$$\rho_0 \left(\frac{\partial \mathbf{u}_0}{\partial t} + \nabla(\mathbf{u}_0 \otimes \mathbf{u}_0) \right) + \nabla p_2 = 0, \tag{1.22}$$

$$\nabla \cdot \mathbf{u}_0 = 0. \tag{1.23}$$

Thus, as we see from the analysis in [77] and [50], the naive derivation of the incompressible Navier-Stokes equations from their compressible counterparts is relatively straight-forward. However, a rigorous mathematical analysis of $M \to 0$ singular limit is significantly more complicated. Such analysis contains at least two parts (see [85]): an existence and uniform boundedness result time-independent of the Mach number, and a convergence result either to the fixed solution of the limit equations or to a limiting profile. Moreover, the nature of the low Mach number limit and its analysis depends on several factors (see [4], [5], [85] for details), such as whether the flow is isentropic or non-isentropic, the initial data is well-prepared ($\nabla \cdot \mathbf{u}(0), \nabla p(0) = \mathcal{O}(M)$) or general (ill-prepared), the domain is bounded, unbounded or periodic, the fluid is viscous or inviscid, efficient or poor thermal conductor, and whether temperature variations are small or large. Below we give a brief overview of the results in some of these cases.

A general framework of quasilinear hyperbolic systems that depend on a large parameter was used in [66] to study the behavior of singular limits of solutions of nonlinear hyperbolic systems and in particular the behavior of solutions of the compressible isentropic Euler and Navier-Stokes equations. This approach allowed to prove some uniform stability and convergence results for the Euler and Navier-Stokes equations independent on viscosity using balanced energy estimates (under certain assumptions, e.g. well-prepared, i.e. incompressible or almost incompressible, initial data). The results from [66] were further extended in [67]. In particular, assuming global existence of the solution of the Euler equations for nearby initial data and a uniformly valid asymptotic expansion on bounded time intervals of the velocity as $M \to 0$, more general uniform stability estimates independent of M were obtained, as well as long-term existence for large data solutions of compressible fluid equations as $M \to 0$. While [66] and [67] considered the isentropic case in the whole space or periodic domains, a short-time existence result was proven in [104] for the non-isentropic compressible Euler equations in a bounded domain, as well as the corresponding convergence results. The case of the isentropic Euler equations with incompressible initial data in an exterior domain was also covered in [57] and [58]. The analysis performed in [53], where global in time uniform convergence of the solutions of the isentropic compressible Navier-Stokes system to the incompressible solutions was established, under the assumption of the existence of a reasonable smooth incompressible solution, reveals that the essential mechanism in this limit process is the hyperbolic effect that drives density to a constant as the Mach number decreases.

As shown in [123], when the incompressibility assumption of the initial data is relaxed (i.e. the case of the ill-prepared initial data is considered), it is still possible to prove the uniform convergence to the incompressible limit in the case of the isentropic Euler equations in the whole space domain, but the uniformity breaks near $t = 0$ due to the formation of an initial layer. In this case, the resulting velocity consists of the incompressible limit flow, and a highly-oscillatory term associated with the sound waves (see [5]). This highly-oscillatory term can be described by the wave equation, at least locally in time (see e.g. [70]). The analysis of the low Mach number limit with ill-prepared initial data is significantly more complicated in the viscous case, since an additional pressure term is generated in the incompressible limit (see a discussion in [26] for details). Furthermore, boundary conditions and the type of the domain have a significant effect on the limit in the case of ill-prepared initial data. For example, when Dirichlet boundary conditions are imposed in a generic bounded domain, the acoustic waves are instantaneously dumped due to a formation of a boundary layer of size $\sqrt{M}$ ([24],[25]). Convergence results for the case of a periodic domain can be found in [21], for the case of exterior domains see [29].

Consideration of the non-isentropic case with general initial data adds another layer of complexity to the analysis, in particular, due to stronger nonlinearity (the matrix multiplying the time derivative depends strongly on the dependent variables - see [85] for details). The inviscid case was treated in [85], where the existence of classical solutions on a time interval independent of M was proven. The rigorous analysis of the full Navier-Stokes case (including effects of large temperature variations and thermal conduction) with some existence and convergence results can be found in [4], and [5]. The inclusion of entropy into consideration helps to reveal two distinct routes to incompressibility: the low Mach number limit yields the incompressible Navier-Stokes when entropy variations are small, and nonhomogeneous Navier-Stokes equations when they are large (see [26]).

In addition to that, the effects of rotation on the low Mach number limit were considered in [38]. Thus, despite that some questions remain open (e.g. convergence for data with critical regularity, the case of bounded domains with more general geometries, and the detailed investigation of the limit for the full Navier-Stokes system - see a discussion in [22]), the convergence of a compressible solution to the corresponding incompressible limit is well-justified under a wide range of conditions, and it is reasonable to expect a well-designed numerical scheme to maintain its accuracy and efficiency as the Mach number goes to zero, at least for well-prepared initial data.

1.2.2 All-speed numerical methods.

The purpose of this section is to provide more details on the typical issues that prevent satisfactory performance of many compressible methods as $M \to 0$ and give examples of some existing strategies of constructing all-speed methods.

Most methods designed for compressible flows employ explicit time discretizations, and thus are subject to the CFL stability condition that reads as $\tau \leq \frac{h}{|\lambda_{max}|}$, where τ is the time-step, h is the space-step, and λ_{max} is the fastest characteristic wave speed, which can be written in terms of flow speed u and sound velocity c as $\lambda_{max} = u \pm c$ (see [20]). In the non-dimensional form this condition becomes

$$\tilde{\tau} \leq M \frac{h}{\max|M\tilde{u} \pm \tilde{c}|}, \tag{1.24}$$

where the tildes denote scaled quantaties (see [20] for details). It is clear from (1.24) that the time-step restriction becomes more and more severe as M decreases, leading to larger computational time required (over-resolution in time). If a flow is time-independent, the number of iterations required to reach the steady state increases significantly with the decrease of M for the same reason.

Another common obstacle the compressible solvers face when dealing with low-Mach number flows is a poor scaling of the artificial dissipation terms used to stabilize the solution at high Mach numbers. For example, the analysis of the Roe scheme in [87], which is also applicable for the general Godunov-type numerical flux functions, reveals that while in the supersonic case the scaling of the Roe (stabilized) flux is the same as the scaling of the central (non-stabilized) flux, it becomes inconsistent with the central flux in the subsonic case. The wrong scaling of the artificial dissipation terms significantly decreases the accuracy of the computations, and the use of unrealistically fine meshes may be required to produce acceptable results (over-resolution in space). Furthermore, upwind and Godunov-type schemes were found to generate pressure fluctuations of the order of $\mathcal{O}(M)$, instead of $\mathcal{O}(M^2)$ that is expected from the asymptotic analysis of the low Mach number limit (see [50] and [49]). Such behavior is fundamentally inherent to the Godunov solution strategy that assumes jumps in the flow variables at the cell interfaces to define Riemann problems there (see e.g. [87], [49]).

A common strategy to improve the performance of compressible solvers at low Mach numbers is preconditioning (see e.g. [121], [122],[87],[50], among other sources). In the steady case, a preconditioner may be used to alter the time dependency of the equations to equalize the eigenvalues of the Jacobians, and thus to cure the stiffness problem that leads to the slow convergence to the steady-state, as well as to rescale the artificial dissipation term, as it is done in [122]. If this approach is used, only the steady-state operator is consistent with the original physical problem, and thus the technique cannot be used for time-dependent flows. It means that while a suitable preconditioner may resolve both over-resolution in space and over-resolution in time (i.e. low rate of convergence) problems for a steady problem, some implicitness of the time-discretization strategy is required to achieve the same in the case of a time-dependent flow. Another possible use of a preconditioner for a reformulation of compressible methods for nearly incompressible flows can be found in [49], where the fluxes computed by Godunov-type schemes are modified by solving a preconditioned Riemann problem instead of the original one.

If one wishes to avoid the solution of the full nonlinear Navier-Stokes or Euler system, an explicit-implicit flux splitting strategy may be developed (see e.g. [20], [27], [68]). The Asymptotic-Preserving (AP) methodology is often used to design such schemes (see [20], [27], and references there). Similar to the continuous compressible model that approaches the incompressible model as $M \rightarrow 0$, a discrete compressible model may be designed to produce a consistent discretization of the incompressible equations in the limit of zero Mach number, with an additional requirement that its stability condition is independent of M. This guarantees that the scheme is consistent with both compressible and incompressible regimes and can be used for computing mixed flows where a part of the flow has local Mach number of the order of unity, and a part of the flow has very small local Mach number ([20]).

The choice of the implicit-explicit splitting is not unique. In [36] two different approaches were proposed: one based on the splitting of the slow and fast eigenvalues of the Jacobian matrix of the fluxes and treating only the “fast” part implicitly, and one based on separating the velocity terms and the pressure terms and using a fractional step method. Other methods focus on formulating a pressure equation, e.g. in [20] a suitable elliptic pressure equation is derived from a semi-implicit methodology where the pressure terms are treated implicitly while the other terms are treated explicitly. Related approaches can be found in [68], [27] and references there. Although very successful, the implicit-explicit flux splitting approach is not without drawbacks. The stability of several implicit-explicit schemes was analyses in [126], and among other results a small instability region for the flux splitting from [68] was found, if this splitting is combined with the IMEX scheme from [90]. Thus, one should be careful when using and implementing these methods, and their properties should be carefully analyzed.

1.2.3 Hyperbolic conservation laws: Riemann problems and invariant domains.

The theory of hyperbolic conservation laws and Riemann problems, and related notions of invariant sets and invariant domains, are in the core of the high-Mach number stabilization strategy used in this chapter. This subsection briefly introduces definitions and important properties relevant to the method described in Section 1.3.2. More detailed and thorough discussions may be found in [48], [45] and other works referenced there.

Hyperbolic conservation laws are essential for modeling fluid flows since the Navier-Stokes system reduces to a system of hyperbolic conservation laws (Euler equations) when viscous and heat conduction effects are neglected, and hyperbolic terms of the Navier-Stokes system are responsible for the most stringent requirements on the numerical methods ([120], Chapter 2). We will not repeat here the basic theory of hyperbolic partial differential equations, assuming that the reader is familiar with it (a good introduction may be found in [120], Chapter 2), limiting the discussion to only a few basic facts directly relevant to the discussion below.

First, following [120], we define hyperbolic systems of partial differential equations and systems of conservation laws.

Definition 1.2.1 *(from [120], Chapter 2)*
A system of partial differential equations

$$\boldsymbol{U}_t + \boldsymbol{A}\,\boldsymbol{U}_x + \boldsymbol{B} = \boldsymbol{0}$$

where $\boldsymbol{U}$ *and* $\boldsymbol{B}$ *are* m*-dimensional vectors and* $\boldsymbol{A}$ *is an* $m \times m$ *matrix, is said to be hyperbolic at a point* (x, t) *if* $\boldsymbol{A}$ *has* m *real eigenvalues and a corresponding set of* m *linearly independent (right) eigenvectors. The system is said to be strictly hyperbolic if the eigenvalues are all distinct.*

Definition 1.2.2 *(from [120], Chapter 2)*
Conservation laws are systems of partial differential equations that can be written in the form

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = \boldsymbol{0} \tag{1.25}$$

where $\boldsymbol{U}$ *is an m-dimensional vector of conserved variables,* $\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{U})$ *is the vector of fluxes.*

Thus, a system of conservation laws is hyperbolic, if its Jacobian $\mathbf{A}(\mathbf{U}) = \dfrac{\partial \mathbf{F}}{\partial \mathbf{U}}$ has real eigenvalues $\{\lambda_i\}$ and a complete set of linearly independent eigenvectors $\{\mathbf{K}_i\}$. $\lambda_i(\mathbf{U})$ is also called a characteristics speed.

Definition 1.2.3 *(from [120], Chapter 2)*

A λ_i*-characteristic field is said to be linearly degenerate if*

$$\nabla\lambda_i(\boldsymbol{U}) \cdot \boldsymbol{K}_i(\boldsymbol{U}) = 0, \forall \boldsymbol{U} \in \mathbb{R}^m$$

.

Definition 1.2.4 *(from [120], Chapter 2)*

A λ_i*-characteristic field is said to be genuinely nonlinear if*

$$\nabla\lambda_i(\boldsymbol{U}) \cdot \boldsymbol{K}_i(\boldsymbol{U}) \neq 0, \forall \boldsymbol{U} \in \mathbb{R}^m$$

The precise meaning of the solution of (1.25) for general initial data is either very technical, or an open problem ([48]), and we concentrate here on the solutions of one-dimensional Riemann problems formulated for the systems of hyperbolic conservation laws:

Definition 1.2.5 *(from [120], Chapter 2)*

The Riemann problem for a general $m \times m$ *nonlinear hyperbolic system with initial data* $\boldsymbol{U}_L$*,* $\boldsymbol{U}_R$ *is the following initial-value problem:*

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = \boldsymbol{0} \tag{1.26}$$

$$\boldsymbol{U}(x,0) = \begin{cases} \boldsymbol{U}_L \;\; \textit{if}\; x < 0, \\ \\ \boldsymbol{U}_R \;\; \textit{if}\; x > 0 \end{cases} \tag{1.27}$$

The structure of the similarity solution of (1.26)-(1.27) is described in [120], Chapter 2. It consists of $m + 1$ constant states separated by m waves, which can be shock waves, contact waves, or rarefactions ([120], Chapter 2). The type of waves present depends on the closure of the system (i.e. the equation of state). If we assume that the constant states $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected by a single wave, the following situations are possible depending on the type of this wave:

- **Shock wave:** $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a jump discontinuity of speed S_i in a genuinely nonlinear field i and the following conditions are satisfied:

  - the Rankine-Hugoniot conditions

  $$\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_R) = S_i(\mathbf{U}_R - \mathbf{U}_L)$$

  - the entropy condition

  $$\lambda_i(\mathbf{U}_L) > S_i > \lambda_i(\mathbf{U}_R)$$

- **Contact wave:** $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a jump discontinuity of speed S_i in a linearly degenerate field i and the following conditions are satisfied:

- the Rankine-Hugoniot conditions

$$\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_R) = S_i(\mathbf{U}_R - \mathbf{U}_L)$$

- constancy of the generalized Riemann Invariants across the wave

$$\frac{du_1}{k_1^{(i)}} = \frac{du_2}{k_2^{(i)}} = ... = \frac{du_m}{k_m^{(i)}}$$

where $\mathbf{U} = (u_1, u_2, ..., u_m)$, $\mathbf{K}_i = (k_1^{(i)}, k_2^{(i)}, ..., k_m^{(i)})$, and du_i is the ratio of change of quantaty u_i across the wave structure.

- the parallel characteristic condition

$$\lambda_i(\mathbf{U}_L) = \lambda_i(\mathbf{U}_R) = S_i$$

- **Rarefaction wave:** $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a smooth transition in a genuinely nonlinear field i and the following conditions are satisfied:

- constancy of the generalized Riemann Invariants across the wave

$$\frac{du_1}{k_1^{(i)}} = \frac{du_2}{k_2^{(i)}} = ... = \frac{du_m}{k_m^{(i)}}$$

where $\mathbf{U} = (u_1, u_2, ..., u_m)$, $\mathbf{K}_i = (k_1^{(i)}, k_2^{(i)}, ..., k_m^{(i)})$, and du_i is the ratio of change of quantaty u_i across the wave structure.

– divergence of characteristics

$$\lambda_i(\mathbf{U}_L) < \lambda_i(\mathbf{U}_R)$$

Following [48], we assume that there exists a nonempty admissible set $\mathbb{A} \subset \mathbb{R}^m$ such that for any pair of states $(\mathbf{U}_L, \mathbf{U}_R) \in \mathbb{A} \times \mathbb{A}$ and any unit vector $\mathbf{n} \in \mathbb{R}^d$, the following one-dimensional Riemann problem

$$\mathbf{U}_t + \partial_x \left(\mathbf{F}(\mathbf{U})\mathbf{n}\right) = \mathbf{0}, \ (x,t) \in \mathbb{R} \times \mathbb{R}_+, \tag{1.28}$$

$$\mathbf{U}(x,0) = \begin{cases} \mathbf{U}_L \text{ if } x < 0, \\ \mathbf{U}_R \text{ if } x > 0 \end{cases} \tag{1.29}$$

has a unique entropy satisfying self-similar solution $\mathbf{U}(\mathbf{n}, \mathbf{U}_L, \mathbf{U}_R, \xi)$, where $\xi = \dfrac{x}{t}$. An important result for the formulation of the stabilization method for the high-Mach number regime is the existence of a maximum wave speed, denoted as $\lambda_{max}(\mathbf{n}, \mathbf{U}_L, \mathbf{U}_R)$, such that

$$\mathbf{U}(\mathbf{n}, \mathbf{U}_L, \mathbf{U}_R, \xi) = \mathbf{U}_L \text{ if } \xi \leq -\lambda_{max}(\mathbf{n}, \mathbf{U}_L, \mathbf{U}_R)$$

and

$$\mathbf{U}(\mathbf{n}, \mathbf{U}_L, \mathbf{U}_R, \xi) = \mathbf{U}_R \text{ if } \xi \geq \lambda_{max}(\mathbf{n}, \mathbf{U}_L, \mathbf{U}_R)$$

(see [48], [45], [47] for details).

When the notion of a Riemann problem is clearly defined, invariants sets can be introduced as:

Definition 1.2.6 *(from [48])*

We say a set $\mathbb{B} \subset \mathbb{A} \subset \mathbb{R}^m$ is invariant for

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = \boldsymbol{S}(\boldsymbol{U}), \text{ for } (\boldsymbol{x}, t) \in \mathbb{R}^d \times \mathbb{R}_+,$$

$$\boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}_0(\boldsymbol{x}), \text{ for } \boldsymbol{x} \in \mathbb{R}^d.$$

if $\mathbb{B}$ is convex and for any pair $(\boldsymbol{U}_L, \boldsymbol{U}_R) \in \mathbb{B} \times \mathbb{B}$, any unit vector $\boldsymbol{n} \in \mathbb{R}^d$, and any $t > 0$ such that $t\lambda_{max}(\boldsymbol{n}, \boldsymbol{U}_L, \boldsymbol{U}_R) \leq \frac{1}{2}$, the average of the entropy solution of the Riemann problem (1.28)-(1.29) over the Riemann fan, say $\tilde{\boldsymbol{U}}(t, \boldsymbol{n}, \boldsymbol{U}_L, \boldsymbol{U}_R)$, remain in $\mathbb{B}$, and if there exists $\tau_0 > 0$ such that for any $\boldsymbol{U} \in \mathbb{B}$ and any $\tau \leq \tau_0$ the quantity $\boldsymbol{U} + \tau \boldsymbol{S}(\boldsymbol{U})$ is in $\mathbb{B}$.

Now, if one considers an approximation process $\mathbf{R}_h$, the notion of invariant domains can be defined as (see [48] for a more rigorous explanation):

Definition 1.2.7 *(from [48])*
A convex invariant set $\mathbb{B} \subset \mathbb{A} \subset \mathbb{R}^m$ is said to be an invariant domain for $\boldsymbol{R}_h$ if and only if for any state $\boldsymbol{U}$ in $\mathbb{B}$, the state $\boldsymbol{R}_h(\boldsymbol{U})$ is also in $\mathbb{B}$.

The notions of the invariant sets and invariant domains are extensions of the maximum principle for scalar conservation laws to the case of nonlinear hyperbolic systems, and thus the preservation of the invariant domains is an important characteristic of a numerical method. Such preservation is achieved in [48], [45] by using λ_{max} to introduce artificial dissipation terms to a finite elements numerical scheme. This technique was a foundation for the finite-difference stabilization proposed in Section 1.3.2.

1.3 Numerical algorithm.

1.3.1 Linearized block ADI method and error reduction.

Governing Equations and Linearization.

In order to avoid numerical errors introduced by non-conservative schemes in the presence of shock waves (see [54],[64]), we consider the compressible Navier-Stokes equations in conservative form. We demonstrate the ideas on the 2D version of the equations but the proposed schemes extend easily to three dimensions. For the single-component case, the system can be written as (see [10]):

$$\begin{aligned}\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}+\frac{\partial \mathbf{G}(\mathbf{U})}{\partial y}=\frac{\partial \mathbf{V}_1(\mathbf{U},\mathbf{U}_x)}{\partial x}+\frac{\partial \mathbf{V}_2(\mathbf{U},\mathbf{U}_y)}{\partial x}\\ +\frac{\partial \mathbf{W}_1(\mathbf{U},\mathbf{U}_x)}{\partial y}+\frac{\partial \mathbf{W}_2(\mathbf{U},\mathbf{U}_y)}{\partial y},\end{aligned} \tag{1.1}$$

where $\mathbf{U}=(\rho,m,n,E)$ is the vector of conservative variables (density, momentum in x and y direction, and total energy), $\mathbf{F}$ and $\mathbf{G}$ are the fluxes associated with the Euler system, and $\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{W}_1$, $\mathbf{W}_2$ are the fluxes associated with the viscous stress tensor (see Appendix A for details).

Using the mixed implicit-explicit Euler time discretization, one can write a semi-discrete version of (1.1) as:

$$\begin{aligned}\frac{\mathbf{U}^{n+1}-\mathbf{U}^n}{\tau}+\frac{\partial \mathbf{F}^{n+1}(\mathbf{U})}{\partial x}+\frac{\partial \mathbf{G}^{n+1}(\mathbf{U})}{\partial y}=\frac{\partial \mathbf{V}_1^{n+1}(\mathbf{U},\mathbf{U}_x)}{\partial x}+\frac{\partial \mathbf{V}_2^{n}(\mathbf{U},\mathbf{U}_y)}{\partial x}\\ +\frac{\partial \mathbf{W}_1^{n}(\mathbf{U},\mathbf{U}_x)}{\partial y}+\frac{\partial \mathbf{W}_2^{n+1}(\mathbf{U},\mathbf{U}_y)}{\partial y},\end{aligned} \tag{1.2}$$

where τ is the time step. Due to the nonlinearity of the fluxes $\mathbf{F}$, $\mathbf{G}$, $\mathbf{V}_1$, and $\mathbf{W}_2$, the space-discretization of (1.2) produces a nonlinear system of algebraic equations. To approximate the solution of the nonlinear system, a simple linearization, similar to the one in [12], can be employed:

$$\mathbf{F}^{n+1} = \mathbf{F}^n + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) \tag{1.3}$$

$$\mathbf{G}^{n+1} = \mathbf{G}^n + \left(\frac{\partial \mathbf{G}}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) \tag{1.4}$$

$$\mathbf{V}_1^{n+1} = \mathbf{V}_1^n + \left(\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) + \left(\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}_x}\right)^n (\mathbf{U}_x^{n+1} - \mathbf{U}_x^n) \tag{1.5}$$

$$\mathbf{W}_2^{n+1} = \mathbf{W}_2^n + \left(\frac{\partial \mathbf{W}_2}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) + \left(\frac{\partial \mathbf{W}_2}{\partial \mathbf{U}_y}\right)^n (\mathbf{U}_y^{n+1} - \mathbf{U}_y^n). \tag{1.6}$$

Substituting expressions (1.3)-(1.6) into (1.2) and combining explicit terms (denoted as $\mathbf{R}^n$) and implicit terms in corresponding directions (denoted as $\mathbf{A}_x\mathbf{U}^{n+1}$ and $\mathbf{A}_y\mathbf{U}^{n+1}$), equation (1.2) can be written as:

$$\left(\mathbf{I} + \tau\mathbf{A}_x + \tau\mathbf{A}_y\right)\mathbf{U}^{n+1} = \tau\mathbf{R}^n. \tag{1.7}$$

Folowing the approach in [101] (p.83), equation (1.7) can be approximated by the following factorized equation:

$$(\mathbf{I} + \tau\mathbf{A}_x)(\mathbf{I} + \tau\mathbf{A}_y)\,\mathbf{U}^{n+1} = \tau\mathbf{R}^n, \tag{1.8}$$

with splitting error $\mathbf{ER}(\mathbf{U}_k^{n+1}) = \tau^2\mathbf{A}_x\mathbf{A}_y\mathbf{U}^{n+1}$, and solved as a sequence of two one-dimensional problems

$$(\mathbf{I} + \mathbf{A}_x)\hat{\mathbf{U}}^{n+1} = \tau \mathbf{R}^n \tag{1.9}$$

$$(\mathbf{I} + \mathbf{A}_y)\mathbf{U}^{n+1} = \hat{\mathbf{U}}^{n+1}. \tag{1.10}$$

Each of these problems requires the solution of block-tridiagonal linear systems only, which can be performed by a block-tridiagonal extension of the Thomas algorithm for tridiagonal systems (e.g. see [31], Volume 1, pp.188-189). The parallel implementation of the Thomas algorithm using the Schur complement technique and domain decomposition, as described in [42], can be easily extended for the block-tridiagonal version of the linear solver. Weak scalability results for this method can be found in Section 1.4.4.

Remark 1.3.1 *The method can be reformulated for different, more accurate time-marching schemes and different splitting strategies to improve accuracy. Here the implicit Euler method and the splitting from [101], (p.83) are chosen for their simplicity and robustness. In particular, this splitting allows for a multicomponent factorization of non-commutative operators. Therefore, an extension of the method to the three-dimensional case is possible. However, more accurate second-order splitting developed in Chapters 2 and 3 can also be employed here.*

Splitting error reduction.

The splitting error introduced by the factorization (1.8) has been found to grow as the Mach number approaches zero, while in one dimension the LBI factored schemes demonstrate a performance similar to the artificial compressibility method (see [17] for details). There are several possible ways of reducing the error (see [13] for a review). One of the possibilities is to perform the following iterations (here the subscript denotes the iteration level):

$$(\mathbf{I} + \mathbf{A}_x)\hat{\mathbf{U}}^{n+1}_{k+1} = \tau\mathbf{R}^n + \mathbf{E}\mathbf{R}(\mathbf{U}^{n+1}_k) \tag{1.11}$$

$$(\mathbf{I} + \mathbf{A}_y)\mathbf{U}^{n+1}_{k+1} = \hat{\mathbf{U}}^{n+1}_{k+1}, \tag{1.12}$$

with $\mathbf{U}^{n+1}_0 = \mathbf{U}^n$. This reduction strategy is similar to the one used in [69] in the context of the two-dimensional Maxwell equations. None of the test cases presented in this chapter required more than one iteration, i.e. simple addition of $\mathbf{E}\mathbf{R}(\mathbf{U}^n)$ to the right-hand-side of the system provided sufficient reduction of the splitting error for the tests with low-Mach number flows. Hence, at every time step the following system was solved:

$$(\mathbf{I} + \mathbf{A}_x)\hat{\mathbf{U}}^{n+1} = \tau\mathbf{R}^n + \mathbf{E}\mathbf{R}(\mathbf{U}^n) \tag{1.13}$$

$$(\mathbf{I} + \mathbf{A}_y)\mathbf{U}^{n+1} = \hat{\mathbf{U}}^{n+1}. \tag{1.14}$$

Note that the proposed technique is equivalent to the preconditioned Richardson iterative method (see e.g. [100], Chapter 6) applied to the linearized system (1.7). Indeed, if

$$\mathbf{A} = \mathbf{I} + \tau\mathbf{A}_x + \tau\mathbf{A}_y \tag{1.15}$$

a preconditioner can be defined as:

$$\mathbf{P}^{-1} = \left((\mathbf{I} + \tau \mathbf{A}_x)(\mathbf{I} + \tau \mathbf{A}_y)\right)^{-1}. \tag{1.16}$$

Then $\mathbf{A} = \mathbf{P} - \mathbf{ER}$ where $\mathbf{ER} = \tau^2 \mathbf{A}_x \mathbf{A}_y$. Then, the system (1.11)-(1.12) is equivalent to

$$\mathbf{P}\mathbf{U}_{k+1}^{n+1} = \mathbf{ER}\mathbf{U}_{k}^{n+1} + \tau \mathbf{R}^n \tag{1.17}$$

or alternatively

$$\mathbf{U}_{k+1}^{n+1} = \mathbf{U}_{k}^{n+1} + \mathbf{P}^{-1}\mathbf{r}_k \tag{1.18}$$

$$\mathbf{r}^k = \tau \mathbf{R}^n - \mathbf{A}\mathbf{U}_{k}^{n+1}. \tag{1.19}$$

With this preconditioner the iterative method converges very rapidly.

1.3.2 Stabilization.

Guermond-Popov (GP) artificial viscosity scheme.

In the supersonic regions of the flow, we need to introduce stabilization terms to dissipate the high-frequency oscillations associated with higher-order spatial discretizations. We use the idea behind the finite element stabilization method for general hyperbolic systems, proposed in [48] (see [46] for the PDE-version of the Guermond-Popov invariant-domain preserving regularization). It can be reformulated in terms of finite differences as follows:

$$\frac{\mathbf{U}_{i,j}^{n+1} - \mathbf{U}_{i,j}^{n}}{\tau} + \frac{\hat{\mathbf{F}}_{i+\frac{1}{2},j}^{n} - \hat{\mathbf{F}}_{i-\frac{1}{2},j}^{n}}{h_x} + \frac{\hat{\mathbf{G}}_{i,j+\frac{1}{2}}^{n} - \hat{\mathbf{G}}_{i,j-\frac{1}{2}}^{n}}{h_y} = 0 \tag{1.20}$$

with numerical numerical fluxes:

$$\hat{\mathbf{F}}^n_{i+\frac{1}{2},j} = \frac{\mathbf{F}^n_{i+1,j} + \mathbf{F}^n_{i,j}}{2} - \lambda_{i+1,j}\left(\mathbf{U}^n_{i+1,j} - \mathbf{U}^n_{i,j}\right). \qquad (1.21)$$

Artificial viscosity coefficients are defined here as:

$$\lambda_{i+1,j} = \lambda_{max}(\mathbf{U}^n_{i+1,j}, \mathbf{U}^n_{i,j}), \qquad (1.22)$$

with $\lambda_{max}(\mathbf{U}_l, \mathbf{U}_r)$ being a maximum wave speed estimation for the corresponding local one-dimensional Riemann problem with initial conditions given by $\mathbf{U}_l$ and $\mathbf{U}_r$ (see [48],[47] for details).

The original algorithm is fully explicit and preserves all the convex invariant sets, which, in the case of the Euler system, guarantees positivity of density and internal energy, and produces a solution that satisfies the entropy inequality for every entropy pair of a hyperbolic system (see [48] for details). These properties are essential for obtaining an approximation of the physical solution to the Euler equations and are achieved without any extra constructions (such as flux limiters or non-oscillatory reconstructions). Unlike Godunov-type schemes, the method does not require an exact or approximate solution of local Riemann problems. Only an estimation of the maximum speed of wave propagation is needed. The efficiency of the algorithm depends on the particular method used to obtain the estimation. The estimation procedure proposed in this study can be found in Section 1.3.4. The procedure is developed for the case of multiple components obeying Stiffened Gas EOS and includes the effects of surface tension.

Remark 1.3.2 *The original finite element first-order method has been employed to construct a second-order invariant domain preserving approximation – see [45]. There invariant domain preserving auxiliary solutions obtained from the first-order method are used to define local bounds for a high-order, invariant domain violating but an entropy-consistent algorithm, via a convex limiting process.*

Remark 1.3.3 *If $\lambda_{max}(\mathbf{U}_l, \mathbf{U}_r) = \frac{h}{4\tau}$, the finite difference version of the method reproduces the classical Lax-Friedrichs scheme. Due to the nonlinear definition of λ_{max} through the maximum wave speed of the local one-dimensional Riemann problems, the GP-method achieves the same level of robustness as the Lax-Friedrichs scheme while maintaining sharper flow discontinuities (see Section 1.4.2 for comparison).*

As can easily be seen by substituting expression (1.21) into equation (1.20), the method adds a second-order artificial dissipation term of the type (GP-dissipation term):

$$\mathbf{D}^{GP}\mathbf{U} = h_x \frac{\partial}{\partial x}\left[\lambda_{max}\frac{\partial}{\partial x}\mathbf{U}^{n+1}\right] + h_y \frac{\partial}{\partial y}\left[\lambda_{max}\frac{\partial}{\partial y}\mathbf{U}^{n+1}\right] \tag{1.23}$$

to the original explicit centered-in-space scheme. The same second-order stabilizing term can be added to the implicit scheme for the compressible NSEs in the high-Mach number regime, and this is equivalent to defining the inviscid fluxes as:

$$\hat{\mathbf{F}}^{n+1}_{i+\frac{1}{2},j} = \frac{\mathbf{F}^{n+1}_{i+1,j} + \mathbf{F}^{n+1}_{i,j}}{2} - \lambda_{i+1,j}\left(\mathbf{U}^{n+1}_{i+1,j} - \mathbf{U}^{n+1}_{i,j}\right), \tag{1.24}$$

and applying the same linearization and factorization procedures after that. This implicit treatment of the dissipation term does not disrupt the block-tridiagonal nature of the resulting linear system and therefore does not increase the cost of the computations.

Damping of high-frequency oscillations.

Like other artificial dissipation terms designed for stabilization around strong shocks, $\mathbf{D}^{GP}U$ starts dominating the physical solution as the Mach number approaches zero. Indeed, in the incompressible limit (where the velocity of the flow is small comparing to the speed of sound, $|u| \ll c$), the maximum wave speed $\lambda = \lambda_{max} = u \pm c \sim c$. The perturbed Euler system in one dimension can be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) - h\frac{\partial}{\partial x}\left(\lambda \frac{\partial}{\partial x}\rho\right) = 0 \tag{1.25}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}\left(\rho u^2 + p\right) - h\frac{\partial}{\partial x}\left(\lambda \frac{\partial}{\partial x}(\rho u)\right) = 0 \tag{1.26}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}((E+p)u) - h\frac{\partial}{\partial x}\left(\lambda \frac{\partial}{\partial x}E\right) = 0. \tag{1.27}$$

Then, introducing the characteristic scales: length L, density $\tilde{\rho}$, and velocity $\tilde{U}$, dimensionless variables can be defined as $x^* = \frac{x}{L}$, $\rho^* = \frac{\rho}{\tilde{\rho}}$, $u^* = \frac{u}{\tilde{U}}$, $t^* = \frac{t}{\tilde{U}/L}$, $M = \frac{\tilde{U}}{c}$, $p^* = \frac{p}{\tilde{\rho}c^2}$, and $E^* = \frac{E}{\tilde{\rho}c^2}$.

Since $\lambda \sim c$, the system can be rewritten in the following non-dimensional form:

$$\frac{\partial \rho^*}{\partial t^*} + \frac{\partial}{\partial x^*}(\rho^* u^*) - \frac{h}{M}\frac{\partial^2}{(\partial x^*)^2}\rho^* = 0 \tag{1.28}$$

$$\frac{\partial \rho^* u^*}{\partial t^*} + \frac{\partial}{\partial x^*}\left(\rho^* (u^*)^2 + \frac{p^*}{M^2}\right) - \frac{h}{M}\frac{\partial^2}{(\partial x^*)^2}(\rho^* u^*) = 0 \tag{1.29}$$

$$\frac{\partial E^*}{\partial t^*} + \frac{\partial}{\partial x^*}((E^* + p^*)u^*) - \frac{h}{M}\frac{\partial^2}{(\partial x^*)^2}E^* = 0. \tag{1.30}$$

Hence, the artificial dissipation term indeed becomes dominant as $M \to 0$ if h is fixed.

At low Mach numbers, the following dissipation term is proposed in this chapter (Low-Mach (LM-) dissipation term):

$$\begin{gathered}\mathbf{D}^{LM}\mathbf{U}^{n+1} = \mathbf{D}_1^{LM}\mathbf{U}^{n+1} + \mathbf{D}_2^{LM}\mathbf{U}^{n+1} = \\ h^2\omega_1\nabla^2\mathbf{U}^{n+1} + h^2\omega_2\left(\frac{\partial^2}{\partial x^2}\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1}\right] + \frac{\partial^2}{\partial y^2}\left[\left(\frac{\partial \mathbf{G}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1}\right]\right),\end{gathered} \tag{1.31}$$

where ω_1 and ω_2 are scalar dimensionless parameters.

The purpose of the first term here (which is the same as the GP-term with $\lambda \equiv h$) is to drain off the kinetic energy and thus maintain the overall stability of computations. The second and the third terms are used to regularize the entries of the corresponding matrix operator and connect odd and even nodes, which are decoupled due to the use of the centered-in-space discretization for hyperbolic fluxes. Jacobians provide proper weights for every variable depending on the magnitude of its contribution to the linearized flux. The effects of both of these terms at different values of the Mach number are illustrated by numerical examples in Section 1.4.1.

Remark 1.3.4 *A related idea of matching the scaling of the artificial viscosity matrix with the scaling of the Jacobians to stabilize the system in the low-Mach number limit was used in [87] in the context of the Roe-Turkel scheme. However, here instead of choosing scaling parameters to match orders of magnitude of the Jacobian's components, these components themselves are used in the stabilizing term. This was found to be more efficient in the context of the present scheme. Furthermore, as is revealed in Section 1.3.3, this artificial dissipation term does not disrupt the velocity and pressure equilibrium at interfaces and is therefore compatible with the Volume of Fluid method.*

As for the GP-dissipation term, the LM-term can be straightforwardly incorporated into the factorization strategy with direction splitting. Since the splitting error due to the LM-term involves the components of the Jacobian, it should be taken into account while performing the reduction strategy described above.

If the flow under consideration features regions of low and high Mach numbers, switching between LM- and GP-dissipation terms can be performed based on the local Mach number M_{loc} as follows:

$$\mathbf{D}\mathbf{U}^{n+1} = k^{(HM)}\mathbf{D}^{GP}\mathbf{U}^n + k^{(LM)}\mathbf{D}^{LM}\mathbf{U}^n, \tag{1.32}$$

where $k^{(HM)}$, $k^{(LM)}$ are scalar dimensionless parameters to be defined. Using an approximation to the Heaviside function and setting a threshold Mach number M_{tr}, $k^{(HM)}$ and $k^{(LM)}$ are defined as:

$$k^{(HM)} = \frac{1}{1 + e^{-2k(M_{loc} - M_{tr})}}, \tag{1.33}$$

$$k^{(LM)} = 1 - k^{(HM)}, \tag{1.34}$$

where k is a scalar parameter.

The described strategy has been shown to be efficient for Mach numbers as low as $M = 10^{-6}$. See Section 1.4.1 for details and numerical illustrations.

Remark 1.3.5 *The parameters ω_1 and ω_2 still have to be tuned manually depending on the particular application. A rigorous theoretically justified adaptive scaling of these coefficients can lead to an improvement of the algorithm.*

The operators $\mathbf{A}_x$ and $\mathbf{A}_y$ are then redefined to include the implicit artificial dissipation terms in the corresponding directions.

1.3.3 Multicomponent case.

The interface capturing is based on the following advection equation for the volume of fluid (VoF) function ϕ (see e.g. [6],[102],[2],[3],[79],[19]):

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{1.35}$$

where $\mathbf{u} = (\frac{m}{\rho}, \frac{n}{\rho}) = (u, v)$ is the flow velocity. Initially, ϕ is set to 1 for the first fluid and 0 for the second one. Due to numerical diffusion, this initial discontinuity is smoothed out and leads to the formation of an artificial mixing layer where $\phi \in (0, 1)$. EOS parameters have to be consistently defined in the layer to avoid spurious pressure oscillations that appear when a conservative formulation is employed (see e.g. [6],[79],[3],[64]). Here we use the Stiffened Gas EOS (SG EOS) (see [32]):

$$p = (\gamma - 1)\rho e - \gamma \pi_\infty, \tag{1.36}$$

where γ and π_∞ are constants for a given fluid, e is the internal energy, defined implicitly through

$$E = \rho e + \rho \frac{u^2 + v^2}{2}. \tag{1.37}$$

Barotropic Mixture Laws.

The SG EOS is commonly used to model compressible multicomponent flows of gases, liquids, and solids (e.g. air, water, copper, uranium) (see [19] for details). Note that the ideal gas EOS is a particular case of the SG EOS with $\pi_\infty = 0$.

If ρ_i, e_i, p_i $(i = 1, 2)$ are density, internal energy and pressure of corresponding fluids, then:

$$\rho = \phi\rho_1 + (1 - \phi)\rho_2 \tag{1.38}$$

$$\rho e = \phi\rho_1 e_1 + (1 - \phi)\rho_2 e_2 \tag{1.39}$$

$$p = \phi p_1 + (1 - \phi)p_2. \tag{1.40}$$

In the regions of pure fluid 1 or pure fluid 2, the pressure is given by the EOS (1.36). In the artificial mixing layer, where both components are present, the following condition must be satisfied (see [6]):

$$p_1 = p_2 = p. \tag{1.41}$$

Then

$$\begin{gathered}\rho e = \phi \frac{p + \gamma_1 \pi_1}{\gamma_1 - 1} + (1 - \phi) \frac{p + \gamma_2 \pi_2}{\gamma_2 - 1} = \\ p \left(\frac{\phi}{\gamma_1 - 1} + \frac{1 - \phi}{\gamma_2 - 1} \right) + \left(\frac{\phi \gamma_1 \pi_1}{\gamma_1 - 1} + \frac{(1 - \phi) \gamma_2 \pi_2}{\gamma_2 - 1} \right),\end{gathered} \tag{1.42}$$

which leads to the following definition of the proper averaging for the EOS coefficients in the mixing layer (γ, π^∞):

$$\alpha = \frac{1}{1 - \gamma} = \phi \frac{1}{1 - \gamma_1} + (1 - \phi) \frac{1}{1 - \gamma_2} \tag{1.43}$$

$$\beta = \frac{\pi^\infty \gamma}{1 - \gamma} = \phi \frac{\pi_1^\infty \gamma_1}{1 - \gamma_1} + (1 - \phi) \frac{\pi_2^\infty \gamma_2}{1 - \gamma_2}. \tag{1.44}$$

Further considerations will reveal the important role of the parameters α and β in preserving contact discontinuities at interfaces.

Preservation of contact discontinuities at interfaces.

Another requirement needed to preserve the pressure and velocity equilibrium in multicomponent flows is that if the velocity and pressure are constants at the interface at time t^n they remain constants at t^{n+1} (preservation of a contact discontinuity at interfaces) (see [102] for details):

if $u_i^n = u = const$ and $p_i^n = p = const$ for any i, then $u_i^{n+1} = u$ and $p_i^{n+1} = p$.

To avoid technicalities, the one-dimensional inviscid case will be considered here. The following analysis can be easily extended to the multidimensional case. The viscous part of the equations does not disrupt the velocity and pressure equilibrium since it only involves terms with velocity derivatives. We include high- and low-Mach number stabilization terms in the analysis to show that both satisfy the contact discontinuity preservation property.

For a one-dimensional inviscid flow, system (1.8) becomes (see Appendix A for details):

$$\begin{aligned}\mathbf{U}^{n+1} + \tau\frac{\partial}{\partial x}\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1}\right] &= \mathbf{U}^n + \tau\frac{\partial}{\partial x}\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^n - \mathbf{F}^n\right] \\ &+\tau\mathbf{D}_1(\mathbf{U}^{n+1}) + \tau\mathbf{D}_2(\mathbf{U}^{n+1}) + \tau\mathbf{D}_3\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1}\right],\end{aligned} \tag{1.45}$$

where

$$\mathbf{D}_1(\mathbf{U}^{n+1}) = k^{(HM)}h\frac{\partial}{\partial x}\left[\lambda\frac{\partial}{\partial x}\mathbf{U}^{n+1}\right] \tag{1.46}$$

$$\mathbf{D}_2(\mathbf{U}^{n+1}) = k^{(LM)}h^2\frac{\partial^2}{\partial x^2}\mathbf{U}^{n+1} \tag{1.47}$$

$$\mathbf{D}_3\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1}\right] = k^{(LM)}h^2\frac{\partial^2}{\partial x^2}\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1}\right]. \tag{1.48}$$

Here, α and β are included in the set of variables, thus extending the set $\mathbf{U} = (\rho, m, E)$ to $\mathbf{U} = (\rho, m, E, \alpha, \beta) = (\rho, \rho u, \frac{\rho u^2}{2} + \beta + \alpha p, \alpha, \beta))$. The advection equation for the interface is written for α and β, rather than for the VoF function ϕ. The following analysis reveals that this is the natural choice of variables to guarantee the desired preservation property.

Given that $u_i^n = u$ and $p_i^n = p$, it is assumed that the same holds at the next time level, i.e. $u_i^{n+1} = u$ and $p_i^{n+1} = p$. Then, we check if this assumption provides the solution for the discrete linearized mass, momentum and energy equations.

Using Appendix A:

$$\mathbf{F}^n = \begin{bmatrix} \rho^n u & \rho^n u^2 + p & \frac{\rho^n u^3}{2} + u\beta^n + up\alpha^n + up \end{bmatrix}^T \tag{1.49}$$

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -u^2 + \frac{u^2}{2\alpha^n} & 2u - \frac{u}{\alpha^n} & \frac{1}{\alpha^n} & \frac{p}{\alpha^n} & -\frac{1}{\alpha^n} \\ \delta_1 & \delta_2 & u\left(1 + \frac{1}{\alpha^n}\right) & -\frac{up}{\alpha^n} & -\frac{u}{\alpha^n} \end{bmatrix} \tag{1.50}$$

where $\delta_1 = -\frac{\left(\alpha^n p + p + \beta^n + \frac{\rho^n u^2}{2}\right)u}{\rho^n} + \frac{u^3}{2\alpha^n}$, $\delta_2 = \frac{\alpha^n p + p + \beta^n + \frac{\rho^n u^2}{2}}{\rho^n} - \frac{u^2}{\alpha^n}$.

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^n = \begin{bmatrix} u\rho^n & u^2\rho^n & \frac{u^3\rho^n}{2} + u\beta^n + up\alpha^n \end{bmatrix}^T \tag{1.51}$$

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^{n+1} = \begin{bmatrix} u\rho^{n+1} & u^2\rho^{n+1} & \frac{u^3\rho^{n+1}}{2} + u\beta^{n+1} + up\alpha^{n+1} \end{bmatrix}^T. \tag{1.52}$$

Hence,

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^n - \mathbf{F}^n = \begin{bmatrix} 0 & -p & -up \end{bmatrix}^T \tag{1.53}$$

and

$$\frac{\partial}{\partial x}\left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n \mathbf{U}^n - \mathbf{F}^n\right] = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T. \tag{1.54}$$

Using (1.51) the mass conservation equation in a discrete form reads as:

$$\rho_i^{n+1} + \tau u \frac{\rho_{i+1}^{n+1} - \rho_{i-1}^{n+1}}{h} - \rho_i^n - \mathbf{D}_1(\rho^{n+1}) - \mathbf{D}_2(\rho^{n+1}) - \mathbf{D}_3(u\rho^{n+1}) = 0, \quad (1.55)$$

and the momentum conservation as:

$$\begin{aligned} \rho_i^{n+1} u + \tau u^2 \frac{\rho_{i+1}^{n+1} - \rho_{i-1}^{n+1}}{h} - \rho_i^n u - \mathbf{D}_1(\rho^{n+1} u) - \mathbf{D}_2(\rho^{n+1} u) \\ -u\mathbf{D}_3(\rho^{n+1} u) = \\ u\left(\rho_i^{n+1} + \tau u \frac{\rho_{i+1}^{n+1} - \rho_{i-1}^{n+1}}{h} - \rho_i^n - \mathbf{D}_1(\rho^{n+1}) - \mathbf{D}_2(\rho^{n+1}) \right. \\ \left. -u\mathbf{D}_3(\rho^{n+1})\right) = 0, \end{aligned} \quad (1.56)$$

which is satisfied if (1.55) is. Hence, the assumed solution satisfies the mass and momentum parts of (1.45). It was achieved by including α and β into the set of variables. The energy equation can be written as:

$$\begin{aligned} E^{n+1} + \tau u \left(\frac{\rho_{i+1}^{n+1} u^2 - \rho_{i-1}^{n+1} u^2}{2h} + \frac{\beta_{i+1}^{n+1} - \beta_{i-1}^{n+1}}{h} + p \frac{\alpha_{i+1}^{n+1} - \alpha_{i-1}^{n+1}}{h} \right) \\ -E^n - \mathbf{D}_1(E^{n+1}) - \mathbf{D}_2(E^{n+1}) - \mathbf{D}_3(uE^{n+1}) = 0. \end{aligned} \quad (1.57)$$

Expressing energy through the EOS as $E = \alpha p + \frac{u^2}{2}\rho + \beta$, and substituting it into the previous equations allows one to rewrite it as:

$$
\begin{aligned}
&\frac{u^2}{2}\Bigg(\rho_i^{n+1}+\tau u\frac{\rho_{i+1}^{n+1}-\rho_{i-1}^{n+1}}{h}-\rho^n-\mathbf{D}_1(\rho^{n+1})-\mathbf{D}_2(\rho^{n+1})- \\
&\qquad\qquad\qquad\qquad u\boldsymbol{D}_3(\rho^{n+1})\Bigg)+ \\
&p\Bigg(\alpha_i^{n+1}+\tau u\frac{\alpha_{i+1}^{n+1}-\alpha_{i-1}^{n+1}}{h}-\alpha^n-\mathbf{D}_1(\alpha^{n+1})-\mathbf{D}_2(\alpha^{n+1}) \\
&\qquad\qquad\qquad\qquad -u\mathbf{D}_3(\alpha^{n+1})\Bigg) \\
&+\Bigg(\beta_i^{n+1}+\tau u\frac{\beta_{i+1}^{n+1}-\beta_{i-1}^{n+1}}{h}-\beta^n-\mathbf{D}_1(\beta^{n+1})-\mathbf{D}_2(\beta^{n+1}) \\
&\qquad\qquad\qquad\qquad -u\mathbf{D}_3(\beta^{n+1})\Bigg)=0.
\end{aligned}
\tag{1.58}
$$

The expression in the first brackets in equation (1.58) is zero if the mass equation (1.55) is satisfied. The expressions in the second and third brackets are equal to zero if the discretization of the advection equations for α and β:

$$\frac{\partial\alpha}{\partial t}+\mathbf{u}\cdot\nabla\alpha=0 \tag{1.59}$$

and

$$\frac{\partial\beta}{\partial t}+\mathbf{u}\cdot\nabla\beta=0, \tag{1.60}$$

is the same as for the energy equation:

$$
\begin{aligned}
\frac{\alpha_i^{n+1}-\alpha_i^n}{\tau}+u^n\frac{\alpha_{i+1}^{n+1}-\alpha_{i-1}^{n+1}}{h}-\mathbf{D}_1(\alpha^{n+1})-\mathbf{D}_2(\alpha^{n+1})& \\
-\mathbf{D}_3(u^n\alpha^{n+1})=0&
\end{aligned}
\tag{1.61}
$$

$$
\begin{aligned}
\frac{\beta_i^{n+1}-\beta_i^n}{\tau}+u^n\frac{\beta_{i+1}^{n+1}-\beta_{i-1}^{n+1}}{h}-\mathbf{D}_1(\beta^{n+1})-\mathbf{D}_2(\beta^{n+1})& \\
-\mathbf{D}_3(u^n\beta^{n+1})=0.&
\end{aligned}
\tag{1.62}
$$

This analysis can be trivially extended to the multidimensional case, where the same argument leads to the use of the same factorization and splitting error reduction strategies for the interfacial advection as for the rest of the system.

Hence, similar to the Godunov-Rusanov scheme from [102], the contact discontinuity preservation condition determines the discretization of the advection equation for the VoF function. Furthermore, this analysis reveals that the parameters α and β are the natural choice for advected variables and should be taken into account when constructing the Jacobians and performing the linearization.

However, solving two similar advection equations for both α and β can be avoided. Indeed, they both follow from the solution of (1.35) and (1.43)-(1.44). ϕ^{n+1} can be found before updating other variables and used to compute α^{n+1} and β^{n+1}. This allows one to compute their contributions to the remaining system as a part of the right-hand-sides of (1.9)-(1.10).

In the most general case, the VoF function is updated through an iterative (until the overall splitting error is reduced enough) solution of:

$$(\mathbf{I} + \tau u^n \partial_x - \tau \mathbf{D}_x)\hat{\phi}^{n+1}_{k+1} = \phi^n + \mathbf{ER}(\phi^{n+1}_k) \tag{1.63}$$

$$(\mathbf{I} + \tau v^n \partial_y - \tau \mathbf{D}_y)\phi^{n+1}_{k+1} = \hat{\phi}^{n+1} \tag{1.64}$$

where

$$\mathbf{D}_x\phi = k^{(HM)}h\frac{\partial}{\partial x}\left[\lambda\frac{\partial}{\partial x}\phi\right] + k^{(LM)}h^2\frac{\partial^2\phi}{\partial x^2} + k^{(LM)}h^2\frac{\partial^2}{\partial x^2}\left(u^n\phi\right), \tag{1.65}$$

$$\mathbf{D}_y\phi = k^{(HM)}h\frac{\partial}{\partial y}\left[\lambda\frac{\partial}{\partial y}\phi\right] + k^{(LM)}h^2\frac{\partial^2\phi}{\partial y^2} + k^{(LM)}h^2\frac{\partial^2}{\partial y^2}\left(v^n\phi\right), \tag{1.66}$$

$$\mathbf{ER}(\phi_k^{n+1}) = \tau^2(u^n\partial_x - \mathbf{D}_x)(v^n\partial_y - \mathbf{D}_y). \tag{1.67}$$

Interface sharpening.

Interface-capturing schemes are known to diffuse interfaces during the course of computations, due to the introduction of artificial dissipation. A sharpening algorithm is proposed in [110] to keep the width of the interface constant (typically several grid cells). To achieve this, ρ and ϕ are updated by solving the Euler explicit discretization of the following artificial compression equations until a steady state with some predefined tolerance tol:

$$\frac{\partial\rho}{\partial T} = H(\phi)\mathbf{n}\cdot\left(\nabla\left(\epsilon_h\mathbf{n}\cdot\nabla\rho\right) - (1-2\phi)\nabla\rho\right) \tag{1.68}$$

$$\frac{\partial\phi}{\partial T} = \mathbf{n}\cdot(\epsilon_h|\nabla\phi| - \phi(1-\phi)), \tag{1.69}$$

where T is an artificial time-like parameter, ϵ_h is the parameter that controls the thickness of the interface, and

$$H(\phi) = tanh((\frac{\phi(1-\phi)}{10^{-2}})^2), \tag{1.70}$$

is a regularized δ-function that limits the artificial compression to the interfacial layer and prevents the density update to influence other types of discontinuities, such as shock waves.

The parameter ϵ_h is typically chosen to be proportional to the grid size, as $\epsilon_h = \epsilon\cdot h$, where ϵ is a scalar dimensionless parameter to be defined for each particular application. The algorithm is robust and efficient, even for high density ratios.

However, this algorithm also maintains a relatively sharp interfacial profile that hampers the computation of the interface normal. Therefore, [110] proposed to use the following function:

$$\psi = \frac{\phi^{\alpha}}{\phi^{\alpha} + (1-\phi)^{\alpha}}, \alpha < 1 \tag{1.71}$$

for computation of the normal vector, as:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} = \frac{\nabla\psi}{|\nabla\psi|}. \tag{1.72}$$

Surface tension effects.

One of the advantages of the interface-capturing approach is a relatively straightforward implementation of surface tension interfacial effects, which can be incorporated into the model by adding a continuous surface force (CSF) explicitly to the governing equation. Reference [96] proposes two possible approaches, non-conservative and conservative formulations of the CSF.

The first approach adds the following contributions to the momentum and energy equations (assuming that $\phi = 1$ for the liquid phase, $\phi = 0$ for the gas phase):

$$\mathbf{F}_{m}^{ST} = \sigma\kappa\nabla\phi \tag{1.73}$$

$$\mathbf{F}_{E}^{ST} = \sigma\kappa\mathbf{u}\cdot\nabla\phi, \tag{1.74}$$

where σ $[\frac{N}{m}]$ is the surface tension coefficient, $\kappa = -\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$ is the interfacial curvature.

An alternative way is to redefine the total energy by including a term associated with the interfacial energy:

$$\hat{E} = E + \sigma|\nabla\phi| \tag{1.75}$$

and write the CSF as:

$$\mathbf{F}_m^{ST} = -\nabla \cdot \left(-\sigma\left(|\nabla\phi|\hat{\mathbf{I}} - \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|}\right)\right) \tag{1.76}$$

$$\mathbf{F}_E^{ST} = -\nabla \cdot \left(-\sigma\left(|\nabla\phi|\hat{\mathbf{I}} - \frac{\nabla\phi \otimes \nabla\phi}{|\nabla\phi|}\right) \cdot \mathbf{u}\right). \tag{1.77}$$

The authors of [96] claim that the conservative approach (i.e. equations (1.76), and (1.77)) may lead to an attenuation of parasitic currents (see [59],[60]), however other researchers (see [33]) associate the parasitic currents with curvature computations and related errors, rather than a surface tension model. A comparison of different CSF models and curvature approximations in the context of the present numerical method is outside of the scope of this study.

In this study, the non-conservative approach is used, and the interfacial curvature is computed based on the interfacial normal defined by (1.72) and the centered-in-space discretization.

Remark 1.3.6 *The present method allows for a simple extension: the CSF can be added implicitly (in the conservative or non-conservative formulation) as another flux and linearized similarly as all the other fluxes in (1.2). It may be expected that such implicitness in the computation of CSF could help to* *relax the CFL conditions associated with capillary effects, which is of the type* $\tau \leq const \cdot \left(\frac{h^3}{\sigma}\right)^{\frac{1}{2}}$ *(see [33]).*

1.3.4 Maximum speed estimation for the multicomponent Stiffened Gas EOS.

The Guermond-Popov artificial viscosity algorithm requires an estimation from above of the maximum speed of propagation in local one-dimensional Riemann problems. The procedure proposed in [47] has to be modified for the multicomponent case when both fluids obey the SG EOS, and the interfacial jump in pressure is present due to the interfacial curvature and surface tension. Suppose the following parameters are given: (ρ_l, u_l, e_l, p_l, γ_l, π_l^∞,ϕ_l) and (ρ_r, u_r, e_r, p_r, γ_r, π_r^∞,ϕ_r), and each of them obeys the EOS:

$$p_z = (\gamma_z - 1)\rho_z e_z - \gamma_z \pi_z^\infty \tag{1.78}$$

where $z = l, r$. The speed of sound can be computed as:

$$c_z = \sqrt{\frac{\gamma_z(p_z + \pi_z^\infty)}{\rho_z}}. \tag{1.79}$$

The following one-dimensional Riemann problem is considered (see [47] for details):

$$\partial_t \mathbf{U} + \partial_x(\mathbf{F}(\mathbf{U}) \cdot \mathbf{n})) = 0, \tag{1.80}$$

where $\mathbf{n}$ is a unit vector normal to a face of a finite difference cell, with the piecewise constant initial conditions described above. The problem is strictly hyperbolic and its solution consists of two genuinely nonlinear waves (shock or rarefaction) and one linearly degenerate middle wave (contact discontinuity) connecting left and right initial states (see [120], Chapter 4 for details). The VoF function ϕ is assumed to have zero jumps across shock and rarefaction waves, and contact discontinuities. If λ_1^- and λ_3^+ are two extreme wave speeds, a maximum speed of wave propagation is defined as:

$$\lambda_{max} = max((\lambda_1^-)_-, (\lambda_3^+)_+), \tag{1.81}$$

where $(\lambda_1^-)_- = max(0, -\lambda_1^-)$, $(\lambda_3^+)^+ = max(0, \lambda_3^+)$. This is the quantity used in the GP dissipation term.

Denoting intermediate pressures by p_l^* and p_r^*, the following interfacial condition must be satisfied:

$$p_r^* = p_l^* + \kappa\sigma, \tag{1.82}$$

where $\kappa = -\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$ is the interfacial curvature. The case of $\sigma = 0$ is considered first, then an extension of the proposed solution for nonzero values of σ is discussed.

Similar to the ideal gas case, described in [120], Chapter 4, the function

$$\eta(p) = f_l(p) + f_r(p) + u_r - u_l \tag{1.83}$$

is considered, where f_z is a shock curve if $p \geq p_z$

$$f_z(p) = (p - p_z)\sqrt{\frac{2}{(\gamma_z + 1)\rho_z}}\left(p + \frac{\gamma_z - 1}{\gamma_z + 1}p_z + \frac{\gamma_z}{\gamma_z + 1}\pi_z^\infty\right)^{-\frac{1}{2}} \tag{1.84}$$

and a rarefaction curve otherwise (see [51])

$$f_z(p) = \frac{2c_z}{\gamma - 1}\left(\left(\frac{p + \pi_z^\infty}{p_z + \pi_z^\infty}\right)^{\frac{\gamma_z - 1}{2\gamma_z}} - 1\right) \tag{1.85}$$

with $z = l, r$.

An intermediate pressure $p_l^* = p_r^* = p^*$ is given then as a solution of (see [120], Chapter 4):

$$\eta(p^*) = 0. \tag{1.86}$$

If it is solved for p^*, then

$$\lambda_1^- = u_l - c_l,\ \lambda_3^+ = u_r + c_r \tag{1.87}$$

for rarefaction waves, and

$$\lambda_1^- = u_l - \frac{Q_l}{\rho_l},\ \lambda_3^+ = u_r + \frac{Q_r}{\rho_r} \tag{1.88}$$

for shocks, where Q_z are the corresponding mass fluxes,

$$Q_z = \sqrt{\frac{(p^* + \frac{\gamma-1}{\gamma+1}p_z + \frac{\gamma}{\gamma+1}\pi_z^\infty)(\gamma_z + 1)\rho_z}{2}}. \tag{1.89}$$

Finally,

$$\lambda_{max} = max((\lambda_1^-)_-, (\lambda_r^+)_+). \tag{1.90}$$

Now, since both f_l and f_r are monotonically increasing and concave down, the Newton-Secant method can be used to solve (1.86) for p^*. Given $p_n^1 < p^* < p_n^2$ one can compute

$$p_{n+1}^1 = p_n^1 - \frac{\eta(p_n^1)}{\eta'(p_n^1)} \tag{1.91}$$

$$p_{n+1}^2 = p_n^2 - \eta(p_n^2)\frac{p_n^2 - p_n^1}{\eta(p_n^2) - \eta(p_n^1)} \tag{1.92}$$

until convergence. The procedure gives the estimation of p^* from above and below, which is used to estimate the maximum speed of propagation.

Shocks and rarefaction waves can be distinguished by using the signs of $\eta(p_l)$ and $\eta(p_r)$ (positive for rarefaction waves, negative for shock waves). If both waves are rarefaction waves, p^* is not needed for the speed estimation and the iterative process can be skipped by immediately computing the wave speeds.

Assuming now a non-zero σ, (1.83) can be reformulated as:

$$f_l(p_l^*) + f_r(p_r^*) + u_r - u_l = 0 \tag{1.93}$$

which needs to be solved in terms of one of the pressures, then the other one can be computed using (1.82) if needed (i.e. if the corresponding wave is a shock).

1.3.5 Summary of the method.

The overall algorithm can be summarized in the pseudocode format as following:

```
$t = 0$
Define initial conditions for $[\rho, m, n, E, \phi]$
Compute $u^0$,$v^0$,$p^0$
Initialize $\phi^0$ and $\rho^0$ using the interface sharpening procedure (Section 1.3.3)
Initialize $\mathbf{U}^0 = [\rho^0, m^0, n^0, E^0, \alpha^0, \beta^0]$ using $\phi^0$,$\rho^0$,$u^0$,$v^0$
$n = 0$
**DO WHILE** $t < T_{end}$
    $t = t + \tau$
    Compute all the Jacobian matrices using $\mathbf{U}^n$ (see Appendix A)
    Compute all the artificial viscosity coefficients for the GP-dissipation term
    **IF** (reduction (Section 1.3.1) is on) **THEN**
        $\mathbf{U}_0^{n+1} = \mathbf{U}^n$, $\phi_0^{n+1} = \phi^n$
    **ELSE**
        $\mathbf{U}_0^{n+1} = 0$, $\phi_0^{n+1} = 0$
    **END IF**
    k = 0
    **DO WHILE** ($\mathbf{ER}(\mathbf{U}_k^{n+1}) > tolerance$)
        Solve advection of the VoF function ((1.63) and (1.64)) for $\phi_{k+1}^{n+1}$
        Solve (1.11) and (1.12) for $\mathbf{U}_{k+1}^{n+1}$
        Compute $\mathbf{ER}(\mathbf{U}_{k+1}^{n+1})$
        k = k+1
    **END DO**
    Compute $u^{n+1}$, $v^{n+1}$, and $p^{n+1}$
    Use interface sharpening (Section 1.3.3) to update $\rho^{n+1}$ and $\phi^{n+1}$
    Update $U^{n+1}$ using new values of $\rho^{n+1}$, $\phi^{n+1}$, $u^{n+1}$, $v^n + 1$, and $p^{n+1}$
    $n = n + 1$
END DO
```

1.4 Numerical Tests.

In this section, numerical tests that verify the performance of the proposed method are presented. Note that no analytical or extensive numerical study on the stability region has been attempted. For each test, the time step (τ) and the space step (h) are chosen to guarantee the desired accuracy and stability, but the choice may not be optimal. Based on the performed tests, low-Mach number applications do not impose significant stability restrictions, i.e. they allow the value of τ to be at least several times higher than h. Stability regions in high-Mach number applications depend on the types and severity of discontinuities in the flow. The stability regions for our method were found to be larger than ones for the explicit version of the same method, for which CFL-condition is of the type $\tau < const \cdot \frac{h}{|\lambda_{max}|}$ where $|\lambda_{max}|$ is the maximum speed of propagation (see [48] for stability analysis of the explicit finite element Guermond-Popov scheme). The use of explicit non-reflection boundary conditions, as in Sections 1.4.9 and 1.4.10, is expected to strengthen the stability restrictions (e.g. see [14]).

1.4.1 Gresho Vortex.

The behavior of the scheme in the low Mach number limit is investigated using the Gresho vortex test (similar to [86]), a time-independent inviscid rotating vortex placed in a square periodic domain. Scaling the pressure with respect to the reference Mach number allows the study of stability and dissipative properties of numerical methods for different Mach numbers (see [87] as an example).

Initial conditions are given by (see Figure 1.1 for initial pressure and Mach number distributions for $M_0 = 10^{-1}$):

$$p_0 = \frac{\rho_0}{\gamma M_0^2} \tag{1.1}$$

$$(u_\phi(r), p(r)) = \begin{cases} (5r, p_0 + \frac{25}{2}r^2), \\ \qquad\qquad\qquad \text{if } 0 \leq r < 0.2 \\ \\ (2 - 5r, p_0 + \frac{25}{2}r^2 + \\ \quad 4(1 - 5r - \ln 0.2 + \ln 5), \\ \qquad\qquad\qquad \text{if } 0.2 \leq r < 0.4 \\ \\ (0, p_0 - 2 + 4\ln 2, \\ \qquad\qquad\qquad\qquad \text{if } 0.4 \leq r \end{cases} \tag{1.2}$$

where M_0 is the reference Mach number, and (r,ϕ) are polar coordinates with the origin placed in the center of the vortex.

The range of Mach numbers considered here is $[10^{-6}; 10^{-1}]$. As can be seen in Figure 1.2, if no splitting error reduction technique is used the vortex is not well preserved in time and is completely dissipated after one revolution (by the

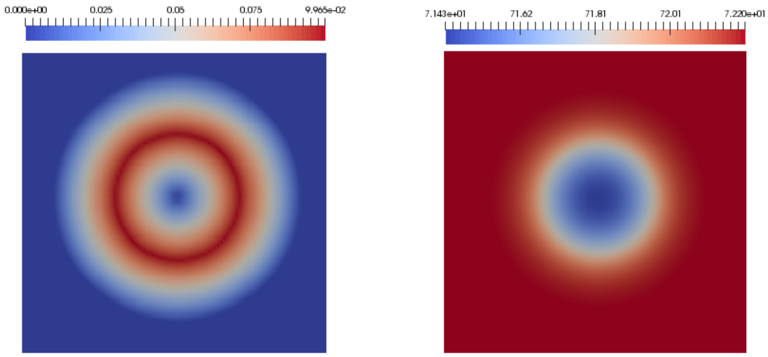

Figure 1.1: Initial Pressure and Mach number distributions for $M_0 = 10^{-1}$. The domain is $[0,1] \times [0,1]$.

time $t = 1$). However, if one step of the splitting error reduction described in Section 1.3.1 is used, the proper time-independent behavior is recovered (see Figure 1.3). The method remains efficient in the considered range of Mach numbers, i.e. choosing the reference Mach number as low as $M_0 = 10^{-6}$ does not require any decrease of time or space steps. Figure 1.4 shows the time evolution of the relative kinetic energy for the computations described above. It can be concluded that the energy loss is due to the splitting error and its reduction allows one to recover conservative properties up to the influence of the artificial dissipation, required for stability.

All the tests are performed using the LM-dissipation term, described in Section 1.3.2. Parameters ω_1 and ω_2 are chosen for each M_0 to guarantee an acceptable rate of kinetic energy dissipation (ω_1) and dumping of high frequency oscillations (ω_2). Figures 1.6 and 1.8 illustrate the effect of these parameters. Figure 1.7 shows the ineffectiveness of the artificial dissipation operator of the type $\mathbf{D}_1^{LM}\mathbf{U}$ in dealing with node-to-node pressure oscillations, which have an increasing effect as M_0 decreases. Figure 1.9 reveals that an increase of ω_2

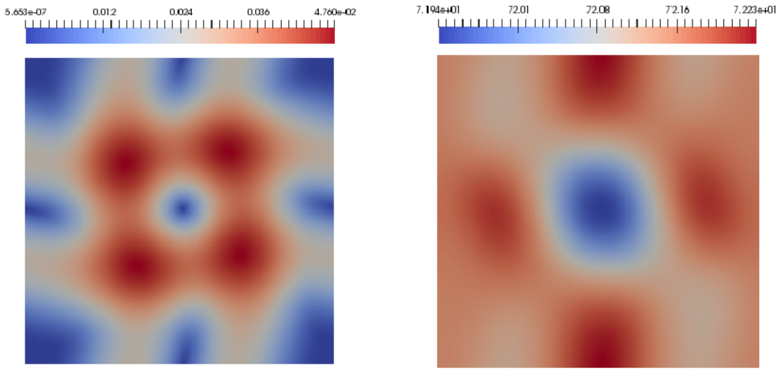

Figure 1.2: Pressure and Mach number distributions at $t = 1$ for $M_0 = 10^{-1}$, no splitting error reduction. The domain is $[0, 1] \times [0, 1]$.

alone leads to an increase of relative kinetic energy and consequently to a rise of instability, especially for relatively high Mach numbers such as $M_0 = 10^{-1}$. Therefore one needs to have both components of the LM-dissipation term. Such a combination allows tuning the algorithm for a wide range of Mach numbers. Figure 1.5 shows that $\frac{|p-p_{max}|}{p_{max}} \sim M^2$, which is the proper scaling in the nearly incompressible regime (see [50] for details). Thus, the presented algorithm remains efficient in the low-Mach number regime, since it maintains the stability and accuracy of the simulations without any decrease of spatial or temporal steps, or any other extra computational cost, except explicit error term computations. Numerical oscillations are successfully removed by the proposed stabilization.

1.4.2 Sod Shock Tube.

The Sod Shock tube test (originally proposed in [111]) is used here to evaluate properties of the method when the GP-dissipation term is used, and compare the implicit version of the scheme (IGP) with the explicit one (EGP) and with

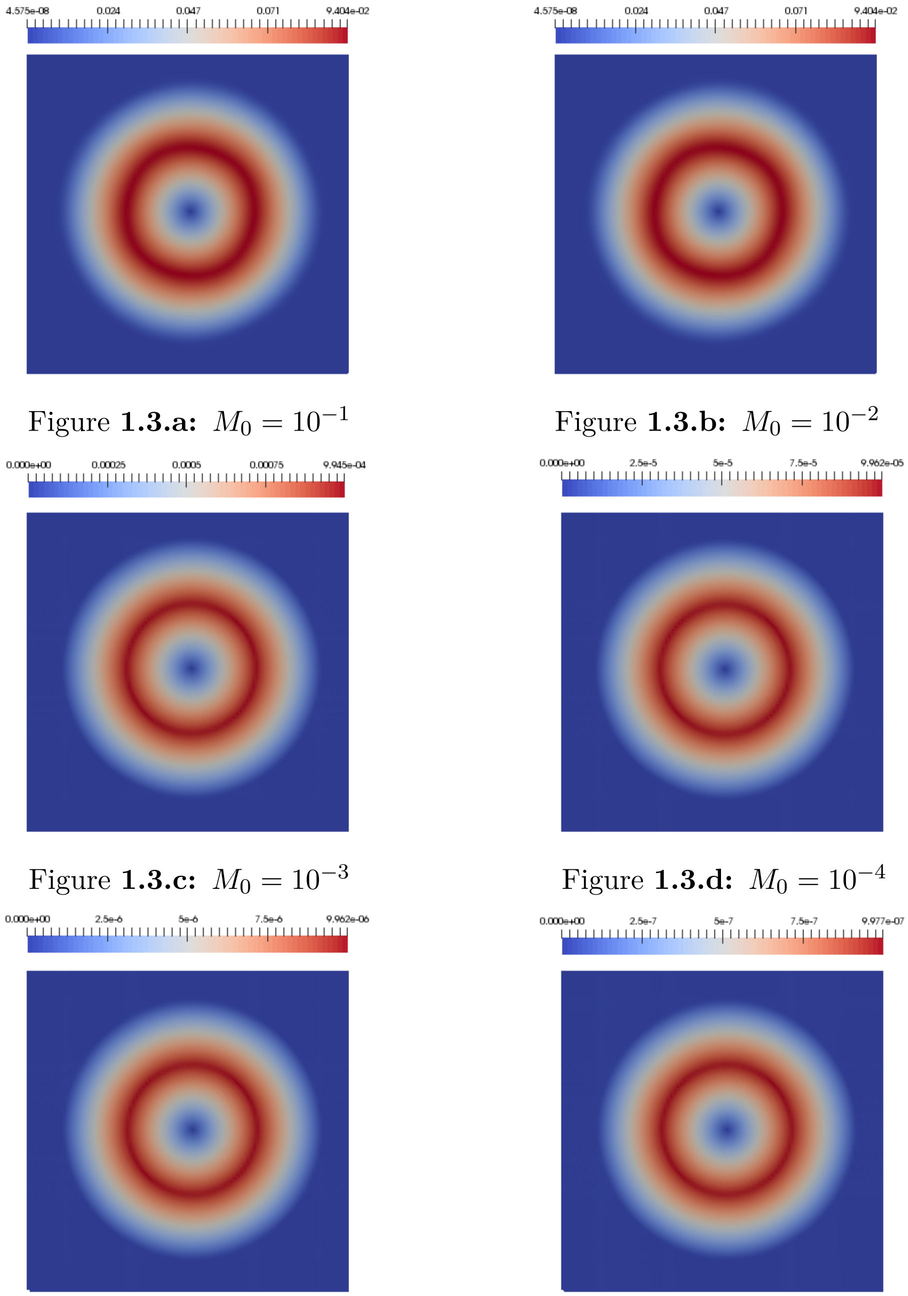

Figure **1.3.a:** $M_0 = 10^{-1}$

Figure **1.3.b:** $M_0 = 10^{-2}$

Figure **1.3.c:** $M_0 = 10^{-3}$

Figure **1.3.d:** $M_0 = 10^{-4}$

Figure **1.3.e:** $M_0 = 10^{-5}$

Figure **1.3.f:** $M_0 = 10^{-6}$

Figure 1.3: Mach number distributions at t=1 for different M_0 computed using the splitting error reduction technique. In all the tests $\tau = 10^{-3}$, grid size: 100x100. The domain is $[0,1] \times [0,1]$.

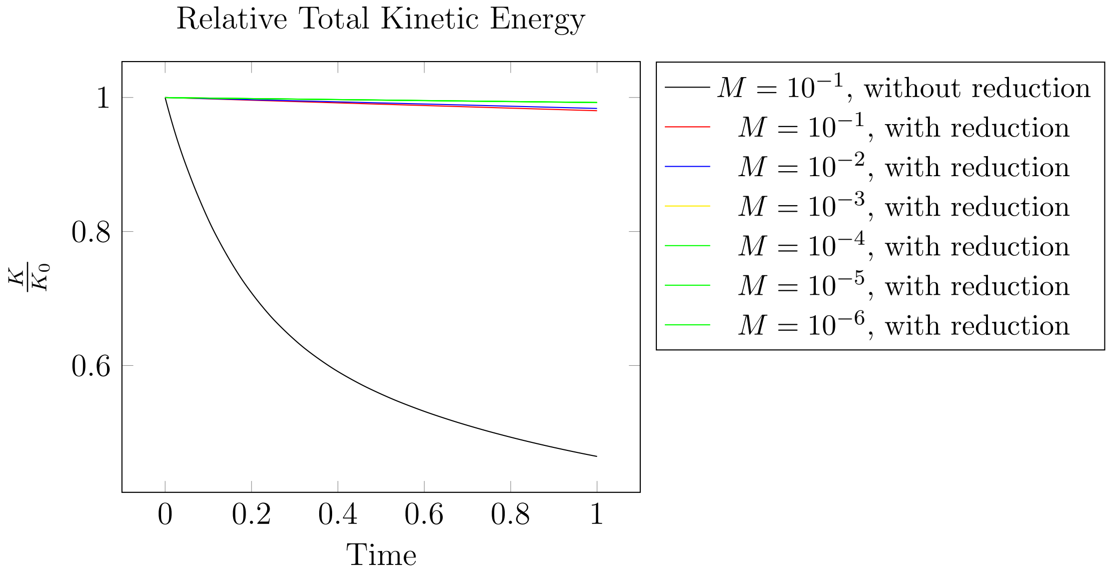

Figure 1.4: Time evolution of the Relative Kinetic Energy for different test cases. In all the tests $\tau = 10^{-3}$, grid size: 100x100.

the classic Lax-Friedrichs method (LF).

The initial conditions are given by:

$$\begin{bmatrix} \rho_L \\ p_L \\ u_L \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \\ 0.0 \end{bmatrix} \tag{1.3}$$

$$\begin{bmatrix} \rho_R \\ p_R \\ u_R \end{bmatrix} = \begin{bmatrix} 0.125 \\ 0.1 \\ 0.0 \end{bmatrix} \tag{1.4}$$

The results at $T_{end} = 0.1$ for the density (ρ), pressure (p), velocity (u) and internal energy (e) are presented in Figure 1.10 ($h = 10^{-3}$) and Figure 1.11

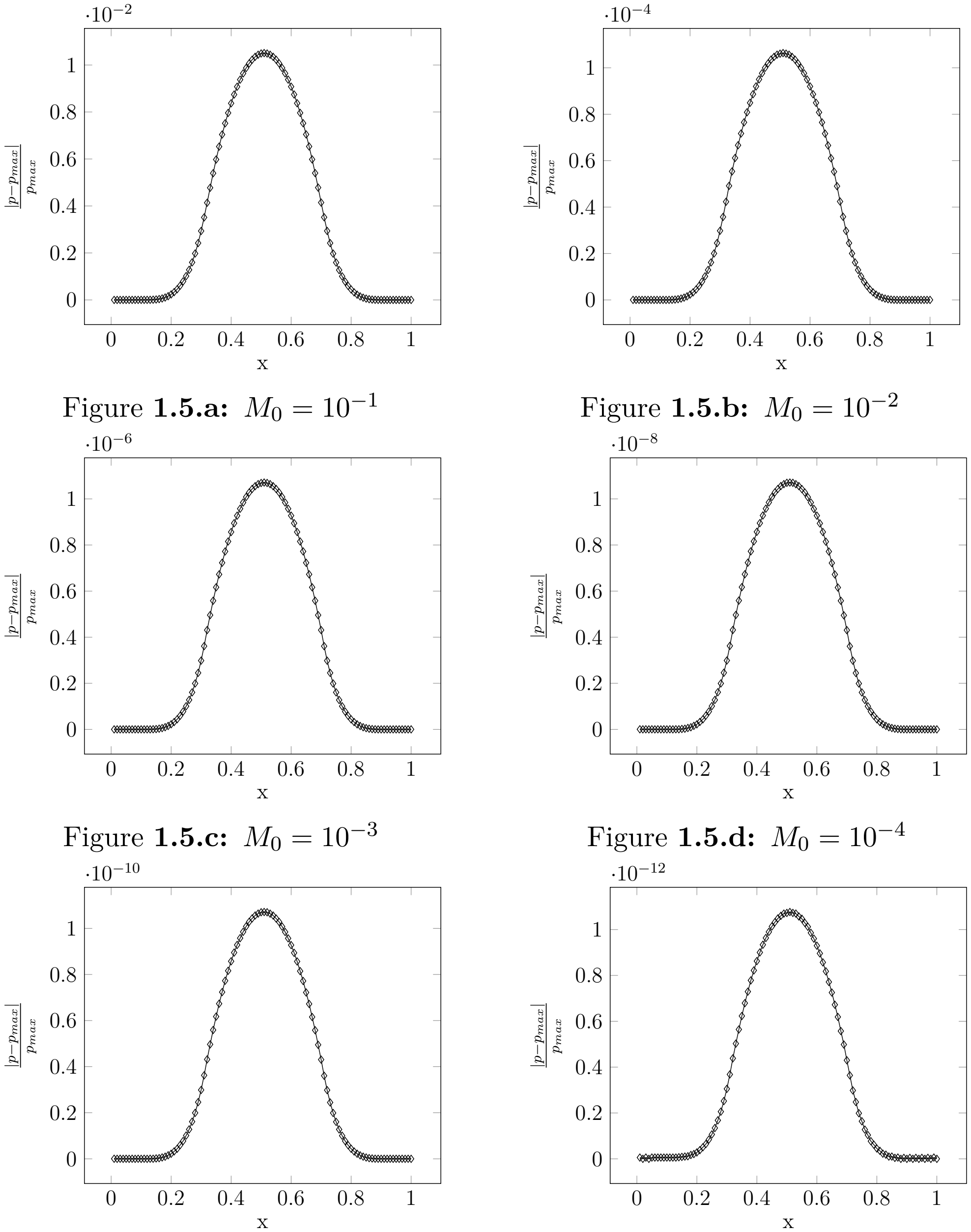

Figure **1.5.a:** $M_0 = 10^{-1}$

Figure **1.5.b:** $M_0 = 10^{-2}$

Figure **1.5.c:** $M_0 = 10^{-3}$

Figure **1.5.d:** $M_0 = 10^{-4}$

Figure **1.5.e:** $M_0 = 10^{-5}$

Figure **1.5.f:** $M_0 = 10^{-6}$

Figure 1.5: Relative pressure variations for different M_0, cross-sections at $x = 0.5$. In all the tests $\tau = 10^{-3}$, grid size: 100x100.

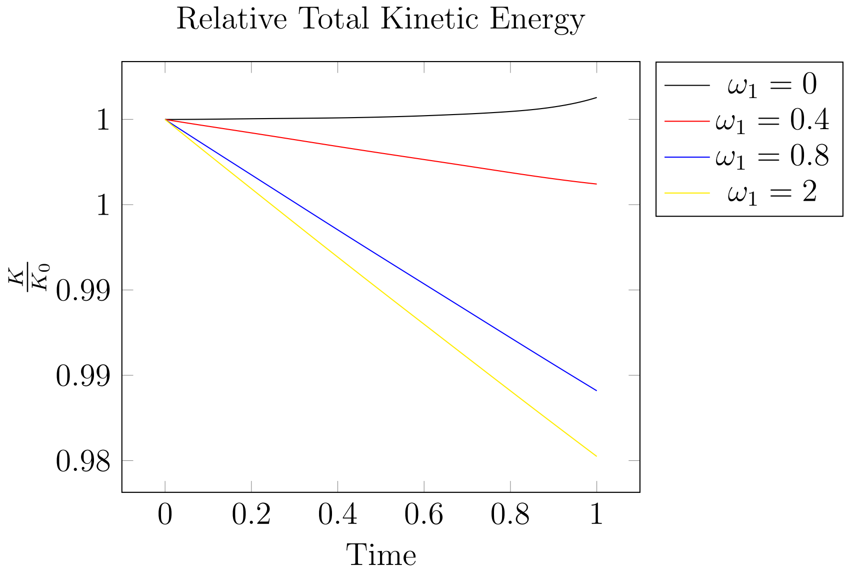

Figure 1.6: Time evolution of the Relative Kinetic Energy depending on the value of ω_1. In all the tests $\omega_2 = 0$, $M_0 = 10^{-1}$, $\tau = 10^{-3}$, grid size: 100x100.

$(h = 0.5 \cdot 10^{-3})$. In both cases $\tau = 10^{-4}$.

It can be seen that explicit and implicit versions of the Guermond-Popov scheme produce almost equivalent solutions in both cases. On a coarser grid, when the space error term has a dominant contribution to the overall error, the Guermond-Popov scheme maintains significantly sharper profiles of discontinuities in the solution than the Lax-Friedrichs method. The results of this test case demonstrate that while the method still adds a first-order dissipation term in its high-Mach number version, the special scaling of the artificial viscosity (proposed in [48]) reduces the amount of dissipation introduced by the stabilization term, without loss of robustness.

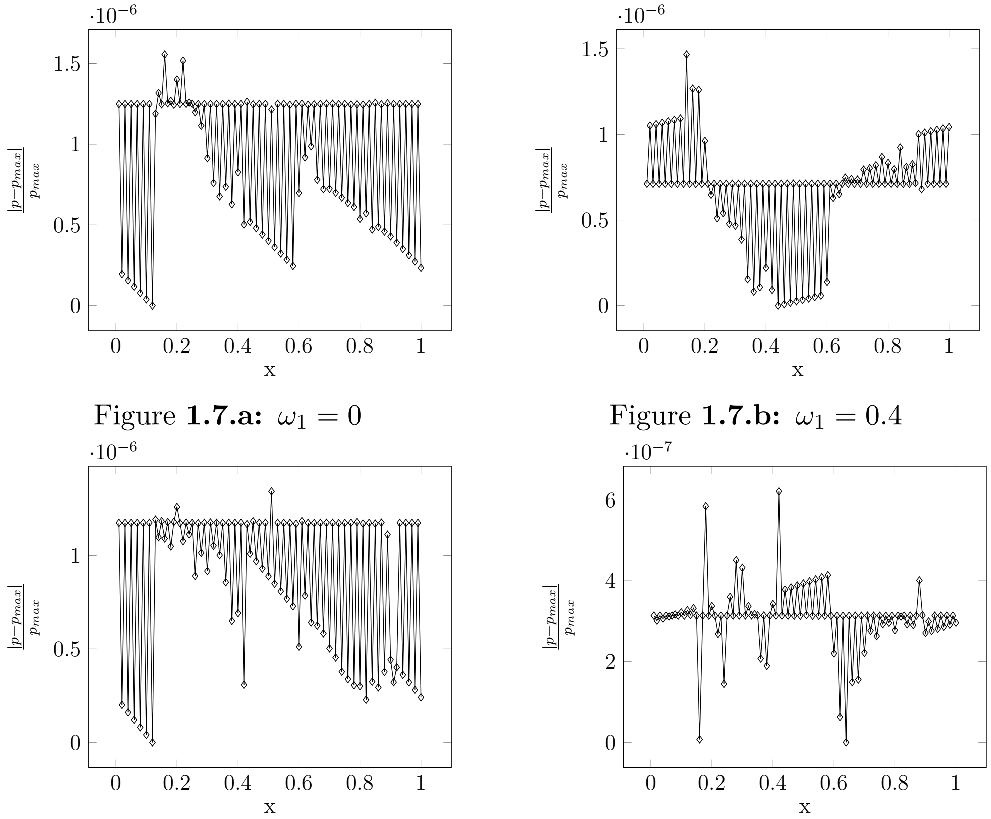

Figure **1.7.a:** $\omega_1 = 0$

Figure **1.7.b:** $\omega_1 = 0.4$

Figure **1.7.c:** $\omega_1 = 0.8$

Figure **1.7.d:** $\omega_1 = 1$

Figure 1.7: Relative pressure variation after one time step, cross-section at $x = 0.5$, for different values of ω_1. Further integration leads to a numerical failure. In all the tests $\omega_2 = 0$, $M_0 = 10^{-6}$, $\tau = 10^{-3}$, grid size: 100x100.

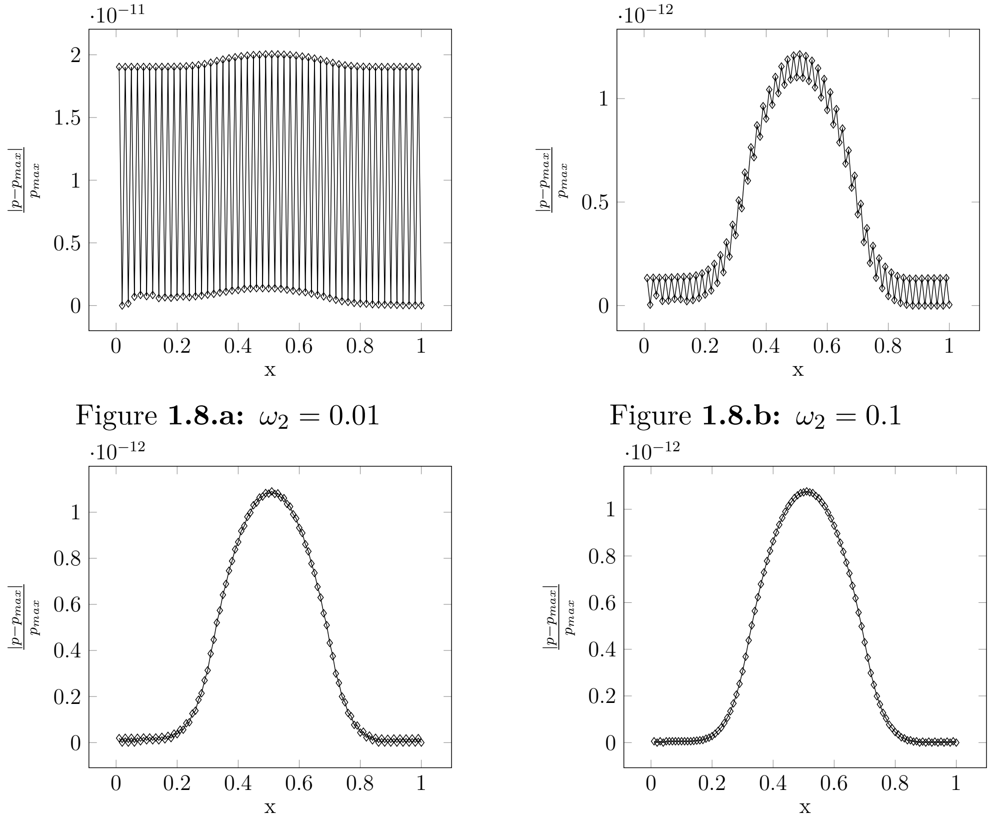

Figure **1.8.a:** $\omega_2 = 0.01$

Figure **1.8.b:** $\omega_2 = 0.1$

Figure **1.8.c:** $\omega_2 = 0.5$

Figure **1.8.d:** $\omega_2 = 1.1$

Figure 1.8: Relative pressure variations after one time step, cross-section at $x = 0.5$, for different values of ω_2. In all the tests $\omega_1 = 1$, $M_0 = 10^{-6}$, $\tau = 10^{-3}$, grid size: 100x100.

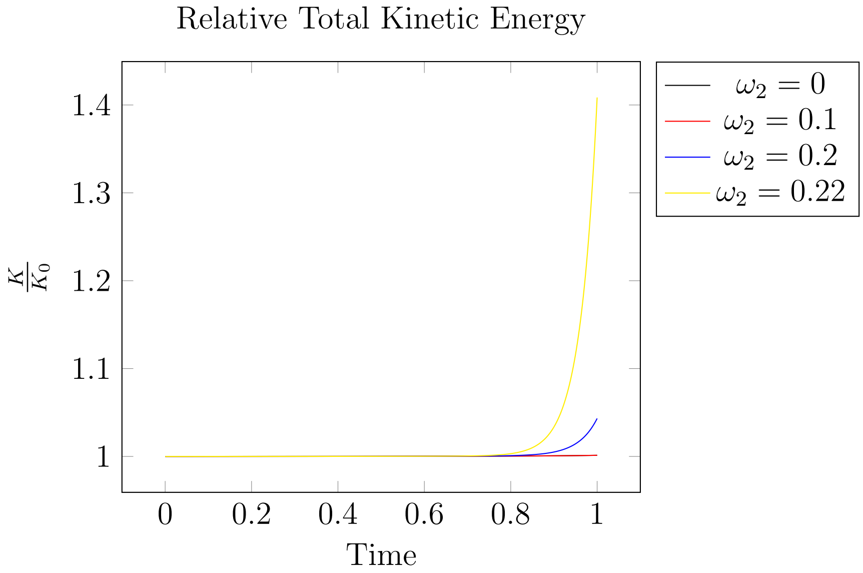

Figure 1.9: Time evolution of the Relative Kinetic Energy depending on the value of ω_2. In all the tests $\omega_1 = 0$, $M_0 = 10^{-1}$, $\tau = 10^{-3}$, grid size: 100x100. Note that the results for $\omega_2 = 0$ and $\omega_2 = 0.1$ are almost identical.

1.4.3 Manufactured Solution.

The method of Manufactured Solution (see [99]) is used here to verify the numerical code implementation and convergence properties of the proposed algorithm. The idea of the method is to choose an analytical solution and modify the governing equations by the inclusion of source terms, computed using the solution. Here, we define primitive variables (ρ,u,v, and p) as:

$$\rho(x, y, t) = \rho_0 + \rho_t \sin t + \rho_x \sin x + \rho_y \cos y \tag{1.5}$$

$$u(x, y, t) = u_0 + u_t \sin t + u_x \sin x + u_y \cos y \tag{1.6}$$

$$v(x, y, t) = v_0 + v_t \cos t + v_x \cos x + v_y \sin y \tag{1.7}$$

$$p(x, y, t) = p_0 + p_t \cos t + p_x \cos x + p_y \sin y, \tag{1.8}$$

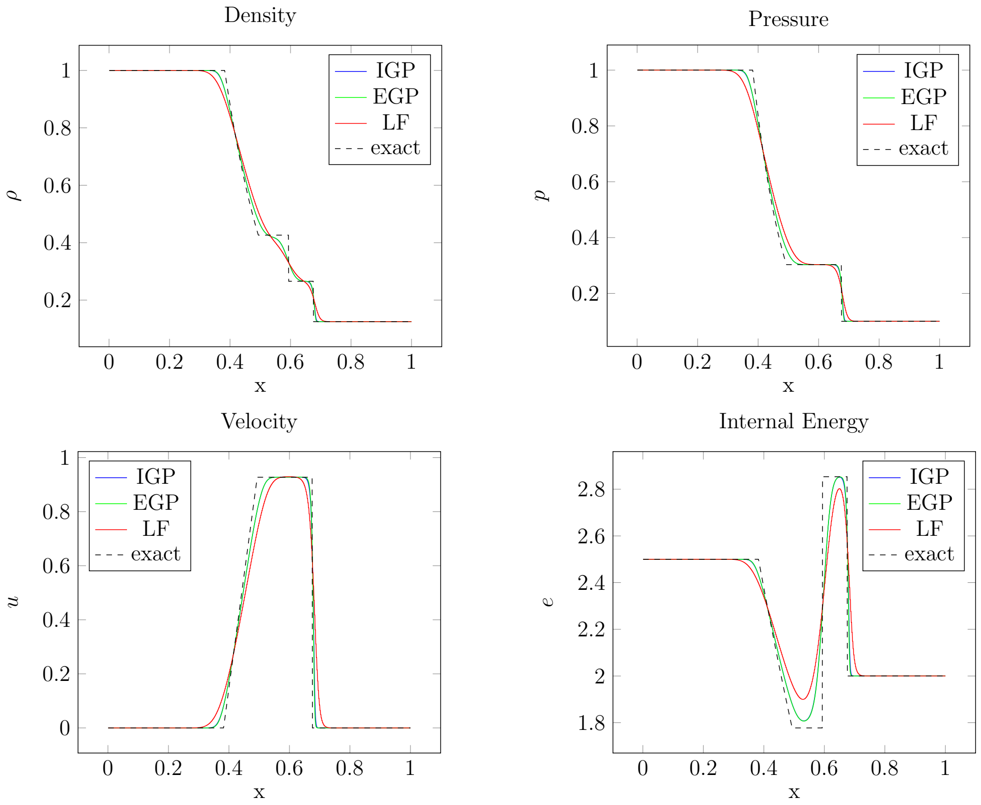

Figure 1.10: Sod Shock Tube test, $T_{end} = 10^{-1}$, $h = 10^{-3}$

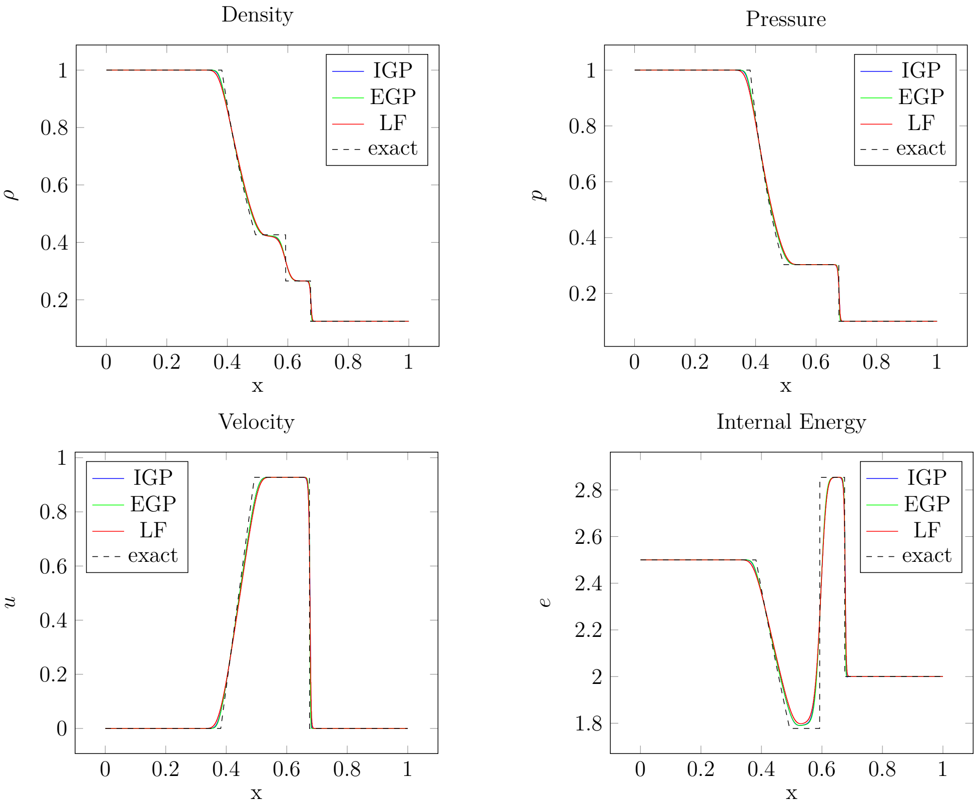

Figure 1.11: Sod Shock Tube test, $T_{end} = 10^{-1}$, $h = 0.5 \cdot 10^{-3}$

where ξ_0, ξ_t, ξ_x, ξ_y (for $\xi = \rho$, u, v, or p) are constants defined separately for high- ($M_0 = 2.14$) and low- ($M_0 = 2.14 \cdot 10^{-4}$) Mach number cases in Tables 1.1 and 1.2. Conservative variables are computed from the primitive ones as usual. In both cases the single fluid ideal gas equation of state is used ($\gamma = 1.4$, $\pi^{\infty} = 0$) with viscosity parameter $\mu = 1$ $[kg/(s \cdot m)]$. The low-Mach number stabilization term $\mathbf{D}^{LM}\mathbf{U}$ is scaled with the parameters $\omega_1 = 1.1$, and $\omega_2 = 10^{-2}$. A single-step version of the splitting error reduction procedure was used for both, high- and low- Mach number cases.

Equation, ξ	ξ_0	ξ_t	ξ_x	ξ_y
$\rho\ (kg/m^3)$	1	0.5	0.015	-0.01
$u\ (m/s)$	8	4	0.05	-0.03
$v\ (m/s)$	8	4	-0.075	0.04
$p\ (N/m^2)$	10	1	0.02	0.05

Table 1.1: Constants for high-Mach number Manufactured Solution.

High-Mach number computations are performed on a $[1 \times 1]$ square domain with exact Dirichlet boundary conditions:

$$\hat{\mathbf{U}}^{n+1} = (\mathbf{I} + \mathbf{A}_y)\mathbf{U}^{n+1}_{exact} \tag{1.9}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^{n+1}_{exact}. \tag{1.10}$$

In the low-Mach number case, a square $[2\pi \times 2\pi]$ domain with periodic boundary conditions is used.

Figures 1.12 and 1.13 show the expected order of accuracy for smooth solutions (first-order convergence in time for both versions of the scheme, first-order convergence in space for the high-Mach number version, and second-order convergence in space for the low-Mach number version). Since the algorithm remains accurate, stable, and oscillation-free for the low-Mach number case, it is demonstrated to be applicable for time-dependent viscous nearly incompressible flows.

Equation, ξ	ξ_0	ξ_t	ξ_x	ξ_y
$\rho\ (kg/m^3)$	1	0.5	0.015	-0.01
$u\ (m/s)$	$8 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$0.05 \cdot 10^{-4}$	$-0.03 \cdot 10^{-4}$
$v\ (m/s)$	$8 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$-0.075 \cdot 10^{-4}$	$0.04 \cdot 10^{-4}$
$p\ (N/m^2)$	10	1	0.02	0.05

Table 1.2: Constants for low-Mach number Manufactured Solution.

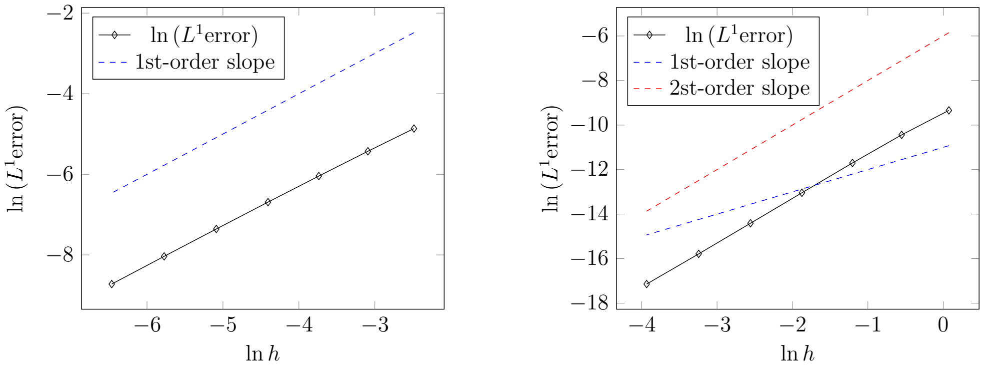

Figure 1.12: *log-log* plots of the discrete L^1 norm of the total energy errors at $t = 10^{-2}$ ($\tau = 10^{-4}$) for high-Mach number (left) and low-Mach number (right) manufactured solutions.

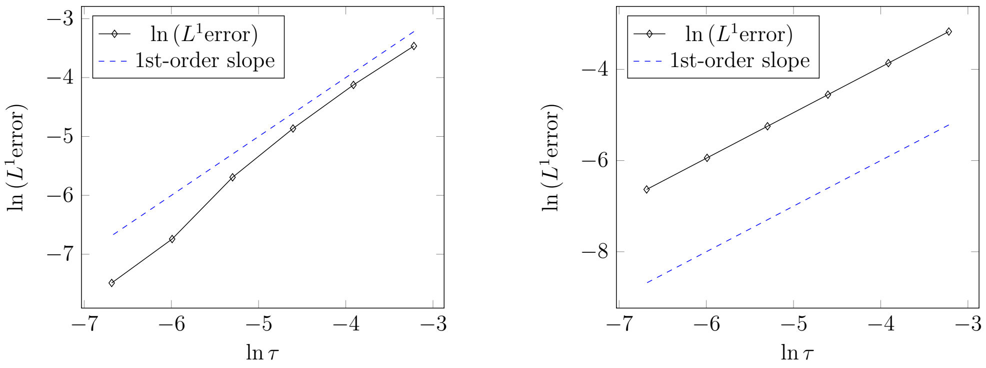

Figure 1.13: *log-log* plots of the discrete L^1 norm of the total energy errors at $t = 1$ on 322×322 uniform grid for high-Mach number (left) and low-Mach number (right) manufactured solutions.

# cores	1(1×1)	64(8×8)	256(16×16)	512(16×32)	992(32×31)
Time Efficiency	31.25 s -	36.78 s 85 %	39.12 s 80 %	41.85 s 75%	45.77 s 68%

Table 1.3: Weak Scalability test, 10^6 grid points per core.

1.4.4 Weak Scalability.

A weak scalability test is provided here to demonstrate the performance of the algorithm on parallel machines. Similar to [42], a fixed number of grid points per CPU core is considered (10^6, to maximize the size of the problem given the memory limitations). Then, CPU-time is recorded for an increasing number of cores. These times, computed while solving the Manufactured Solution test in the high-Mach number regime, can be found in Table 1.3. Scaling efficiency, computed as the ratio of the CPU-time on one core to the CPU-time on n cores, is also presented in Table 1.3. Taking into account that no attempts have been made to optimize the code in general, or the interprocessor communications in particular, the algorithm shows reasonable weak scalability and thus can be considered as a promising one for parallel computations. All these computations were performed on the GRAHAM cluster provided by Compute Canada (www.computecanada.ca).

1.4.5 One-Dimensional Interface Advection.

Here the advection of an interface between air ($\gamma = 1.4$, $\pi^\infty = 0$) and water ($\gamma = 6.12$, non-dimensiolized $\pi^\infty = 0.1631$) under the atmospheric pressure is considered (similar to [19]). The purpose of the test is to demonstrate the absence of spurious pressure oscillations and preservation of pressure and velocity equilibrium for the method with both GP- and LM- ($\omega_1 = 10$, $\omega_2 = 1$) dissipation terms.

The initial conditions are given by:

$$(\rho, u, p, \phi) = \begin{cases} (1, 0.1, 4.819 \times 10^{-5}, 1), & \text{if } x \leq 0.3 \\ (1.204 \times 10^{-3}, 0.1, 4.819 \times 10^{-5}, 0), & \text{if } x > 0.3, \end{cases} \tag{1.11}$$

and are regularized after that by performing the interface sharpening procedure until convergence. It is evolved with $\tau = 10^{-2}$ on a uniform grid of 130 cells until $t_{end} = 4$. The results and initial conditions are shown in Figure 1.14.

In both cases (LM- and GP-dissipation terms) contact discontinuity is well preserved and no spurious oscillations are introduced, despite the high density ratio. The sharp profile of the interface is well maintained. Hence, the test confirms the analysis of the contact discontinuities preservation property of the method, and the compatibility of the presented formulation of the VoF method and interface sharpening with the discretization strategy and the proposed stabilization terms.

1.4.6 Two-Dimensional Interface Advection.

The purpose of this test case is to verify the performance of the interface capturing algorithm in two dimensions. A water drop of radius 0.15 is considered. It is located in the lower-left corner of a square $[1 \times 1]$ domain at $(0.3, 0.3)$ at the initial moment of time and surrounded by air (both water and air have the same parameters as in the previous test). The bubble is then advected to the upper right corner of the domain, $u = v = 0.1$. The problem is discretized

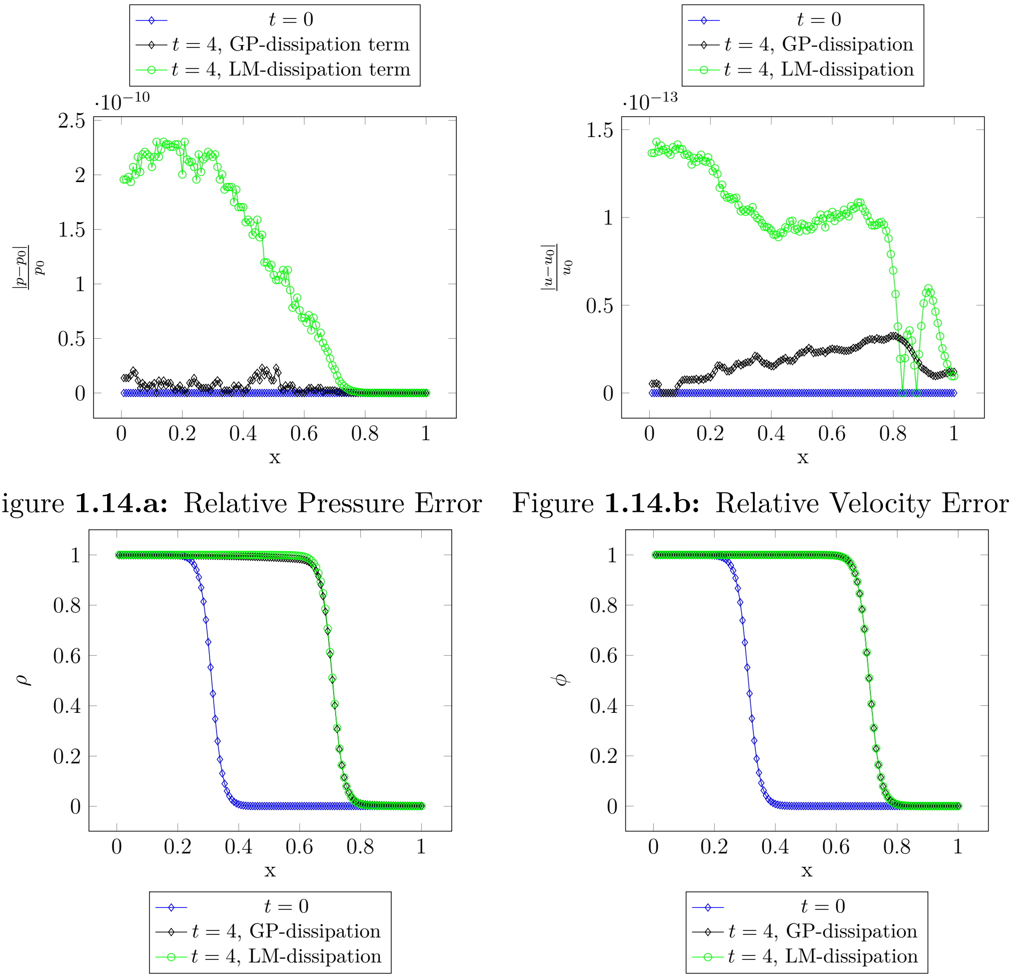

Figure **1.14.a:** Relative Pressure Error

Figure **1.14.b:** Relative Velocity Error

Figure **1.14.c:** Density

Figure **1.14.d:** VoF-function

Figure 1.14: Interface advection using GP- and LM- artificial dissipation terms

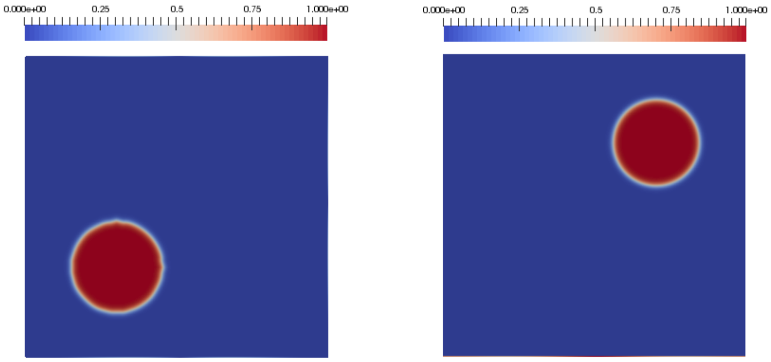

Figure 1.15: VoF-function at $t = 0$ (left) and $t = 4$ (right)

with $\tau = 10^{-2}$ on a 200×200 uniform grid. No splitting error reduction was needed for this test. LM-dissipation was used ($\omega_1 = 10$, $\omega_2 = 0$).

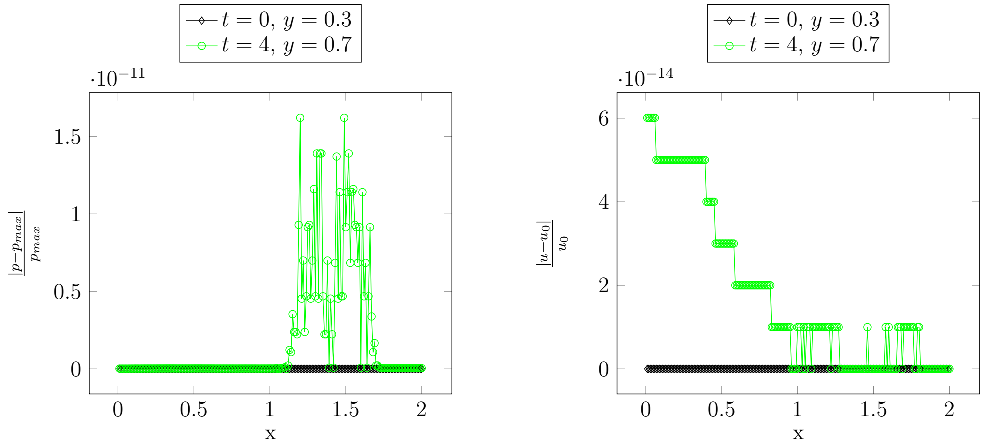

Figure **1.16.a:** Pressure

Figure **1.16.b:** Velocity

Figure 1.16: Relative Pressure and Velocity Errors, cross-sections at $y = 0.3$ and $y = 0.7$

As can be seen in Figures 1.15 and 1.16, the shape of the bubble is well preserved and no spurious oscillations or artificial acoustic waves are introduced. Thus, the preservation of contact discontinuities property of the method extends to the multidimensional case as expected. Furthermore, the test shows the ability of the interface-capturing methodology chosen for the scheme to handle nontrivial interface configurations when used in combination with the proposed discretization technique.

1.4.7 Laplace formula.

Following [96] and [33], the implementation of surface tension is verified by reproducing the pressure jump across a curved interface, given by the Laplace formula. In the case of a cylindrical interface:

$$\Delta p_{exact} = \sigma / R, \tag{1.12}$$

where R is the radius of the cylinder.

At the initial moment of time, a liquid (ρ_l, $\gamma = 2.4$, $\pi^{\infty} = 10^7\ Pa$) cylindrical drop of radius $R = 0.3\ m$ centered at $(0.5\ m, 0.5\ m)$ is placed in a $[1m \times 1m]$ square domain with symmetry boundary conditions, surrounded by gas ($\rho_g = 1\ kg/m^3$, $\gamma = 1.4$, $\pi^{\infty} = 0\ Pa$). The pressure is given by the Laplace formula, with the pressure outside of the drop being $p_{out} = 100\ Pa$, and the pressure inside being $p_{in} = p_{out} + \Delta p_{exact}$.

The relative pressure jump error $E(\Delta p) = \frac{\Delta p - \Delta p_{exact}}{\Delta p_{exact}}$, and the radius of the cylindrical drop R are computed by an averaging procedure, similar to the one used in [96]. Nodes with $\phi \geq 0.9$ are considered to be inside, and nodes with $\phi < 0.9$ to be outside of the drop. As in [33], spatial convergence of the method is evaluated by measuring $E(\Delta p)$ after one time step for different grid sizes. No splitting error reduction was performed for this test. Results are shown in Figure 1.17.a.

Figure 1.17.b shows the increase in the discrete maximum norm of the velocity (i.e. an increase of the magnitude of parasitic current) under grid refinement. A similar effect was described in [33] for some other implementations of the CSF approach. No quantitative results on parasitic currents for the method were presented in the original paper [96] to compare with, thus their influence on accuracy and stability of the method remains an open question. Figures 1.18-1.19 demonstrate the dependence of pressure and velocity errors after one time step on the values of σ and ρ_l. The long-time evolution of $E(\Delta p)$ for $\rho_l = 10\ kg/m^3$, $\sigma = 1\ N/m$, and different values of $h_x = h_y = h$ is shown in Figure 1.20. The thickness of the interface is also expected to play a role in the accuracy of the approximation. As is shown in Table 1.4, a thinner interface (smaller value of ϵ) leads to a smaller error in the pressure jump.

For all values of the parameters considered above, errors remain small. The method is shown to be convergent in space in terms of the pressure error, thus the Laplace law is reproduced by the algorithm. Further investigation of parasitic currents and the influence of various parameters of the problem, as well as the amount of artificial dissipation, on long-time accuracy and stability of the method is an interesting problem. Therefore, the test demonstrated a potential of coupling the surface tension formulation from [96] with the interface sharpening technique from [110], as well as a need for more detailed studies of its properties and the effects of its coupling with various discretization and stabilization methods, and interface treatments under different conditions.

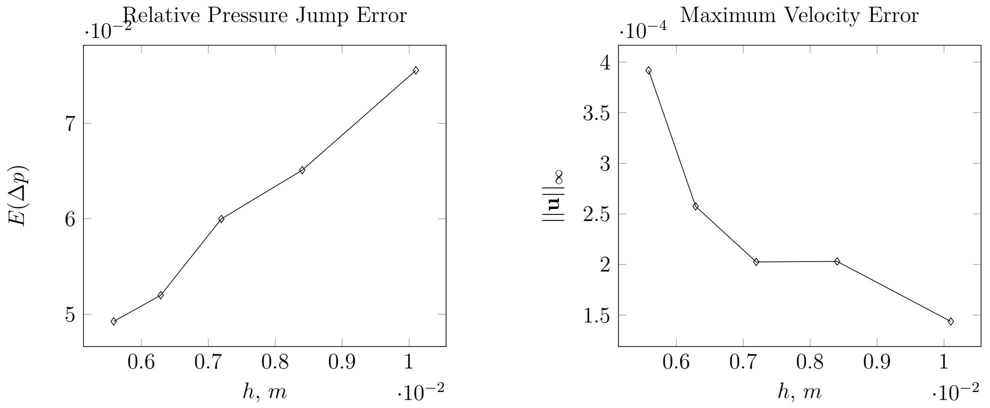

Figure **1.17.a**

Figure **1.17.b**

Figure 1.17: $E(\Delta p)$ and $||\mathbf{u}||_\infty$ after one time step on different grids. $\sigma = 1$ N/m, $\rho_l = 100$ kg/m^3, $\tau = 10^{-5}$ s.

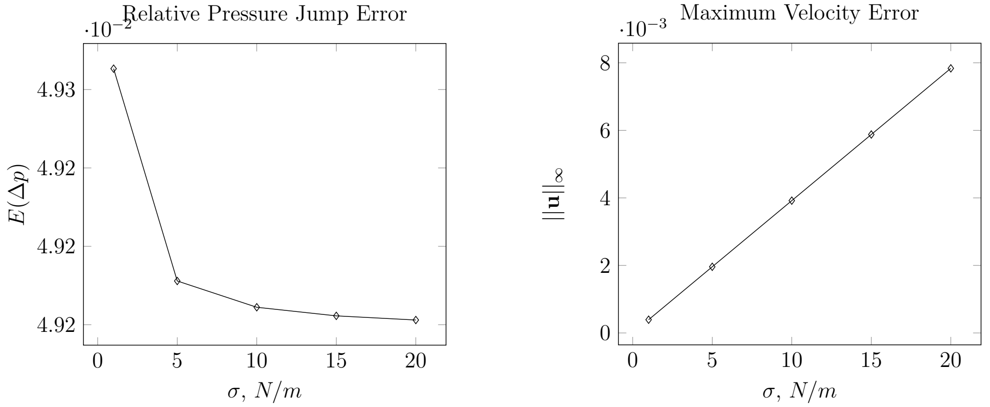

Figure **1.18.a**

Figure **1.18.b**

Figure 1.18: $E(\Delta p)$ and $||\mathbf{u}||_\infty$ after one time step for different σ. $h = 5.6 \cdot 10^{-3}$ m, $\rho_l = 100$ kg/m^3, $\tau = 10^{-5}$ s.

ϵ	$E(\Delta p)$	$\|\|\mathbf{u}\|\|_\infty$
0.5	0.0755	$1.79 \cdot 10^{-5}$
1	0.0793	$4.1 \cdot 10^{-4}$
1.5	0.1149	$2.4 \cdot 10^{-4}$

Table 1.4: $E(\Delta p)$ and $||u||_\infty$ for different values of ϵ (i.e. for different interface thickness).

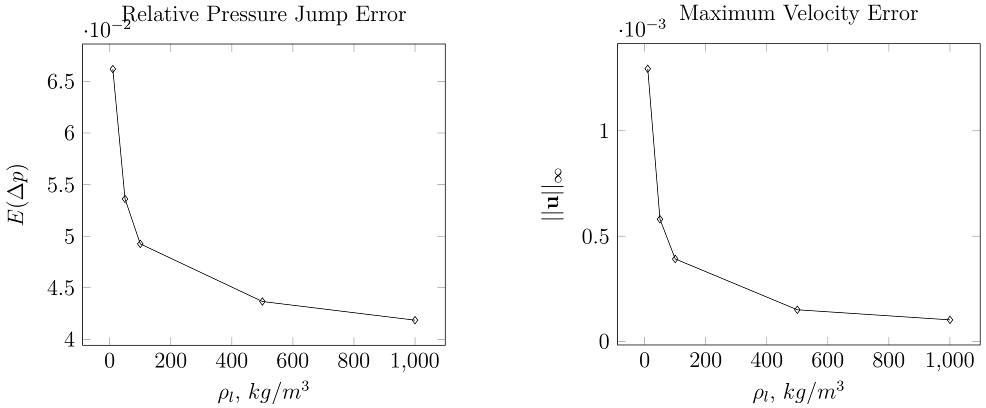

Figure **1.19.a**

Figure **1.19.b**

Figure 1.19: $E(\Delta p)$ and $||\mathbf{u}||_\infty$ after one time step for different ρ_l. $h = 5.6 \cdot 10^{-3}$ m, $\sigma = 1$ N/m, $\tau = 10^{-5}$ s.

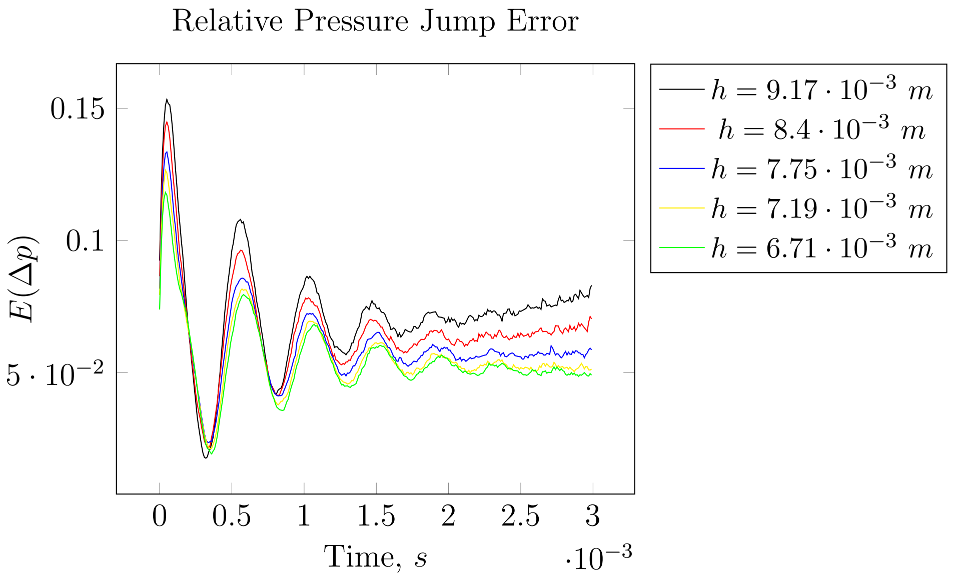

Figure 1.20: Time evolution of the Relative Pressure Jump Error with different grid sizes. In all the tests $\tau = 10^{-5}$ s, $\sigma = 1$ N/m, $\rho_l = 10$ kg/m^3.

1.4.8 One-Dimensional Shock Wave-Interface Interaction.

Next, a one-dimensional shock-interface interaction is studied using a test case from [78] and [19]. This test verifies the performance of the GP-stabilization in the multicomponent case, as well as the robustness of the interface capturing approach. A strong (Mach 8.96) shockwave is travelling in helium ($\gamma = 1.667$, $\pi^{\infty} = 0$) towards a material interface with air ($\gamma = 1.4$, $\pi^{\infty} = 0$). Both materials are assumed to be inviscid. Following [61], non-dimensional initial conditions are given by:

$$(\rho, u, p, \phi) = \begin{cases} (0.386, 26.59, 100, 1) & \text{if } 0 \leq x < 0.2 \\ (0.1, -0.5, 1, 1) & \text{if } 0.2 \leq x < 0.8 \\ (1, -0.5, 1, 0) & \text{if } 0.8 \leq x < 2. \end{cases} \tag{1.13}$$

Such problems are known to be challenging, since interface sharpening methods often lead to miscomputations of shock positions and speeds (see [19]). Here, a numerical solution at $T_{end} = 0.07$ ($\tau = 1 \cdot 10^{-5}$, $h = 10^{-3}$) is compared with the exact one (derived in [78]). The GP-dissipation term is employed to stabilize the solution. The interface sharpening procedure is used with the parameter $\epsilon = 0.5$. The results are present in Figure 1.21.

As can be seen, the interface remains very sharp, although density appears to be smeared in the regions adjacent to the artificial mixing layer. Shock positions and strengths are computed correctly and no oscillations are present, which validates both the interface capturing technique and the GP-stabilization term, as well as the general discretization strategy used in this study. So, the present method can be used to compute shock-interface interaction problems.

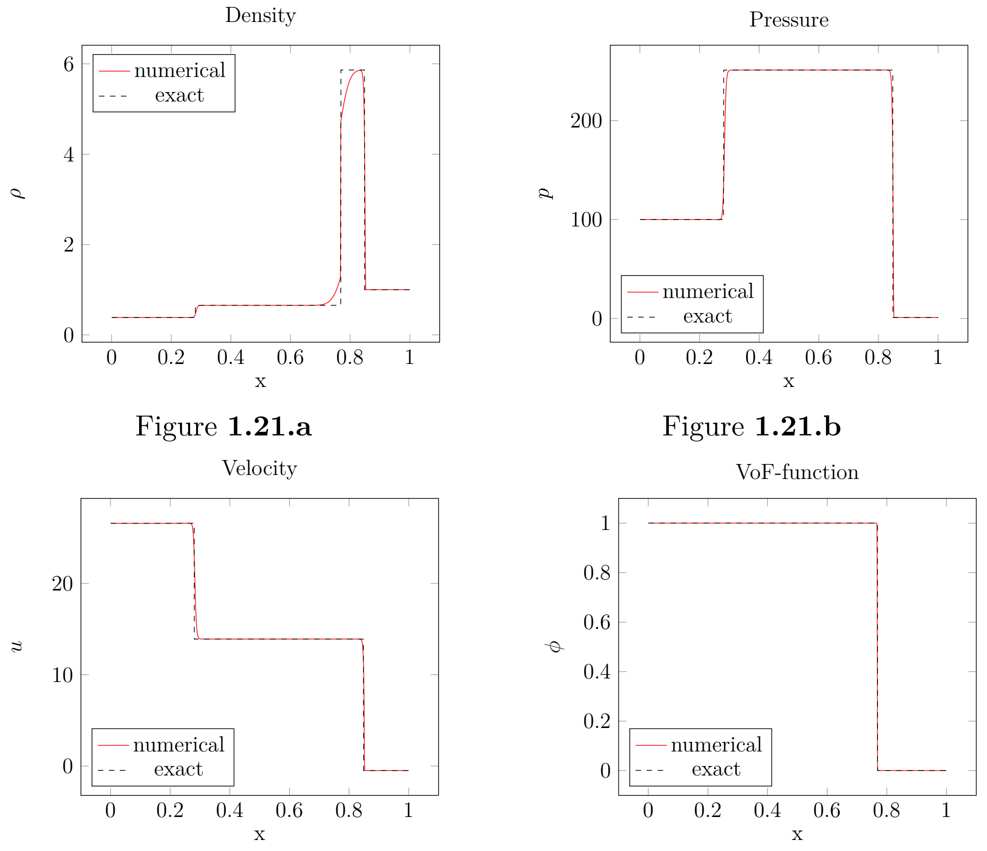

Figure **1.21.a**

Figure **1.21.b**

Figure **1.21.c**

Figure **1.21.d**

Figure 1.21: Density, pressure, velocity and VoF-function at $T_{end} = 0.07$. Exact and numerical ($\tau = 1 \cdot 10^{-5}$, $h = 10^{-3}$) solutions.

1.4.9 Shock Wave Refraction.

Shock-interface interactions in two dimensions are known to produce nontrivial refraction patterns and they may serve as good tests for multidimensional, multicomponent algorithms at high Mach numbers. Following [91], a two-dimensional shock-interface interaction is studied using the Euler equations (no surface tension, viscous and heat transfer effects). In a square $[1m \times 1m]$ domain, a normal incident shock wave is propagating to the left in water ($\gamma = 4.4$, $\pi_\infty = 6 \cdot 10^8\ Pa$) and impacting a planar material interface with air ($\gamma = 1.4$), inclined on an angle β.

The following pre- and post-shock conditions are specified for water in terms of the primitive variables (p, ρ, u, v) in $(Pa, kg/m^3, m/s, m/s)$ (similar to [91]):

$$(p, \rho, u, v)_{pre} = (1 \cdot 10^5, 1000, 0, 0) \tag{1.14}$$

$$(p, \rho, u, v)_{post} = (1.9 \cdot 10^9, 1323.65, -681.58, 0), \tag{1.15}$$

and the initial state of air is specified as:

$$(p, \rho, u, v)_{air} = (1 \cdot 10^5, 1, 0, 0). \tag{1.16}$$

Initial fields of ρ for angles $\beta = \frac{\pi}{6}, \frac{\pi}{4.5}, \frac{\pi}{3.6}, \frac{\pi}{2.5}$ are shown in Figure 1.22.

Non-reflection boundary conditions (proposed in [119]) are prescribed explicitly at the left, right, and top boundaries of the domain and symmetry boundary conditions are used at the bottom. The computations are performed on a 800×800 uniform grid, with $\tau = 10^{-7}\ s$. No splitting error reduction was used for this test.

Figures 1.23-1.26 represent $\log p$, ϕ, and ρ at $T_{end} = 1.5 \cdot 10^{-4}\ s$. The results are in good agreement with those from [91]. The material interface remains thin throughout the computations ($\epsilon = 1$), and the refraction patterns

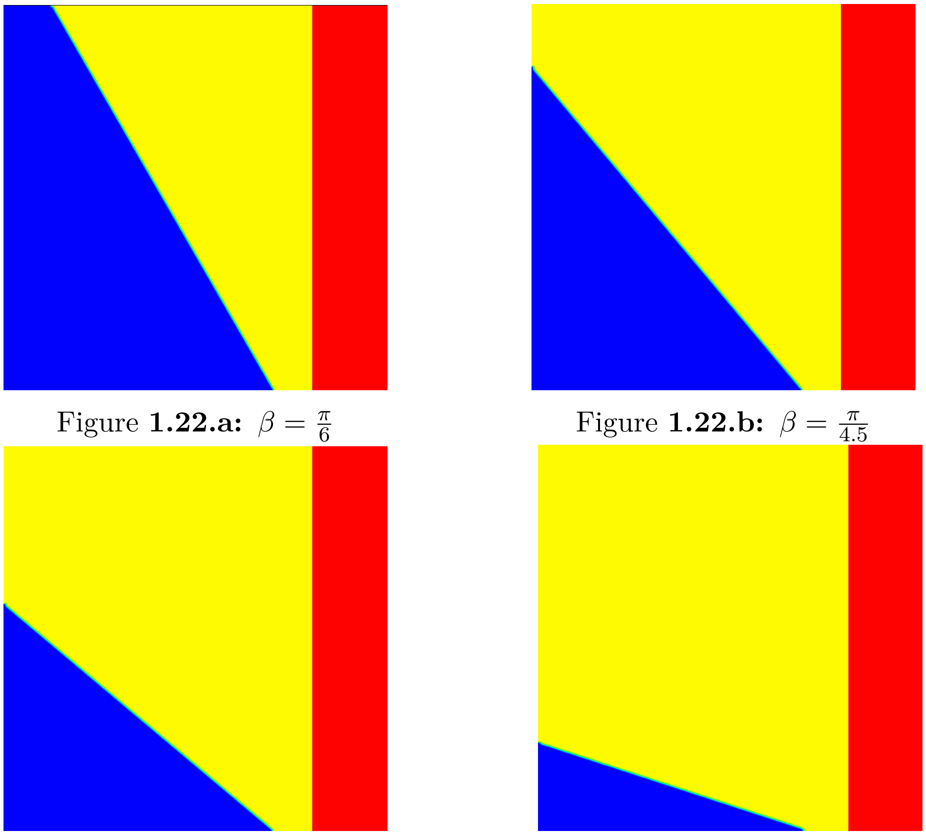

Figure **1.22.a:** $\beta = \frac{\pi}{6}$

Figure **1.22.b:** $\beta = \frac{\pi}{4.5}$

Figure **1.22.c:** $\beta = \frac{\pi}{3.6}$

Figure **1.22.d:** $\beta = \frac{\pi}{2.5}$

Figure 1.22: Initial ρ-fields (pseudocolor) for different values of β. Red - water post-shock state, yellow - water pre-shock state, blue - air.

are captured correctly. The cases of $\beta = \frac{\pi}{6}, \frac{\pi}{4.5}, \frac{\pi}{3.6}$ produce regular refraction patterns with reflected expansion (RRE), while the case of $\beta = \frac{\pi}{2.5}$ gives rise to irregular concave forward refraction (CFR), as the impacted wave interacts with the reflected expansion, "which results in a mutual annihilation and formation of a compound wave" (see [91]). The compound wave is curved due to the speed difference between the incident shock and the refraction node (intersection of the compound and transmitted waves), and weaker than the impacting shock. In all the test cases a water jet is formed by the convergence of shock-induced flows, which is also in agreement with the results of [91]. Therefore, the algorithm has demonstrated its ability to simulate multidimensional inviscid flows featuring shock-interface interactions without topological changes and successfully capture nontrivial refraction patterns.

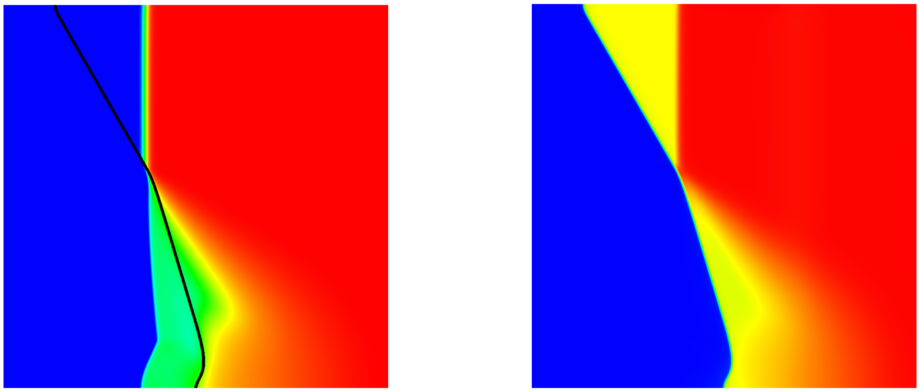

Figure **1.23.a:** $\log p$ and ϕ

Figure **1.23.b:** ρ

Figure 1.23: $\log p$ (pseudocolor), ϕ (contour), and ρ (pseudocolor) plots at $T_{end} = 1.5 \cdot 10^{-4}$ s for $\beta = \frac{\pi}{6}$

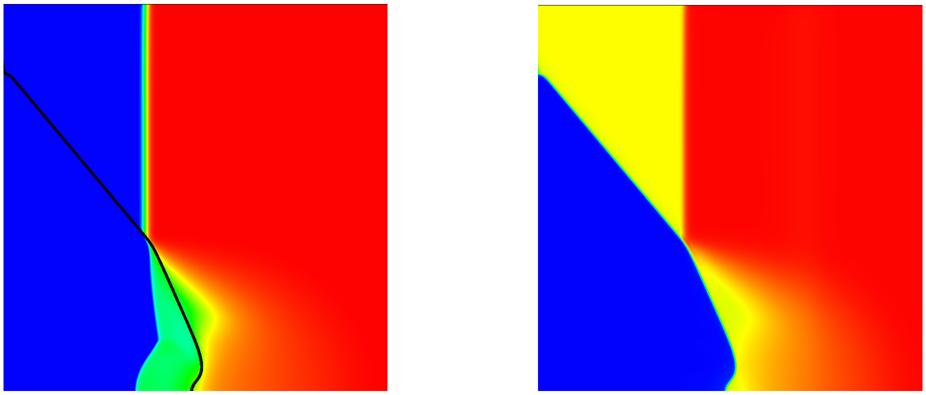

Figure **1.24.a:** $\log p$ and ϕ

Figure **1.24.b:** ρ

Figure 1.24: $\log p$ (pseudcolor), ϕ (contour), and ρ (pseudocolor) plots at $T_{end} = 1.5 \cdot 10^{-4}$ s for $\beta = \frac{\pi}{4.5}$

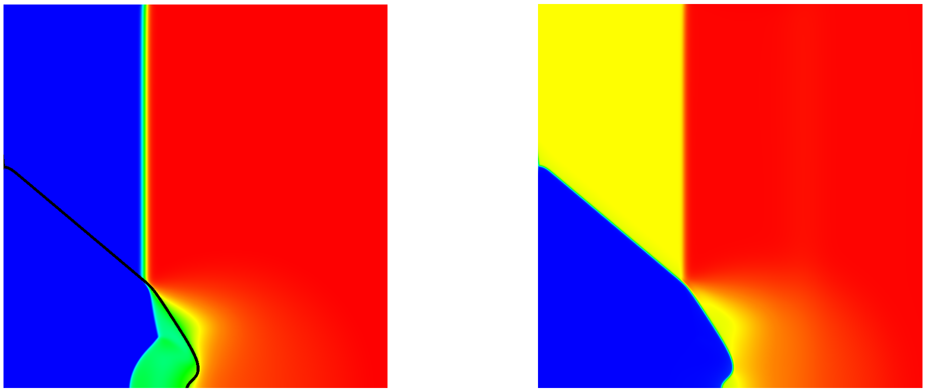

Figure **1.25.a:** $\log p$ and ϕ

Figure **1.25.b:** ρ

Figure 1.25: $\log p$ (pseudcolor), ϕ (contour), and ρ (pseudocolor) plots at $T_{end} = 1.5 \cdot 10^{-4}$ s for $\beta = \frac{\pi}{3.6}$

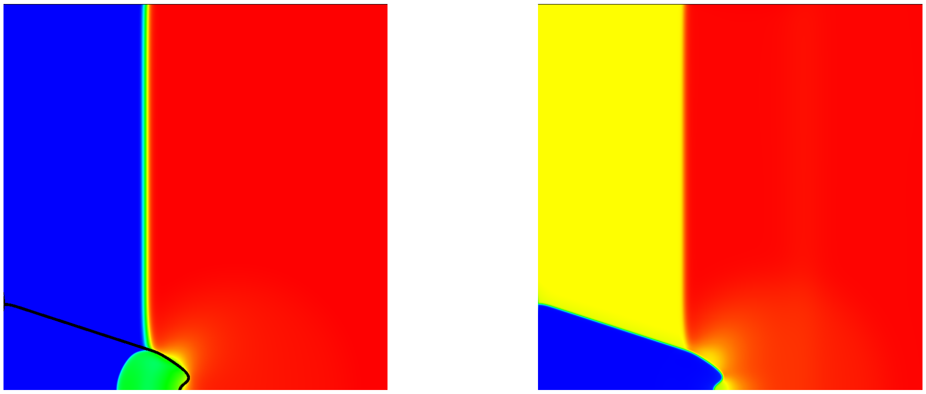

Figure **1.26.a:** $\log p$ and ϕ

Figure **1.26.b:** ρ

Figure 1.26: $\log p$ (pseudcolor), ϕ (contour), and ρ (pseudocolor) plots at $T_{end} = 1.5 \cdot 10^{-4}$ s for $\beta = \frac{\pi}{2.5}$

1.4.10 Shock Wave-Bubble Interaction.

Finally, a shock-curved interface interaction test is considered to validate the applicability of the method to this class of problems and demonstrate the possibility of handling topological changes. In this test case the same materials and the same initial conditions (pre-, post-shock states, location of the shock wave) are used as in the previous test, but the interface forms an air bubble of radius $R = 0.2m$ centered at $(0.55m, 0.5m)$. The full system of NSEs is considered here, with $\mu_{water} = 10^{-3}$ $kg/(s \cdot m)$, $\mu_{air} = 10^{-5}$ $kg/(s \cdot m)$, $\sigma = 0.073$ N/m. Interface sharpening is performed with $\epsilon = 1.5$. Initial conditions for air are given by:

$$(p, \rho, u, v)_{air} = (1 \cdot 10^5 + \frac{\sigma}{R}, 1, 0, 0). \tag{1.17}$$

The computations are performed on a 800×800 uniform grid, with $\tau = 10^{-7}$ s. No splitting error reduction was used for this test. The fields ρ and ϕ are presented at different times in Figures 1.27-1.30. The results are in good agreement with [110], where a similar test case was presented. The interface remains thin throughout the computations, and the effects of wave refraction and eventual topological change (bubble collapse) with the formation of a high-speed jet are well-captured. This test reveals the potential of the scheme to solve a full system of multicomponent NSEs in the high-Mach number regime, as well as the ability of the methods to handle topological changes of the interface.

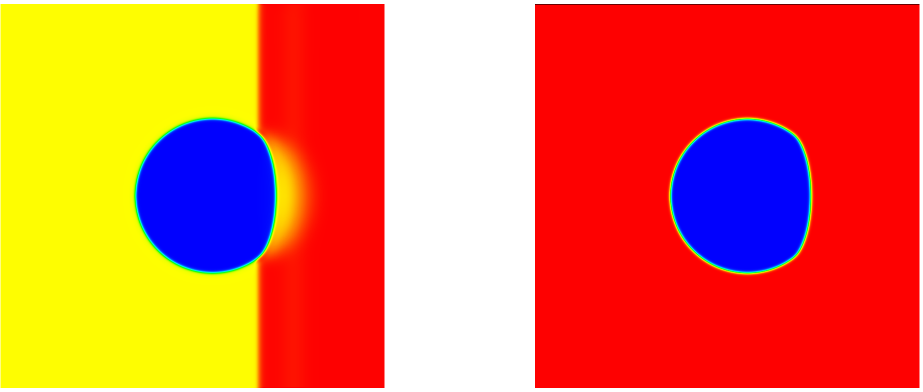

Figure **1.27.a:** ρ

Figure **1.27.b:** ϕ

Figure 1.27: ρ and ϕ pseudocolor fields at $t = 4.5 \cdot 10^{-5}\ s$

1.5 Conclusion

In this chapter, an efficient numerical algorithm applicable to a wide range of compressible multicomponent flows has been built based on the linearized block implicit (LBI) factored schemes. The main contribution of this study to the development of such schemes is the splitting error reduction technique, which enables their use for low-Mach number flows.

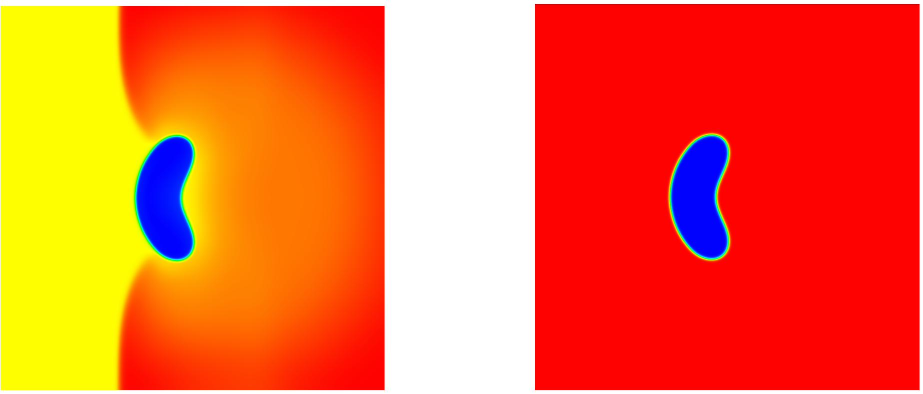

Figure **1.28.a:** ρ

Figure **1.28.b:** ϕ

Figure 1.28: ρ and ϕ pseudocolor fields at $t = 1.75 \cdot 10^{-4}$ s

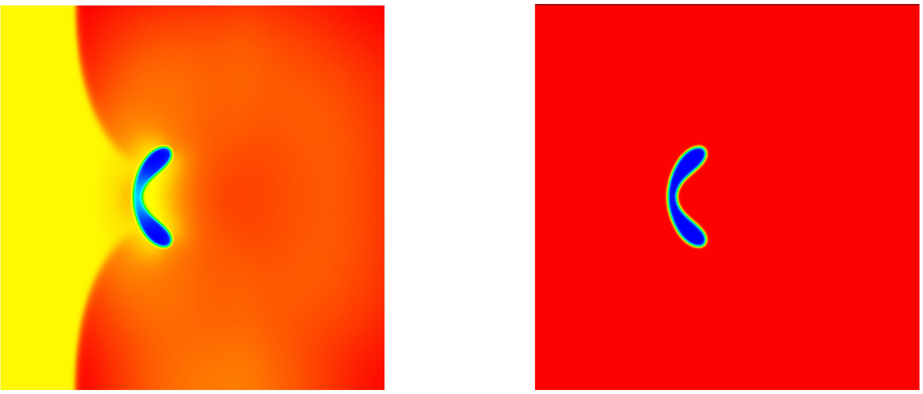

Figure **1.29.a:** ρ

Figure **1.29.b:** ϕ

Figure 1.29: ρ and ϕ pseudocolor fields at $t = 2.16 \cdot 10^{-4}$ s

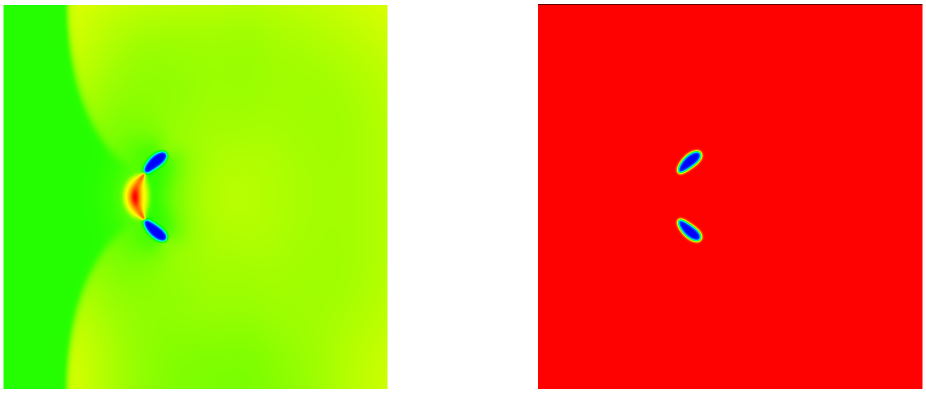

Figure **1.30.a:** ρ

Figure **1.30.b:** ϕ

Figure 1.30: ρ and ϕ pseudocolor fields at $t = 2.276 \cdot 10^{-4}$ s

An artificial dissipation term for high-Mach number applications has been constructed as a finite difference interpretation of a novel finite element technique from [48], which introduces artificial viscosity coefficients based on the maximum speed of propagation in local one-dimensional Riemann problems. The estimation algorithm for the maximum speed of propagation proposed in [47] has been extended here to the case of the stiffened gas equation of state. The results presented in this chapter show the advantages of the approach proposed in [48] and continue the development of this class of schemes.

Another contribution of this work is the novel artificial dissipation term designed for low-Mach number applications. It is based on second-order differences of conservative variables and products of conservative variables and the Jacobians. Such a combination is shown to be effective in eliminating the odd-even decoupling problem while controlling the total kinetic energy of the system. This allows for the use of LBI factored schemes in terms of conservative variables on non-staggered grids for low-Mach number applications.

A consistent coupling of this technique with the interface-capturing approach without disrupting the interfacial equilibrium is achieved by introducing functions of EOS coefficients as variables, as well as the special design of the stabilization terms. The sharpening technique from [110] is shown to be compatible with the method, and with the CSF implementation of the surface tension from [96].

Possible extensions and directions for further research may include the use of more accurate time-marching methods or a defect-correction technique to lift the order of accuracy of the algorithm, incorporating implicit non-reflection boundary conditions, and extending the method to the three-dimensional case and more complicated geometries.

Chapter 2

A direction splitting scheme for the Navier-Stokes-Boussinesq system in spherical shell geometries

2.1 Introduction

This chapter presents a new direction splitting scheme for solving the incompressible Navier-Stokes-Boussinesq system:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \mathrm{Pr}\,\Delta\mathbf{u} &= \mathbf{g}\,\mathrm{Pr}\,\mathrm{Ra}T \text{ in } \ \Omega\times(0,T_f] \\ \nabla\cdot\mathbf{u} &= 0 \text{ in } \ \Omega\times(0,T_f] \\ \mathbf{u} &= \mathbf{0} \text{ on } \ \partial\Omega\times(0,T_f] \end{aligned} \tag{2.1}$$

$$\begin{aligned} \frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)T - \Delta T = 0 \text{ in } \ \Omega\times(0,T_f] \\ T = 0 \text{ on } \ \partial\Omega\times(0,T_f] \end{aligned} \tag{2.2}$$

in a spherical shell domain that can be defined in terms of a spherical coordinate triple (r,θ,ϕ) as:

$$\Omega = \left\{(r,\theta,\phi)\in[R_1,R_2]\times[0,\pi]\times[0,2\pi)\right\}.$$

In the above, $\mathbf{g}$ is the unit vector in the direction of gravity, and Pr,Ra are the Prandtl and Rayleigh numbers, respectively, $R_1 >> 0$. The system (2.1)-(2.2) models the flow of a heat-conducting fluid, under the assumption that the temperature-induced density variation influences significantly only the buoyancy force and the fluid remains incompressible. It is widely applied to model the flow in the atmospheric boundary layer ([81]), oceanic flows ([112]), as well as, if combined with an equation for the magnetic field, the flow in the Earth's dynamo ([103]) (see a more detailed discussion in Section 2.2.1). Even though for most of the discussion, we assume homogeneous Dirichlet boundary conditions on the two spherical surfaces $r = R_1, r = R_2$, the approach is applicable to Neumann and Robin boundary conditions as well.

One possible method for numerical approximations of differential equations in spherical shell geometries is based on the use of a spherical transformation to transform the domain into a parallelepiped. The obvious advantage of this approach is the simple computational domain, which allows for the use of structured grids and the efficient schemes developed for them. Moreover, the grid can naturally follow the geometry of the domain, without requiring too many cells, as would possibly be in the case of a Cartesian formulation. However, the singularity of the transformation and the grid convergence near the poles have for many years been a difficulty in the development of accurate numerical schemes. Several different treatments have been proposed for dealing with these problems. For example, in [55], the pole singularity issue is avoided by replacing the equations at the poles with equations analogous to boundary conditions, while in [89] a redefinition of the singular coordinates is proposed. Other suggested approaches include applying L'Hospital's rule [40] to singular terms and switching to Cartesian formulation around the poles [34]. On the other hand, the grid convergence has been a more serious problem. In particular, it produces a solution with an uneven resolution, requires very small time steps for explicit or IMEX schemes since the time step size is limited by the minimum grid size, and causes convergence problems for iterative solvers. Therefore, different grid systems with quasi-uniform resolution that avoid the grid convergence problems have been suggested in the literature. One such approach is the "cubed sphere" of [98], which is a grid that covers a spherical surface with six components corresponding to six faces of a cube. Even though the resulting grid is quasi-uniform, it still has singularities at the corner points of the faces and it is non-orthogonal. Some of the other suggested unstructured grids include the icosahedral grid of [8] and the non-orthogonal rhombohedral grid of [129].

In this study we adapt an alternative method proposed by [63], employing the so-called Yin-Yang grids. Some advantages of the Yin-Yang approach are that the metric tensors are simple, the resolution is quasi-uniform, and it requires modest programming efforts for extending the code from a single latitude-longitude grid, and thus it is better suited for high-performance numerical methods we propose in this chapter. It starts with a decomposition of the domain into two overlapping subdomains, combined with two different spherical transforms whose axes are perpendicular to each other, cf. Fig. 2.1. As a result, both subdomains are transformed into identical parallelepipeds that can be gridded with the same uniform meshes. This approach automatically removes the transform's singularities at the poles, at the expense of the introduction of two subdomains, so that the two local solutions must be coupled employing the Schwarz-type iterations (see Section 2.2.2 for details on the Schwarz domain decomposition iterative methods). The Yin-Yang grid has been used for simulations of mantle convection [115], core collapse supernovae [125], atmospherical general circulation model [7] and visualization in spherical regions [93]. The main novelty of this work is that the Yin-Yang domain decomposition is combined with a direction splitting time discretization that, in the case of linear parabolic equations, is unconditionally stable in the spherically transformed domains. The advection can be included either in an IMEX fashion or by including the linearized advection into the entire operator that is further split direction-wise. The resulting splitting scheme is conditionally stable, since the direction-wise operators are not positive, but our numerical experience demonstrated that the second approach yields an algorithm that has better stability performance. This is why the rest of this chapter concerns only this type of schemes. To our knowledge, the stability of the direction splitting approach has not been rigorously studied in the context of spherical coordinate systems. Therefore, we prove below that it is unconditionally stable in the case of a scalar heat equation, in a simply shaped domain (in terms of spherical coordinates). The case of the full Navier-Stokes-Boussinesq system is more involved and we do not provide rigorous proof here. However, our numerical experience shows that the direction splitting is still unconditionally stable if the advection terms are omitted and if the velocity-pressure decoupling is done via the AC method proposed in [44].

The rest of this chapter is organized as follows. In the next section, preliminary theoretical background and literature review is given on the Navier-Stokes-Boussinesq system (Section 2.2.1), Schwarz domain decompositions methods (Section 2.2.2), and the artificial compressibility approximation of the incompressible Navier-Stokes equations (Section 2.2.3). Then we briefly recall the definition of the Yin-Yang domain decomposition in Section 2.3.1. In Section 2.3.2 we present the numerical scheme for the advection-diffusion and Navier-Stokes equations on each of the subdomains. In Section 2.3.3, we discuss the implementation details, and in Section 2.4 we present the numerical experiments.

2.2 Theoretical background.

2.2.1 Navier-Stokes-Boussinesq system.

The Navier-Stokes-Boussinesq system (2.1)-(2.2), also known as the Boussinesq or Oberbeck-Boussinesq approximation (see the original papers [11] and [92]), describes the thermal response of a linearly viscous fluid that is mechanically incompressible but thermally compressible ([97]). In other words, density variations are neglected in the mass and momentum conservation equations except when they are coupled to the gravitational acceleration in the buoyancy force ([113]), where the density changes exist due to the caused temperature changes that disrupt the hydrostatic equilibrium ([30]). Such flows are common in natural and industrial processes, and the applications of the Boussinesq approximation include solar convection, magnetoconvection, and magnetic buoyancy, mantle convection, atmospheric lee waves, Rayleigh–Bénard instabilities, oceanic general circulation, buoyancy-driven flows in crystal-growth melts, environmental fluid mechanics, fluid-dynamical problems in Galaxies, and many others (see [128] and references there for more examples).

A rigorous justification of the Boussinesq approximation from the point of view of continuum mechanics was given in [97]. Correcting some of the shortcomings and inconsistencies of previous studies, authors of [97] start with the mass, momentum, and energy conservation equations (i.e. the Navier-Stokes system) supplemented by the second law of thermodynamics in form of the Clausius-Duhem inequality, and find a proper non-dimensional small parameter (namely $\epsilon = \dfrac{U}{(g\nu)^{\frac{1}{2}}}$, where U is the representative velocity, g is the gravitational acceleration, and ν is the kinematic viscosity) with which to perturb the physical quantities. An expansion of velocity, temperature, and pressure in terms of ϵ leads to the Oberbeck-Boussinesq system as an approximation to the full Navier-Stokes equations to the order ϵ^4.

In a related work [62], the Oberbeck-Boussinesq system is obtained as a constitutive limit of the full Navier-Stokes system as the thermal expansion and the isothermal compressibility coefficients approach zero, provided that the weak solutions of the full thermo-mechanical system satisfying a uniform estimate exist. Under certain assumptions on the approximation parameters, this approach also allows one to recover the results from [97].

A different approach is taken in [30], where the Oberbeck-Boussinesq system is derived as a singular limit of the Navier-Stokes-Fourier system (compressible Navier-Stokes equations coupled with the Fourier law of heat conductivity) as both the Mach number and the Froude number (ratio of inertial and gravitational forces) tend to zero. The convergence is shown in the case of a large time interval and for ill-prepared initial data.

2.2.2 Schwarz domain decomposition methods.

The Yin-Yang grid described below can be viewed as a composite overlapping mesh consisting of logically rectangular curvilinear grids with their union covering the whole domain (see [16] for a detailed description and analysis of this type of meshes). Each of these grids covers a part of the domain (a spherical shell, in our case) thus dividing the region into two overlapping subdomains and allowing one to use the Schwarz domain decomposition methodology to approximate the solution of the original system of PDEs (see e.g. [127]). Here we follow the description from [83] to introduce the basics of the Schwarz method. Readers are referred to [83] and references there for a more comprehensive description.

In general, domain decomposition formulations can be of two types: overlapping and non-overlapping. Several methods (also called hybrid formulations) exists to deal with each of the two settings, e.g. the Steklov-Poincare and the Lagrange multiplier formulations require a non-overlapping decompositions, while the Schwarz methods are based on overlapping domains. Regardless of the type of the domain decomposition chosen, a hybrid formulation is a coupled system of equations which is equivalent to the original system with true solutions on each subdomain as unknowns. For the purpose of illustration of basic ideas of the Schwarz method, we follow the steps from [83] and consider the case of a 2nd order elliptic PDE (note that the extension of the method to the case of other types of PDEs is straight-forward):

$$L(u) = \ \nabla \cdot (a(x)\nabla u) + \mathbf{b}(x) \cdot \nabla u + c(x)u = f, \text{ in } \Omega \tag{2.1}$$

$$u = 0, \text{ on } \partial\Omega \tag{2.2}$$

for $\Omega \subset \mathbb{R}^d$. The coefficient $a(x)$ is assumed to satisfy

$$0 < a_0 \leq a(x), \ \forall x \in \Omega,$$

while $\mathbf{b}(x)$ and $c(x) \geq 0$ are smooth, and $f(x) \in L^2(\Omega)$. Let us introduce some important definitions required for the formulation of the Schwarz domain decomposition methodology:

Definition 2.2.1 *(from [83], Chapter 1)*
A collection of two open subregions $\Omega_i \subset \Omega$ for $i = 1, 2$ will be referred to as an overlapping decomposition of Ω if the following holds:

$$\Omega_1 \cup \Omega_2 = \Omega.$$

Boundaries of the subdomains will be denoted $B_i \equiv \partial\Omega_i$ and their interior and exterior segments by $B^{(i)} \equiv \partial\Omega_i \cap \Omega$ and $B_{[i]} \equiv \partial\Omega_i \cap \partial\Omega$, respectively.

Note that the number of subdomains is limited by two for simplicity, and an extension to the case of more subdomains is trivial.

Definition 2.2.2 *(from [83], Chapter 1)*
A partition of unity subordinate to the overlapping domains Ω_1 and Ω_2 consists of smooth functions $\chi_1(x)$ and $\chi_2(x)$ satisfying:

$$\chi_i(x) \geq 0, \text{ in } \overline{\Omega}_i,$$

$$\chi_i(x) = 0, \text{ in } \Omega \backslash \overline{\Omega}_i,$$

$$\chi_1(x) + \chi_2(x) = 1, \text{ in } \overline{\Omega}.$$

To define a hybrid formulation for the original problem (2.1) one must ensure that the following requirements are satisfied:

- Consistency: the restriction $u_i(x)$ of the true solution of (2.1) to each subdomain Ω_i must solve the hybrid system, i.e. the solution of hybrid formulation $(\omega_1(x), \omega_2(x))$ must satisfy $\omega_i(x) = u_i(x)$ for $i = 1, 2$.

- Wel-posededness: the solution of the hybrid formulation $(\omega_1(x), \omega_2(x))$

must exist, be unique, and depend continuously on the data. This guarantees the hybrid formulation is stable and uniquely solvable, which is essential for stability of the numerical approximation.

The solutions of the original problem is expressed in terms of the local solutions $\omega_i(x)$ using an appropriate partition of unity as:

$$u(x) = \chi_1(x)\omega_1(x) + \chi_2(x)\omega_2(x).$$

Thus, a local problem will be solved on each subdomain coupled by an appropriate matching condition to ensure that the hybrid formulation is equivalent to the original problem. In the Schwarz framework it leads to the following formulation of the local problems:

$$\begin{cases} L\omega_1 = f, \text{ in } \Omega_1 \\ \\ \omega_1 = \omega_2, \text{ on } B^{(1)} \\ \\ \omega_1 = 0, \text{ on } B_{[1]} \end{cases} \text{and} \begin{cases} L\omega_2 = f, \text{ in } \Omega_2 \\ \\ \omega_2 = \omega_1, \text{ on } B^{(2)} \\ \\ \omega_2 = 0, \text{ on } B_{[2]} \end{cases}$$

(See [83] Chapters 1 and 15 for consistency and well-posedness proofs of the hybrid formulation given above).

Two iterative methods can be employed to solve these local problems, multiplicative (sequential or alternating) Schwarz method, and additive (parallel) Schwarz method. Both are described below.

Definition 2.2.3 *Schwarz Alternating Method (from [83], Chapter 1)*

1. *For* $k = 0, 1, ...,$ *until convergence do:*

2. *Solve for* $\omega_1^{(k+1)}$ *as follows:*
$$\begin{cases} L\omega_1^{(k+1)} = f_1, \ in \ \Omega_1 \\ \ \omega_1^{(k+1)} = v^{(k)}, \ on \ B^{(1)} \\ \ \omega_1^{(k+1)} = 0, \ on \ B_{[1]} \end{cases}$$

Define $v^{k+\frac{1}{2}}$ *as follows:*

$$v^{k+\frac{1}{2}} = \begin{cases} \omega_1^{(k+1)} \ on \ \Omega_1, \\ v_1^{(k)} \ on \ \Omega \backslash \Omega_1. \end{cases}$$

3. *Solve for* $\omega_2^{(k+1)}$ *as follows:*
$$\begin{cases} L\omega_2^{(k+1)} = f_2, \ in \ \Omega_2 \\ \ \omega_2^{(k+1)} = v^{(k+\frac{1}{2})}, \ on \ B^{(2)} \\ \ \omega_2^{(k+1)} = 0, \ on \ B_{[2]} \end{cases}$$

Define v^{k+1} *as follows:*

$$v^{k+1} = \begin{cases} \omega_2^{(k+1)} \ on \ \Omega_2, \\ v_1^{(k+\frac{1}{2})} \ on \ \Omega \backslash \Omega_2. \end{cases}$$

4. Endfor

Output: $v^{(k)}$

Definition 2.2.4 *Parallel Schwarz Method (from [83], Chapter 1)*

1. For $k = 0, 1, ...,$ *until convergence do:*

2. *For* $i = 1, 2$ *determine* $\omega_i^{(k+1)}$ *in parallel:*

```
        $\begin{cases} L\omega_i^{(k+1)} = f,\ in\ \Omega_i \\ \omega_i^{(k+1)} = \chi_1(x)\omega_1^{(k)}(x) + \chi_2(x)\omega_2^{(k)}(x),\ on\ B^{(i)} \\ \omega_i^{(k+1)} = 0,\ on\ B_{[i]} \end{cases}$
  3.    Endfor
  4. Endfor
Output: $(\omega_1^{(k+1)}, \omega_2^{(k+1)})$
```

If $c(x) \geq c_0 > 0$ and there is sufficient overlap, the iterates $v^{(k)}$ defined by

$$v^{(k)} \equiv \chi_1(x)\omega_1^{(k)}(x) + \chi_2(x)\omega_2^{(k)}(x)$$

will converge geometrically to the solution of (2.1) (see [83], Chapter 15). Note that in practice a discrete version of these algorithms is employed, in particular an interpolation is required for non-matching meshes (such as the Yin-Yang grid used in this work) to impose the conditions $\omega_1^{(k+1)} = v^{(k)}$ on $B^{(1)}$ and $\omega_2^{(k+1)} = v^{(k+\frac{1}{2})}$ on $B^{(2)}$.

These methods can be easily extended to the case of time-dependent problems, where the subdomain iterations (multiplicative or additive) are performed at each time step until convergence.

2.2.3 Artificial compressibility.

The artificial compressibility methods used in this chapter is a direct implementation of the scheme described in [44]. This subsection follows [44] to give more background on the family of the artificial compressibility methods and the theory behind the approach from [44]. See [44], and [37] Section 23 for more details.

One of the main difficulties in solving the incompressible Navier-Stokes system is the coupling of pressure and velocity. A regularization of the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$ may be performed to resolve the issue, e.g. by introducing a penalty parameter $\epsilon > 0$ and replacing the constraint with $\epsilon p + \nabla \cdot \mathbf{u} = 0$, as it was proposed in [117]. Such perturbation allows one to eliminate pressure from the momentum equation and thus the only equation that needs to be solved is

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \nabla^2 \mathbf{u} - \frac{1}{\epsilon} \nabla \nabla \cdot \mathbf{u} = 0,$$

which is similar to the compressible momentum conservation equation. If we choose $\epsilon = \tau$ to make the scheme first-order accurate in time, the PDE that has to be solved at each time step takes the form of $\mathbf{u} - \nabla\nabla \cdot \mathbf{u} + l.o.t = \mathbf{g}$. The discrete variant of this system has a condition number that scales as $\mathcal{O}(h^{-2})$, while some alternative first-order schemes have a $\mathcal{O}(h^{-1})$ condition number. Higher order versions of this scheme would require taking $\epsilon = \tau^l$ which yields a discrete system with condition number $\mathcal{O}(\tau^{1-l} h^{-2})$, which is impractical for $l \geq 2$.

The described penalty regularization of the $\nabla \cdot \mathbf{u} = 0$ constraint can be improved by using the time derivative of pressure in the regularization:

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

This particular perturbation can be justified by considering the compressible mass conservation in the low Mach number limit with $\epsilon = M^{-2}$. This, or similar regularizations have been proposed by several authors and used in combinations with explicit, and various direction splitting schemes (see [118], [18], and [71] Section 9, Chapter VI). If $\epsilon \sim \tau$, this method has several appealing properties, such as uniform stability with respect to ϵ at the continuous level, and, if combined with implicit time-discretization, reasonable computational cost and efficiency that are superior comparing to the non-incremental projection methods (see [44] for details). One limitation is that a stable and efficient second-order extension of this strategy cannot be achieved by taking $\epsilon = \tau^2$ and considering the perturbation of the incompressibility constraint with the second derivative of pressure $\epsilon \dfrac{\partial^2 p}{\partial t^2} + \nabla \cdot \mathbf{u} = 0$, since this perturbation is unstable (see [44] and [107]). Nevertheless, such an extension was proposed in [44] with the help of a bootstrapping technique. This method is adapted in this chapter for spherical coordinates.

The following heuristic argument is given in [44] to motivate the strategy. Let $(\mathbf{u}, p)$ be the solution to the incompressible Navier-Stokes system:

$$\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \nu \Delta \mathbf{u} &= \mathbf{0} \text{ in } \ \Omega \times (0, T_f] \\
\nabla \cdot \mathbf{u} &= 0 \text{ in } \ \Omega \times (0, T_f] \\
\mathbf{u}|_{\Gamma} &= 0 \text{ in } \ (0, T_f] \\
\mathbf{u}|_{t=0} &= \mathbf{u}_0 \text{ in } \ \Omega.
\end{aligned}$$

Let r be some approximation of the exact pressure p. Then one may consider the following problem with $\epsilon > 0$:

$$\frac{\partial \mathbf{w}}{\partial t} + (\mathbf{w} \cdot \nabla)\mathbf{w} + \nabla p - \nu \Delta \mathbf{w} = \mathbf{0},\ \mathbf{w}|_{\Gamma} = 0,\ \mathbf{w}|_{t=0} = \mathbf{w}_0$$
$$\epsilon \frac{\partial (s - r)}{\partial t} + \nabla \cdot \mathbf{w} = 0,\ s|_{t=0} = p_0.$$

Let $\mathbf{e} = \mathbf{u} - \mathbf{w}$ and $\delta = p - s$, then:

$$\begin{gathered} \frac{\partial \mathbf{e}}{\partial t} + (\mathbf{e} \cdot \nabla)\mathbf{e} + \nabla \delta - \nu \Delta \mathbf{e} = \mathbf{0},\ \mathbf{e}|_{\Gamma} = 0,\ \mathbf{e}|_{t=0} = \mathbf{e}_0 \\ \epsilon \frac{\partial \delta}{\partial t} + \nabla \cdot \mathbf{e} = \epsilon \frac{\partial (p - r)}{\partial t},\ \delta|_{t=0} = 0. \end{gathered} \tag{2.3}$$

Now, if r is an $\mathcal{O}(\epsilon^l)$ approximation of p, then $\epsilon \frac{\partial (p - r)}{\partial t} = \mathcal{O}(\epsilon^{l+1})$ and if (2.3) is stable with respect to perturbations in the mass equation (see [44] for the proof), then one should get $\mathbf{e} = \mathcal{O}(\epsilon^{l+1})$ and $\delta = \mathcal{O}(\epsilon^{l+1})$, i.e. the accuracy of $(\mathbf{w}, s)$ is increased by one order of ϵ. Hence, the following family of approximation methods may be considered. Let l be a positive integer, $s_0 = 0$, and consider the following velocity-pressure pairs $(\mathbf{w}_1, s_1), ..., (\mathbf{w}_l, s_l)$ solving:

$$\frac{\partial \mathbf{w}_i}{\partial t} + (\mathbf{w}_i \cdot \nabla)\mathbf{w}_i + \nabla s_i - \nu \Delta \mathbf{w}_i = \mathbf{f},\ \mathbf{w}|_{\Gamma} = 0,\ \mathbf{w}|_{t=0} = \mathbf{u}_0$$
$$\epsilon \frac{\partial (s_i - s_{i-1})}{\partial t} + \nabla \cdot \mathbf{w}_i = 0,\ s_i|_{t=0} = p_0.$$

This method was further analysed and justified in [44]. Its second-order version is described in Section 2.3.2.

2.3 Algorithm.

2.3.1 Spatial discretization and the Yin-Yang grid.

In this section, we briefly recall the definition of the composite Yin-Yang grid following [63]. The grid consists of two identical overlapping latitude-longitude grids whose axes are perpendicular to each other. The Yin grid is based on a spherical transformation

$$\begin{cases} x &= r\sin\theta\cos\phi \\ y &= r\sin\theta\sin\phi \\ z &= r\cos\theta, \end{cases}$$

and covers the region

$$\Omega_1 := \left\{(r,\theta,\phi) \in [R_1,R_2] \times \left[\frac{\pi}{4}-\varepsilon, \frac{3\pi}{4}+\varepsilon\right] \times \left[\frac{\pi}{4}-\varepsilon, \frac{7\pi}{4}+\varepsilon\right]\right\},$$

where $\varepsilon \ll 1$ is a parameter determining the overlap. The Yang grid is obtained via another spherical transformation:

$$\begin{cases} x &= -r\sin\tilde{\theta}\cos\tilde{\phi} \\ y &= r\cos\tilde{\theta} \\ z &= r\sin\tilde{\theta}\sin\tilde{\phi}, \end{cases}$$

such that its axes are perpendicular to the axes of the Yin transform, and it covers the region

$$\Omega_2 := \left\{ \left(r, \tilde{\theta}, \tilde{\phi}\right) \in [R_1, R_2] \times \left[\frac{\pi}{4} - \varepsilon, \frac{3\pi}{4} + \varepsilon\right] \times \left[\frac{\pi}{4} - \varepsilon, \frac{7\pi}{4} + \varepsilon\right]\right\}.$$

The choice of the second axes should be such that the Yang grid fully covers the gap of the Yin one, and the overlapping subregions are of the same size (see Fig. 2.1,). Otherwise, it is identical to the Yin grid modulo two rotations. The resulting Yin-Yang grids are quasiuniform, the coordinate transformations from (r, θ, ϕ) to $(r, \tilde{\theta}, \tilde{\phi})$ and its inverse, as well as the metric tensors on both grids are identical. As a consequence, the methods and codes developed for the standard latitude-longitude grid can be applied to both grids.

2.3.2 Direction splitting time discretization.

Preliminaries

For the rest of this chapter we will frequently make use of the following notations. For a time sequence $w^k, k = 1, 2, \dots$ we denote the average between two time levels as $w^{k+1/2} = (w^{k+1} + w^k)/2$, and the explicit extrapolation to level $k + 1/2$ by $w^{*,k+1/2} = 3w^k/2 - w^{k-1}/2$. For two regular enough functions u, v defined in the spherical shell we denote their weighted L^2 inner product as: $(u, v)_\omega := \int\limits_{R_1}^{R_2} \int\limits_0^\pi \int\limits_0^{2\pi} uv\omega dr d\theta d\phi$, where ω denotes a non-negative weight. In most cases the weight is given by the weight of the spherical transform $\omega = r^2 \sin\theta$, however, in some of the estimates given below, the weight will be appropriately modified. The corresponding norm is given by $\|u\|_\omega^2 = (u, u)_\omega$.

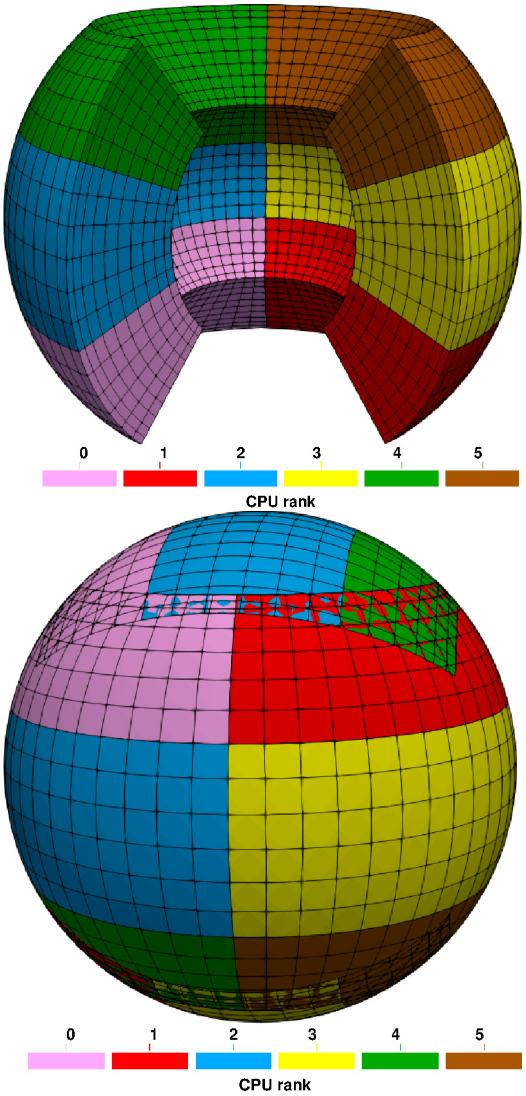

Figure 2.1: Yang (left) and Yin-Yang (right) grids. Each subgrid is further decomposed into blocks for a parallel implementation corresponding to a CPU distribution $1 \times 3 \times 2$.

Direction splitting of the advection-diffusion equation.

Since the PDEs are identical in both domains, it is sufficient to develop the numerical scheme for the Yin domain. Then the Schwarz domain decomposition method can be used to iterate between the subdomains.

We first present a Douglas-type (see [28]) direction splitting scheme for the heat equation. Consider

$$\begin{aligned} \partial_t T - \kappa \Delta T = 0 \text{ in } \ \Omega_1 \times (0, T_f], \\ T = 0 \text{ on } \ \partial\Omega_1 \times (0, T_f], \end{aligned} \tag{2.1}$$

where the Laplacian in spherical coordinates is given by $\Delta = \mathrm{D}_{rr} + \mathrm{D}_{\theta\theta} + \mathrm{D}_{\phi\phi}$,

$$\mathrm{D}_{rr} := \frac{1}{r^2}\partial_r\left(r^2\partial_r\right)$$

$$\mathrm{D}_{\theta\theta} := \frac{1}{r^2\sin\theta}\partial_\theta\left(\sin\theta\partial_\theta\right)$$

$$\mathrm{D}_{\phi\phi} := \frac{\partial_{\phi\phi}}{r^2\sin^2\theta}.$$

The Douglas direction splitting scheme for this equation can be summarized in the following factorized form:

$$\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{rr}\right]\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{\theta\theta}\right]\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{\phi\phi}\right]\frac{\delta T^{n+1}}{\tau} = \Delta T^n, \tag{2.2}$$

where $\delta T^{n+1} := T^{n+1} - T^n$ denotes the first time difference of the time sequence T^k, τ is the time step, and I is the identity operator. We first notice that this splitting can be considered as an Euler explicit scheme whose time difference operator is multiplied by $\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{rr}\right]\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{\theta\theta}\right]\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{\phi\phi}\right]$, that is a consistent perturbation of I and stabilizes the scheme. If the spatial derivative operators are positive and commute with respect to some inner product, the stability of this scheme is not hard to establish.

Unfortunately, $\mathrm{D}_{rr}, \mathrm{D}_{\theta\theta}$, and $\mathrm{D}_{\phi\phi}$ do not commute with respect to the weighted product $(.,.)_\omega$, and their positivity is far from being clear. The main obstacle to the commutativity of the one-dimensional operators comes from the non-constant terms in the denominators of $\mathrm{D}_{\theta\theta}$ and $\mathrm{D}_{\phi\phi}$. Therefore, the scheme (2.2) should be modified as follows. We first introduce the modified spatial operators:

$$\hat{\mathrm{D}}_{\theta\theta} := \frac{1}{R_1^2 \sin\theta}\partial_\theta\left(\sin\theta\partial_\theta\right), \hat{\mathrm{D}}_{\phi\phi} := \frac{\partial_{\phi\phi}}{R_1^2 \sin^2\theta_1}, \quad \text{and } \hat{\Delta} := \mathrm{D}_{rr} + \hat{\mathrm{D}}_{\theta\theta} + \hat{\mathrm{D}}_{\phi\phi},$$

where $\theta_1 = \dfrac{\pi}{4} - \varepsilon$. Then it is easy to check that $\mathrm{D}_{rr}, \hat{\mathrm{D}}_{\theta\theta}$ and $\hat{\mathrm{D}}_{\phi\phi}$, supplied with zero Dirichlet boundary conditions, commute. Moreover,

$$-\left(\hat{\mathrm{D}}_{\theta\theta}T, T\right)_\omega \geq 0, \quad -\left(\hat{\mathrm{D}}_{\phi\phi}T, T\right)_\omega \geq 0 \tag{2.3}$$

and

$$-\left(\left[\hat{\mathrm{D}}_{\theta\theta} - \mathrm{D}_{\theta\theta}\right]T, T\right)_\omega \geq 0 \text{ and } -\left(\left[\hat{\mathrm{D}}_{\phi\phi} - \mathrm{D}_{\phi\phi}\right]T, T\right)_\omega \geq 0. \tag{2.4}$$

These inequalities immediately yield that:

$$\left(-\hat{\Delta}T, T\right)_\omega \geq \left(-\Delta T, T\right)_\omega. \tag{2.5}$$

In order to obtain an unconditionally stable second-order scheme, we start from the second-order Adams-Bashforth method:

$$\frac{\delta T^{n+1}}{\tau} = \Delta T^{*,n+1/2},$$

and stabilize it by multiplying the time difference in the left hand side by $\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{rr}\right]\left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\theta\theta}\right]\left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\phi\phi}\right]$. Since this perturbation is only first-order consistent with the identity operator, we subtract from the right hand side the first-order perturbation term, taken at the previous time level. The resulting splitting scheme reads:

$$\left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{rr}\right]\left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\theta\theta}\right]\left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\phi\phi}\right]\frac{\delta T^{n+1}}{\tau} = \Delta T^{*,n+1/2} - \frac{1}{2}\hat{\Delta}\delta T^{n}. \qquad (2.6)$$

Note that, assuming enough regularity of the exact solution in space and time, this is a second-order perturbation of the second-order explicit Adams-Bashforth scheme (2.3.2), the perturbation being given by:

$$\frac{\tau^2}{2}\hat{\Delta}\frac{\delta^2 T^{n+1}}{\tau^2} + \left[\frac{\tau^2}{4}(\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta} + \mathrm{D}_{rr}\hat{\mathrm{D}}_{\phi\phi} + \hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}) - \frac{\tau^3}{8}\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\right]\frac{\delta T^{n+1}}{\tau}.$$

We have the following stability result for the scheme (2.6).

Theorem 2.3.1 *Assuming enough regularity of the exact solution T of the semi-discrete scheme* (2.6)*, it is unconditionally stable; more precisely, it sat-*

isfies the following estimate:

$$\begin{aligned}&\tau \sum_{n=1}^{N-1} \frac{\|T^{n+1}-T^{n}\|_{\omega}^{2}}{\tau^{2}} + \frac{1}{2}\|\nabla T^{N}\|_{\omega}^{2} + \\ &\frac{1}{4}\left(\|\partial_{\theta}\left(T^{N}-T^{N-1}\right)\|_{\omega_1}^{2} + \|\partial_{\phi}\left(T^{N}-T^{N-1}\right)\|_{\omega_2}^{2}\right) \\ &\leq \frac{1}{2}\|\nabla T^{1}\|_{\omega}^{2} + \frac{1}{4}\left(\|\partial_{\theta}\left(T^{1}-T^{0}\right)\|_{\omega_1}^{2} + \|\partial_{\phi}\left(T^{1}-T^{0}\right)\|_{\omega_2}^{2}\right),\end{aligned} \tag{2.7}$$

where $\omega_1 = \left(1 - \frac{r^2}{R_1^2}\right)\sin\theta \geq 0$ *and* $\omega_2 = \left(\frac{r^2}{R_1^2} - 1\right)\frac{\sin\theta}{\sin^2\theta_1} \geq 0$.

Proof 2.3.2 *Expanding the left hand side of* (2.6) *we get:*

$$\begin{aligned}&\left[\mathrm{I} - \frac{\tau}{2}\hat{\Delta} + \frac{\tau^2}{4}\left(\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta} + \mathrm{D}_{rr}\hat{\mathrm{D}}_{\phi\phi} + \hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\right) - \frac{\tau^3}{8}\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\right]\frac{\delta T^{n+1}}{\tau} \\ &\qquad = \Delta T^{*,n+1/2} - \frac{1}{2}\hat{\Delta}\delta T^{n}.\end{aligned} \tag{2.8}$$

Rearranging all the Δ *and* $\hat{\Delta}$ *terms, we obtain*

$$\begin{aligned}&\frac{\delta T^{n+1}}{\tau} - \frac{1}{2}\left[\Delta - \hat{\Delta}\right]\left(T^{n+1} - 2T^{n} + T^{n-1}\right) - \Delta T^{n+1/2} \\ &+ \left[\frac{\tau^2}{4}\left(\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta} + \mathrm{D}_{rr}\hat{\mathrm{D}}_{\phi\phi} + \hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\right) - \frac{\tau^3}{8}\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\right]\frac{\delta T^{n+1}}{\tau} = 0.\end{aligned} \tag{2.9}$$

Next we multiply (2.9) *by* $v = \delta T^{n+1}$ *and integrate by parts. Then the second* *term gives*

$$
\begin{aligned}
&-\frac{1}{2}\left(\left[\Delta - \hat{\Delta}\right]\left(T^{n+1} - 2T^{n} + T^{n-1}\right), T^{n+1} - T^{n}\right)_{\omega} \\
&= \frac{1}{4}\left[\|\partial_{\theta}\left(T^{n+1} - T^{n}\right)\|_{\omega_1}^{2} - \|\partial_{\theta}\left(T^{n} - T^{n-1}\right)\|_{\omega_1}^{2}\right] \\
&+ \left[\|\partial_{\theta}\left(T^{n+1} - 2T^{n} + T^{n-1}\right)\|_{\omega_1}^{2}\right] \\
&+ \frac{1}{4}\left[\|\partial_{\phi}\left(T^{n+1} - T^{n}\right)\|_{\omega_2}^{2} - \|\partial_{\phi}\left(T^{n} - T^{n-1}\right)\|_{\omega_2}^{2}\right] \\
&+ \left[\|\partial_{\phi}\left(T^{n+1} - 2T^{n} + T^{n-1}\right)\|_{\omega_2}^{2}\right].
\end{aligned}
\tag{2.10}
$$

The third term is

$$
-\left(\Delta T^{n+1/2}, T^{n+1} - T^{n}\right)_{\omega} = \frac{1}{2}\left(\|\nabla T^{n+1}\|_{\omega}^{2} - \|\nabla T^{n}\|_{\omega}^{2}\right). \tag{2.11}
$$

The remaining terms are all dissipative:

$$
\left(\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta}\delta T^{n+1}, \delta T^{n+1}\right)_{\omega}^{2} = \int\limits_{\Omega} \frac{r^{2}\sin\theta}{R_1^{2}}|\partial_{r\theta}\delta T^{n+1}|^{2}, \tag{2.12}
$$

$$
\left(\mathrm{D}_{rr}\hat{\mathrm{D}}_{\phi\phi}\delta T^{n+1}, \delta T^{n+1}\right)_{\omega} = \int\limits_{\Omega} \frac{r^{2}\sin\theta}{R_1^{2}\sin^{2}\theta_1}|\partial_{r\phi}\delta T^{n+1}|^{2}, \tag{2.13}
$$

$$
\left(\hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\delta T^{n+1}, \delta T^{n+1}\right)_{\omega} = \int\limits_{\Omega} \frac{r^{2}\sin\theta}{R_1^{4}\sin^{2}\theta_1}|\partial_{\theta\phi}\delta T^{n+1}|^{2}, \tag{2.14}
$$

and

$$-\left(\mathrm{D}_{rr}\hat{\mathrm{D}}_{\theta\theta}\hat{\mathrm{D}}_{\phi\phi}\delta T^{n+1}, \delta T^{n+1}\right)_{\omega} = \int\limits_{\Omega} \frac{r^2 \sin\theta}{R_1^4 \sin^2\theta_1} |\partial_{r\theta\phi}\delta T^{n+1}|^2. \tag{2.15}$$

Substituting (2.10)-(2.15) *into* (2.9), *and summing for* $n = 1, \ldots, N-1$ *completes the proof.*

The factorized scheme for the advection-diffusion equation (2.2) is obtained in a similar fashion and takes the following form:

$$\begin{aligned}\left[\mathrm{I} - \frac{\tau}{2}\left(\mathrm{D}_{rr} - u_r^{n+1/2}\partial_r\right)\right]\left[\mathrm{I} - \frac{\tau}{2}\left(\hat{\mathrm{D}}_{\theta\theta} - u_\theta^{n+1/2}\frac{\partial_\theta}{r}\right)\right]& \\ \left[\mathrm{I} - \frac{\tau}{2}\left(\hat{\mathrm{D}}_{\phi\phi} - u_\phi^{n+1/2}\frac{\partial_\phi}{r\sin\theta}\right)\right]\frac{\delta T^{n+1}}{\tau}& \\ = \Delta T^{*,n+1/2} - \frac{1}{2}\hat{\Delta}\delta T^n + \mathbf{u}^{n+1/2}\cdot\nabla T^n.&\end{aligned} \tag{2.16}$$

Direction splitting discretization of the Navier-Stokes system.

Now we present the direction splitting scheme for the Navier-Stokes equations (2.1). Our numerical scheme is based on the AC regularization:

$$\begin{aligned}\partial_t \mathbf{u}_1 + (\mathbf{u}_1\cdot\nabla)\mathbf{u}_1 + \nabla p_1 - \frac{1}{\mathrm{Re}}\Delta\mathbf{u}_1 &= \mathbf{0} \\ \chi\tau\partial_t p_1 + \nabla\cdot\mathbf{u}_1 &= 0,\end{aligned} \tag{2.17}$$

where $\chi = \mathcal{O}(1)$ is an artificial compressibility regularization parameter, and Re is the Reynolds number. It is well-known that the resulting approximation $(\mathbf{u}_1, p_1)$ is first-order accurate in time (see [108]). A second-order scheme can be constructed using the bootstrapping approach of [44, 41], which requires additionally to solve the system:

$$\begin{aligned} \partial_t \mathbf{u}_2 + (\mathbf{u}_2 \cdot \nabla)\mathbf{u}_2 + \nabla p_2 - \frac{1}{\mathrm{Re}}\Delta \mathbf{u}_2 &= \mathbf{0} \\ \chi\tau\partial_t \left(p_2 - p_1\right) + \nabla \cdot \mathbf{u}_2 &= 0, \end{aligned} \tag{2.18}$$

p_1 being given by (2.17). In the following, for the sake of brevity, we will only discuss the direction splitting implementation of the first-order approximation (2.17). The higher order correction for $\mathbf{u}_2, p_2$ is solved identically. First, consider the standard semi-implicit Crank-Nicholson approximation of the system for $(\mathbf{u}_1, p_1)$:

$$\begin{aligned} \frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\tau} + \mathbf{u}_2^{*,n+1/2} \cdot \nabla \mathbf{u}_1^{n+1/2} + \nabla p_1^{n+1/2} - \frac{1}{\mathrm{Re}}\Delta \mathbf{u}_1^{n+1/2} &= \mathbf{0} \\ \chi\left(p_1^{n+1} - p_1^n\right) + \nabla \cdot \mathbf{u}_1^{n+1/2} &= 0 \end{aligned}$$

Note that we use the second-order velocity $\mathbf{u}_2$ as advecting velocity, which allows us to assemble a single linear system for both systems. We can rewrite the momentum equation by eliminating p_1^{n+1} from the first equation:

$$\begin{aligned} \frac{\mathbf{u}_1^{n+1} - \mathbf{u}_1^n}{\tau} + \mathbf{u}_2^{*,n+1/2} \cdot \nabla \mathbf{u}_1^{n+1/2} + \nabla p_1^n - \frac{1}{\mathrm{Re}}\Delta \mathbf{u}_1^{n+1/2} - \frac{1}{2\chi}\nabla\nabla\cdot\mathbf{u}_1^{n+1/2} &= \mathbf{0} \\ p_1^{n+1} = p_1^n - \frac{1}{\chi}\nabla \cdot \mathbf{u}_1^{n+1/2}&. \end{aligned}$$

In order to produce a factorized scheme for each velocity component, the $\nabla\nabla\cdot$ operator must be also split somehow, and we use the Gauss-Seidel type splitting of the $\nabla\nabla\cdot$ operator, which was originally proposed in [41] in the

Cartesian case:

$$\nabla\nabla\cdot\mathbf{u}^{n+1/2} \simeq \begin{pmatrix} \partial_r\left(\frac{\partial_r\left(r^2u_r^{n+1/2}\right)}{r^2}+\frac{\partial_\theta\left(\sin\theta u_\theta^{*,n+1/2}\right)}{r\sin\theta}+\frac{\partial_\phi u_\phi^{*,n+1/2}}{r\sin\theta}\right) \\ \\ \frac{\partial_\theta}{r}\left(\frac{\partial_r\left(r^2u_r^{n+1/2}\right)}{r^2}+\frac{\partial_\theta\left(\sin\theta u_\theta^{n+1/2}\right)}{r\sin\theta}+\frac{\partial_\phi u_\phi^{*,n+1/2}}{r\sin\theta}\right) \\ \\ \frac{\partial_\phi}{r\sin\theta}\left(\frac{\partial_r\left(r^2u_r^{n+1/2}\right)}{r^2}+\frac{\partial_\theta\left(\sin\theta u_\theta^{n+1/2}\right)}{r\sin\theta}+\frac{\partial_\phi u_\phi^{n+1/2}}{r\sin\theta}\right) \end{pmatrix} := \begin{pmatrix} D_{11}+D_{12}+D_{13} \\ D_{21}+D_{22}+D_{23} \\ D_{31}+D_{32}+D_{33} \end{pmatrix}\mathbf{u}^{n+1/2}$$

Equation for the r-component of the velocity. Using the mass conservation equation $\nabla\cdot\mathbf{u}=0$, it is possible to write the first component of the system as follows:

$$\partial_t u_r+\mathbf{u}\cdot\nabla u_r-\frac{\Delta u_r}{\mathrm{Re}}+\partial_r p+\frac{1}{\mathrm{Re}}\frac{2u_r}{r^2}-\frac{1}{\mathrm{Re}}\frac{2}{r^3}\partial_r\left(u_r r^2\right)-\frac{u_\theta^2+u_\phi^2}{r}=0,$$

where $\mathbf{u}\cdot\nabla v=u_r\partial_r v+u_\theta\frac{\partial_\theta v}{r}+u_\phi\frac{\partial_\phi v}{r\sin\theta}$ is the advection operator. Let $\mathrm{L}_{rr},\mathrm{L}_{r\theta}$ and $\mathrm{L}_{r\phi}$ be the differential operators that act in each space direction:

$$\mathrm{L}_{rr}u=\frac{1}{\mathrm{Re}}\left(\mathrm{D}_{rr}u-\frac{2u}{r^2}+\frac{2\partial_r\left(r^2u\right)}{r^3}\right)+D_{11}u-u_{2,r}^{*,n+1/2}\cdot\partial_r u_r,\mathrm{L}_{r\theta}u=\left(\frac{\hat{\mathrm{D}}_{\theta\theta}}{\mathrm{Re}}-u_{2,\theta}^{*,n+1/2}\cdot\frac{\partial_\theta}{r}\right)u,$$

$$\mathrm{L}_{r\phi}u=\left(\frac{\hat{\mathrm{D}}_{\phi\phi}}{\mathrm{Re}}-u_{2,\phi}^{*,n+1/2}\cdot\frac{\partial_\phi}{r\sin\theta}\right)u \text{ and } L_r=\mathrm{L}_{rr}+\mathrm{L}_{r\theta}+\mathrm{L}_{r\phi}$$

The factorized scheme for the r-component takes the following form:

$$\begin{aligned}\left[\mathrm{I}-\frac{\tau}{2}\mathrm{L}_{r\theta}\right]\left[\mathrm{I}-\frac{\tau}{2}\mathrm{L}_{r\phi}\right]\left[\mathrm{I}-\frac{\tau}{2}\mathrm{L}_{rr}\right]\frac{u_{1,r}^{n+1}-u_{1,r}^{n}}{\tau} &= L_r u_{1,r}^{*,n+1/2}+ \\ \hat{\Delta}u_{1,r}^{n-1/2}-\partial_r p_1^n+\frac{D_{12}u_{1,\theta}^{*,n+1/2}+D_{13}u_{1,\phi}^{*,n+1/2}}{2\chi} & \\ +\frac{\left(u_\theta^{*,n+1/2}\right)^2+\left(u_\phi^{*,n+1/2}\right)^2}{r}. &\end{aligned} \tag{2.19}$$

Equation for the θ–component of the velocity. Again using $\nabla\cdot\mathbf{u}=0$, the θ-component of the momentum equation can be expressed as:

$$\begin{aligned}\partial_t u_\theta+\mathbf{u}\cdot\nabla u_\theta-\frac{\Delta u_\theta}{\mathrm{Re}}+\frac{\partial_\theta p}{r}+\frac{1}{\mathrm{Re}}\frac{u_\theta}{r^2\sin^2\theta}-\frac{2\cos\theta}{\mathrm{Re}}\frac{\partial_\theta\left(u_\theta\sin\theta\right)}{r^2\sin^2\theta}& \\ -\frac{2}{\mathrm{Re}}\frac{\partial_\theta u_r}{r^2}-\frac{2\cos\theta}{\mathrm{Re}}\frac{\partial_r\left(u_r r^2\right)}{r^3\sin\theta}+\frac{u_r u_\theta-u_\phi^2\cot\theta}{r}&=0.\end{aligned}$$

Let $\mathrm{L}_{\theta r},\mathrm{L}_{\theta\theta}$ and $\mathrm{L}_{\theta\phi}$ be defined as follows:

$$\mathrm{L}_{\theta r}u=\left(\frac{\mathrm{D}_{rr}}{\mathrm{Re}}-u_{2,r}^{*,n+1/2}\cdot\partial_r\right)u,\mathrm{L}_{\theta\phi}u=\left(\frac{\hat{\mathrm{D}}_{\phi\phi}}{\mathrm{Re}}-u_{2,\phi}^{*,n+1/2}\cdot\frac{\partial_\phi}{r\sin\theta}\right)u,$$

$$\mathrm{L}_{\theta\theta}u=\frac{1}{\mathrm{Re}}\left(\hat{\mathrm{D}}_{\theta\theta}u-\frac{u}{r^2\sin^2\theta}+\frac{2\cos\theta}{\sin\theta}\partial_\theta\left(u\sin\theta\right)\right)+\frac{u\cdot u_{2,\phi}^{*,n+1/2}\cot\theta}{r}+u_{2,\theta}^{*,n+1/2}\cdot\frac{\partial_\theta u}{r}+\frac{D_{22}u}{2\chi},$$

$$\text{and } L_\theta=\mathrm{L}_{\theta r}+\mathrm{L}_{\theta\theta}+\mathrm{L}_{\theta\phi}$$

The factorized scheme for the θ-component takes the following form:

$$\begin{aligned}\left[\mathrm{I}-\frac{\tau}{2}\mathrm{L}_{\theta\phi}\right]\left[\mathrm{I}-\frac{\tau}{2}\mathrm{L}_{\theta r}\right]\left[\mathrm{I}-\frac{\tau}{2}\mathrm{L}_{\theta\theta}\right]\frac{u_{1,\theta}^{n+1}-u_{1,\theta}^{n}}{\tau} &= L_\theta u_{1,\theta}^{*,n+1/2} \\ -\frac{\partial_\theta p_1^n}{r}+\frac{D_{21}u_{1,r}^{n+1/2}+D_{23}u_{1,\phi}^{*,n+1/2}}{2\chi}-\frac{u_r^{*,n+1/2}\cdot u_\phi^{*,n+1/2}}{r} & \\ +\frac{1}{\mathrm{Re}}\left(\frac{2}{r^2}\partial_\theta u_{1,r}^{n+1/2}+\frac{2\cos\theta}{r^3\sin\theta}\partial_r\left(u_{1,r}^{n+1/2}r^2\right)\right)+\hat{\Delta}u_{1,\theta}^{n-1/2}. &\end{aligned} \tag{2.20}$$

Equation for the ϕ–component of the velocity. The ϕ-component of the momentum equation is given by:

$$\begin{aligned}
&\partial_t u_\phi + \mathbf{u}\cdot\nabla u_\phi + \frac{u_r u_\phi + u_\theta u_\phi \cot\theta}{r} - \frac{\Delta u_\phi}{\mathrm{Re}} + \frac{\partial_\phi p}{r\sin\theta} \\
&+\frac{1}{\mathrm{Re}}\left(\frac{u_\phi}{r^2\sin^2\theta} - \frac{2\cos\theta}{r^2\sin^2\theta}\partial_\phi u_\theta - \frac{2}{r^2\sin\theta}\partial_\phi u_r\right) = 0
\end{aligned}$$

Let $\mathrm{L}_{\phi r}, \mathrm{L}_{\phi\theta}$ and $\mathrm{L}_{\phi\phi}$ be defined as follows:

$$\mathrm{L}_{\phi r} u = \left(\frac{\mathrm{D}_{rr}}{\mathrm{Re}} - u_{2,r}^{*,n+1/2}\cdot\partial_r\right)u \text{ and } \mathrm{L}_{\phi\theta} u = \left(\frac{\hat{\mathrm{D}}_{\theta\theta}}{\mathrm{Re}} - u_{2,\theta}^{*,n+1/2}\cdot\frac{\partial_\theta}{r}\right)u$$

$$\mathrm{L}_{\phi\phi} u = \frac{1}{\mathrm{Re}}\left(\hat{\mathrm{D}}_{\phi\phi} - \frac{1}{r^2\sin^2\theta}\right)u - \frac{u_\phi^{*,n+1/2}\cdot u}{r\sin\theta} - \frac{u_{2,r}^{*,n+1/2} + u_{2,\theta}^{*,n+1/2}\cot\theta}{r}u$$

and

$$L_\phi = \mathrm{L}_{\phi r} + \mathrm{L}_{\phi\theta} + \mathrm{L}_{\phi\phi}$$

The factorized scheme for the ϕ-component is then:

$$\begin{aligned}
&\left[\mathrm{I} - \frac{\tau}{2}\mathrm{L}_{\phi r}\right]\left[\mathrm{I} - \frac{\tau}{2}\mathrm{L}_{\phi\theta}\right]\left[\mathrm{I} - \frac{\tau}{2}\mathrm{L}_{\phi\phi}\right]\frac{u_{1,\phi}^{n+1} - u_{1,\phi}^{n}}{\tau} = L_\phi u_{1,\phi}^{*,n+1/2} + \hat{\Delta} u_{1,\phi}^{n-1/2} \\
&\quad -\frac{\partial_\phi p_1^n}{r\sin\theta} + \frac{1}{\mathrm{Re}}\left(\frac{2}{r^2\sin\theta}\partial_\phi u_{1,r}^{n+1/2} + \frac{2\cos\theta}{r^2\sin^2\theta}\partial_\phi u_{1,\theta}^{n+1/2}\right) \\
&\qquad + D_{31}u_{1,r}^{n+1/2} + D_{32}u_{1,\theta}^{n+1/2}.
\end{aligned} \tag{2.21}$$

Pressure update.

$$p_1^{n+1} = p_1^n - \frac{1}{\chi}\nabla\cdot\mathbf{u}_1^{n+1/2}. \tag{2.22}$$

2.3.3 Implementation and parallelization.

The equations (2.16), (2.19)-(2.21) are solved as a sequence of 1D equations in each space direction. For example, solving (2.16) consists of the following steps:

$$
\begin{aligned}
&\frac{\xi^{n+1}}{\tau} := \frac{1}{2}\Delta T^{*,n+1/2} - \frac{1}{2}\hat{\Delta}\delta T^n + \mathbf{u}^{n+1/2}\cdot\nabla T^n \\
&\frac{\eta^{n+1}}{\tau} := \left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\theta\theta}\right]\left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\phi\phi}\right]\frac{T^{n+1}-T^n}{\tau} \Rightarrow \left[\mathrm{I} - \frac{\tau}{2}\mathrm{D}_{rr}\right]\eta^{n+1} = \xi^{n+1} \\
&\frac{\zeta^{n+1}}{\tau} := \left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\phi\phi}\right]\frac{T^{n+1}-T^n}{\tau} \Rightarrow \left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\theta\theta}\right]\zeta^{n+1} = \eta^{n+1} \\
&\left[\mathrm{I} - \frac{\tau}{2}\hat{\mathrm{D}}_{\phi\phi}\right]\left(T^{n+1}-T^n\right) = \zeta^{n+1} \Rightarrow T^{n+1} = \left(T^{n+1}-T^n\right) + T^n.
\end{aligned}
$$

Similar strategy is applied for the Navier-Stokes approximation. Each 1D system is spatially approximated using second-order centered finite differences on a non-uniform grid. To ensure the inf-sup stability, the unknowns are approximated on a MAC grid, where the velocity components are stored at the face centers of the cells, while the scalar variables are stored at the cell centers (see [52]).

To solve the system on each domain in parallel we use the approach developed in [43], where we first perform Cartesian domain decomposition of both computational grids using MPI and then solve the resulting set of tridiagonal linear systems using domain-decomposition-induced Schur complement technique. Note, that the Schur complement can be computed explicitly (see [43] for details) and so the system in each direction can be solved directly by the Thomas algorithm, avoiding the need for iterations on each of the two subdomains. Then, to obtain the approximation on the entire spherical shell, we iterate between the Yin and Yang grids using either additive or multiplicative overlapping Schwarz methods. The solution on each grid is computed using only boundary data that is interpolated from the currently available solution on the other grid, using Lagrange interpolation.

In the additive Schwarz implementation, we use an even total number of CPUs. Then we split the global communicator into two equal parts, and assign to each grid one of the communicators. In the multiplicative Schwarz implementation, we use the global communicator to solve the problem on each grid sequentially.

The overall solution procedure in the case of the multiplicative Schwarz iteration can be summarized as follows:

Repeat until convergence:

For $i = 1, 2$

1) Obtain interpolated boundary values T_{bd} for $\partial\Omega_i$ from Ω_{3-i}.
2) Solve the temperature equation in Ω_i with using extrapolated velocity values $\mathbf{u}_2^{*,n+1/2}$.
3) Obtain interpolated boundary values $\mathbf{u}_{bd}$ for $\partial\Omega_i$ from Ω_{3-i}.
4) If $\left| \int\limits_{\partial\Omega_i} \mathbf{u}_{bd} \cdot \mathbf{n} \right| \geq \text{tol}$, then minimize the functional ($\varepsilon \ll 1$):

$$\mathrm{J}\left(\mathbf{v}\right) := \frac{1}{2}\left|\mathbf{v} - \mathbf{u}_{bd}\right|^2_{\ell^2} + \frac{1}{2\varepsilon|\partial\Omega_i|^2} \left| \int\limits_{\partial\Omega_i \cap \{\theta,\phi \text{ bdry }\}} \mathbf{v}\cdot\mathbf{n} + \int\limits_{\partial\Omega_i \cap \{r \text{ bdry }\}} \mathbf{u}_{bd}\cdot\mathbf{n} \right|^2,$$

using the Conjugate Gradient Algorithm until $\mathrm{J}\left(\cdot\right) \leq \text{tol}$.

5) Update $\mathbf{u}_{bd} := \mathbf{v}$ and solve the momentum equation in Ω_i with the interpolated Dirichlet boundary conditions in θ, ϕ directions and with the original boundary conditions in the r direction.

6) Compute the pressure in Ω_i using the second equation in (2.17).

7) Interpolate the pressure values at the boundary of $\partial\Omega_i$ using the available pressure on Ω_{3-i}.

End for.

Step 4 is meant to ensure that there is no spurious mass flux generated through the internal (artificial) boundaries due to the interpolation. It is optional, and as our numerical experience shows, it rarely changes the results significantly. Therefore, it is skipped while producing the numerical results presented in the next section. Skipping Step 7, however, can seriously reduce the rate of convergence of the Schwarz iteration, as observed in the numerical simulations. The AC method for the Navier-Stokes equations does not require boundary conditions on the pressure. Nevertheless, the exchange of the pressure values does influence the pressure gradient that appears in (2.19)-(2.21), and thus it seems to influence significantly the convergence of the overall iteration. This effect is not well understood and while some other authors (see for example [116]) also interpolate the pressure values near the internal boundaries, others (e.g. [84]) interpolate only the velocity.

Another interesting feature of the domain decomposition iterations described above is that it allows to use the previously computed iterates in order to reduce the splitting error of the direction splitting approximation. For example, if the factorized form of the direction-split approximation for a given quantity ψ is given by:

$$(I - L_{\psi,r})(I - L_{\psi,\theta})(I - L_{\psi,\phi})(\psi^{n,k} - \psi^{n-1}) = G$$

where the superscript n denotes the time level of the solution and k denotes the domain decomposition iteration level, then the splitting error can be reduced by using the modified equation:

$$\begin{gathered}(I - L_{\psi,r})(I - L_{\psi,\theta})(I - L_{\psi,\phi})(\psi^{n,k} - \psi^{n,k-1}) = \\ G + (\psi^{n,k-1} - \psi^{n-1}) - L_{\psi}(\psi^{n,k-1} - \psi^{n-1}),\end{gathered} \tag{2.23}$$

where $L_{\psi} = L_{\psi,r} + L_{\psi,\theta} + L_{\psi,\phi}$. Indeed, in (2.23), the splitting error term $(L_{\psi,r}L_{\psi,\theta} + L_{\psi,r}L_{\psi,\phi} + L_{\psi,\theta}L_{\psi,\phi} - L_{\psi,r}L_{\psi,\theta}L_{\psi,\phi})(\psi^{n,k} - \psi^{n-1})$ at iteration level k has been reduced by the same term at the previous iteration level $(L_{\psi,r}L_{\psi,\theta} + L_{\psi,r}L_{\psi,\phi} + L_{\psi,\theta}L_{\psi,\phi} - L_{\psi,r}L_{\psi,\theta}L_{\psi,\phi})(\psi^{n,k-1} - \psi^{n-1})$. If this error reduction is employed, then the iteration becomes a block-preconditioned overlapping domain decomposition iteration, the preconditioner being the factorized operator $(I - L_{\psi,r})(I - L_{\psi,\theta})(I - L_{\psi,\phi})$. We must also remark here that the Schwartz iterations need to converge to an accuracy of the order of τ^2 for the solution of equation (2.17) and τ^3 for the solution of equation (2.18), to preserve the second-order temporal accuracy of the overall algorithm.

2.4 Numerical tests.

2.4.1 Time and space convergence.

We verify the convergence rates in space and time using the following manufactured solution, given in a Cartesian form:

$$\mathbf{u} = \cos(t)\left(2x^2yz, -xy^2z, -xyz^2\right)^{\mathrm{T}}, p = \cos(t)xyz, T = 2\cos(t)x^2yz. \quad (2.1)$$

The parameters used in this test are $R_1 = 1, R_2 = 2, Ra = 1, Pr = 1$, and the grids used in the tests are uniform in each direction. The convergence of the approximation is tested using both, the additive and multiplicative versions of the scheme. The grid used for the time convergence tests consists of $20 \times 92 \times 192$ MAC cells on each of the two subdomains. The solution error is computed at the final time $T_f = 10$. For the space convergence tests, the time step is chosen small enough to not influence the overall error, $\tau = 0.0001$, and the final time is $T_f = 1$. The grid diameter is computed as the maximum diameter of the MAC cells in Cartesian coordinates. In both cases, the domain decomposition iterations are converged so that the l^2 norm of the difference between two subsequent iterates, for any of the computed quantities, is less than 10^{-6} (l^2 norm denotes the standard mid-point approximation to the L^2 norm). Also, the splitting error reduction, as outlined by equation (2.23), is employed at each iteration.

The graphs of the l^2 norm of the errors in both cases are presented in Figure 2.3. They demonstrate the second-order accuracy of the scheme in space and time.

Next, we verify the accuracy of the proposed algorithm on a physically relevant analytic solution of the Navier-Stokes equations in a spherical setting, due to Landau (see [72] and [76] for a recent review). The source term of the equations is equal to zero in this case, and the solution is steady and axisymmetric. In all cases presented in figure 2.4 the multiplicative Schwarz version of the algorithm is used with its convergence tolerance being set to 10^{-6}, the time step is equal to 10^{-3}, and $R_1 = 1, R_2 = 2$. We first present in the top graph of figure 2.4 the l^2 error for the velocity and pressure as a function of the grid diameter, at $Re = 1$ and the overlap is $\epsilon = 0.1$ The scheme clearly exhibits the expected second-order convergence rate in space. In the second graph we demonstrate the influence of the overlap size on the error at $Re = 1$, the grid size in the r, ϕ, θ directions being $2.7778 \times 10^{-2}, 1.7027 \times 10^{-2}, 3.6121 \times 10^{-2}$ correspondingly. The effect of the overlap on the error is insignificant, however, it seriously impacts the stability of the algorithm i.e. the increase of the overlap improves the stability, particularly at large Reynolds numbers. Finally, the bottom graph demonstrates the effect of the Reynolds number on the error. Again, the overlap is 0.1 and the grid sizes are equal to $2.7778 \times 10^{-2}, 1.7027 \times 10^{-2}, 3.6121 \times 10^{-2}$. We should note that the exact solution for the velocity scales like Re^{-1} and therefore the errors in the graph are multiplied by the corresponding Reynolds number. The oscillations in the error decrease slower with the increase of the Reynolds number. These oscillations are due to the artificial compressibility algorithm, since the initial data for the pressure corresponds to a divergence-free velocity, while the pressure evolution is determined by a perturbed continuity equation (see [94] and [23] for a detailed discussion on this issue).

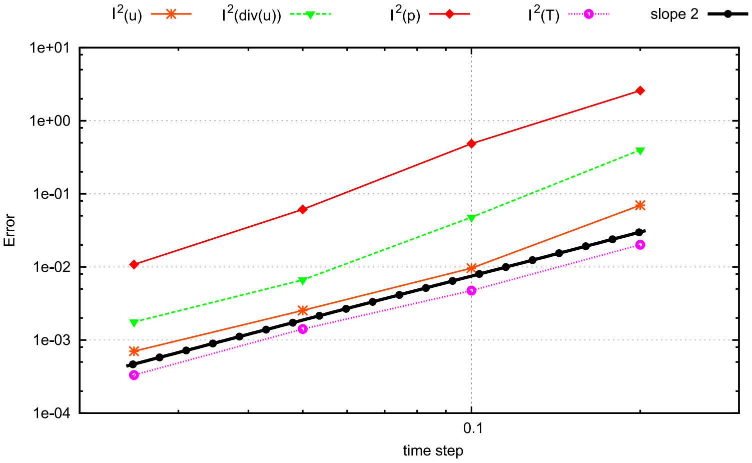

Figure **2.2.a:** Temporal errors at $T_f = 10$.

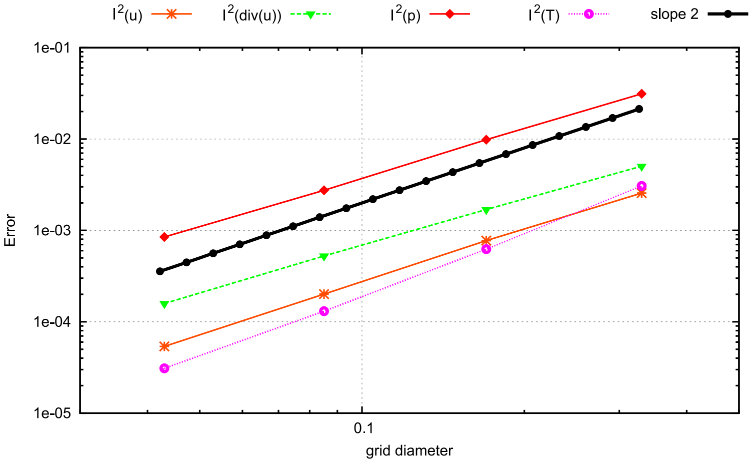

Figure **2.2.b:** Spatial errors at $T_f = 1$.

Figure 2.2: Log-log plot of the errors; multiplicative Schwartz iterations. $R_1 = 1, R_2 = 2, Ra = 1, Pr = 1$.

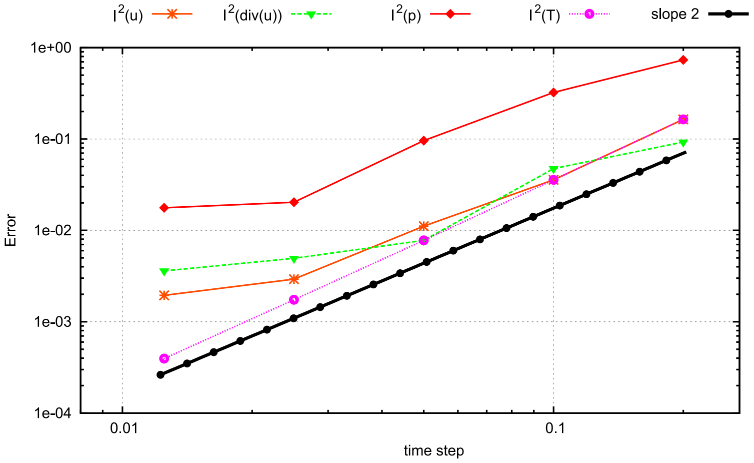

Figure **2.3.a:** Temporal errors at $T_f = 10$.

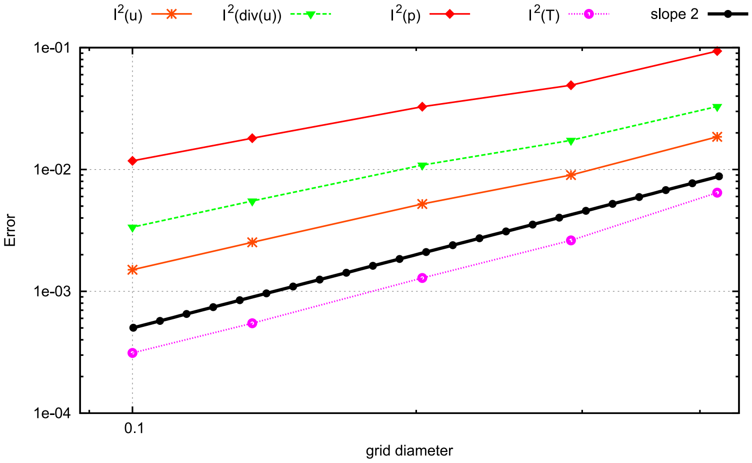

Figure **2.3.b:** Spatial errors at $T_f = 1$.

Figure 2.3: Log-log plot of the errors; additive Schwartz approach. $R_1 = 1, R_2 = 2, Ra = 1, Pr = 1$.

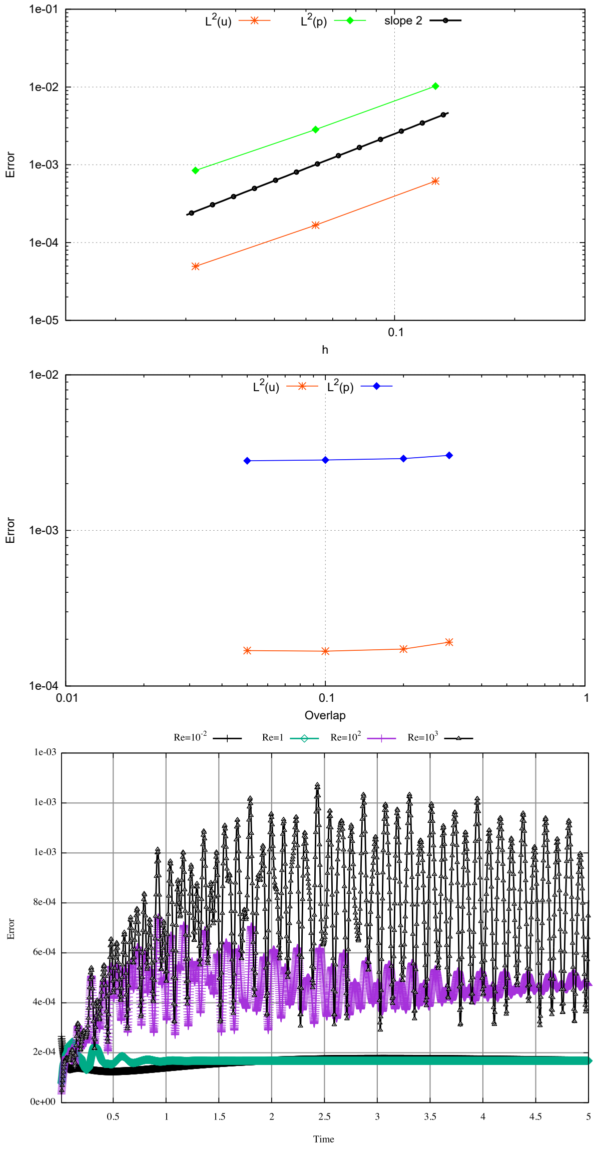

Figure 2.4: l^2 errors. Top: convergence in space, Re=1. Center: effect of the overlap on the error; Re=1. Bottom: effect of the Reynolds number on the velocity error as a function of time.

2.4.2 Weak parallel efficiency.

Next, we test the parallel efficiency of the code based on the scheme introduced in the previous section. Since we are interested in solving large-size problems, we only measure the weak scalability of our code. The problem size is $100 \times 100 \times 100$ grid cells per each of the Yin and Yang grids on each CPU, and the maximum number of CPUs used is 960. Besides, since in the possible applications of this technique (atmospheric boundary layer, Earth's dynamo) the thickness of the spherical shell is much smaller than the diameter of the shell, we use a two-dimensional grid of processors for the grid partitioning. It must be noted though, that making the grid partitioning three dimensional does not change much the parallel efficiency results presented in this section. The scaling efficiency is computed as the ratio of the CPU time on 32 cores divided by the CPU time on $n \geq 32$ cores. The reason for this definition of efficiency is that the particular cluster used in the scaling tests has processors containing 32 cores each, and the efficiency drops significantly between 1 and 32 cores (to about 75%). After this, when the number of cores is a multiple of 32 the efficiency remains very close to the one at 32 cores. One possible explanation of this phenomenon is that in the case when the number of cores is significantly less than 32 cores, they need to share the memory bandwidth and cache with a smaller number of cores, since presumably the rest of the available cores on the given processor are idle (see e.g. [65], p. 152). Again, we are interested in very large computations, and therefore, using a minimum of 32 cores is reasonable.

The scaling results are performed using the Compute Canada (see https://www.computecanada.ca/) Graham cluster of 2.1GHz Intel $E5 - 2683$ v4 CPU cores, 32 cores per node, and each node connected via a 100 Gb/s network. The results were calculated using the wall clock time taken to simulate 10 time steps. We ran two tests, using a fixed number of 1 and 10 domain decomposition iterations, and we present the scaling results in Fig. 2.5. The parallel efficiency is very slightly dependent on the number of domain decomposition iterations and remains above 90% for the number of cores ranging between 32 and 960.

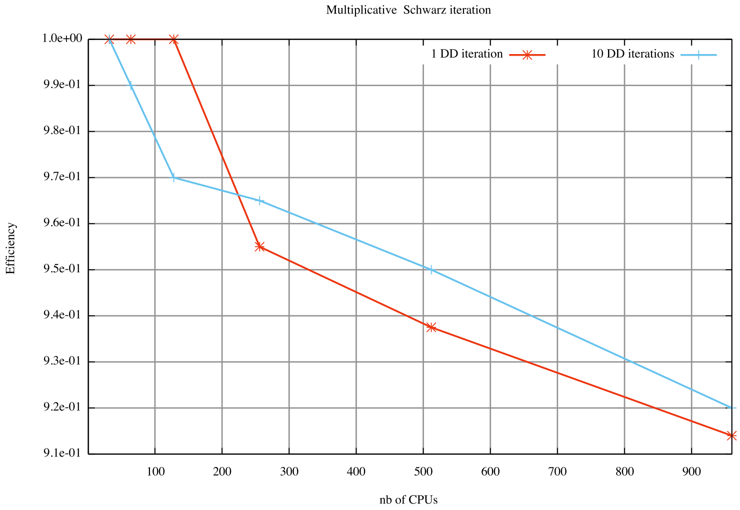

Figure 2.5: Parallel scalability using up to 960 CPU cores

2.5 Conclusion.

This chapter introduced a formally second-order direction splitting method for solving the incompressible Navier-Stokes-Boussinesq system in a spherical shell region. The main novelty of this work is the formulation of splitting strategies for the advection-diffusion and incompressible Navier-Stokes equations in spherical coordinates in combination with an artificial compressibility regularization method for incompressible flows. The equations were solved on the overset Yin-Yang grid, which is based on spherical coordinates. This approach allows one to avoid the singularities at the poles and keep the grid size relatively uniform. The spatial discretization is based on second-order finite differences on the Marker-And-Cell (MAC) stencil to avoid odd-even decoupling and ensure inf-sup stability.

The stability, accuracy and parallel scalability properties of the method were studied using a manufactured solution of the Navier-Stokes-Boussinesq system and the Landau solution of the Navier-Stokes equations on a sphere. The algorithm may be used for simulations of natural convection flows, modeling of mantle convection, and global ocean and atmosphere in processes where the Boussinesq approximation is valid. Furthermore, the techniques developed in this chapter and Chapter 1 lay a foundation for a numerical method for solving compressible Navier-Stokes equations in spherical geometry in low-Mach number regime that is described in Chapter 3.

Chapter 3

An efficient algorithm for weakly compressible flows in spherical geometries

3.1 Introduction.

This chapter relies on the progress made in Chapters 1 and 2 to propose a numerical method for weakly compressible flows in spherical geometries, i.e. for solving the compressible Navier-Stokes equations in spherical coordinates at small and moderate values of the Mach number. The main motivation to design such a scheme comes from the atmospheric science and oceanography, where reliable dynamical cores for global and local ocean-atmosphere circulations are required to decrease uncertainties in numerical weather prediction and climate modeling. Despite recent advances in numerical methods for atmospheric and oceanic flows, there remain to be several important challenges in this field, such as the necessity to take into account nonlinear nature of the governing equations, avoid simplifications of the model that may only be valid in certain asymptotic limits, improve the efficiency and increase the accuracy and resolution of the computations while maintaining stability, consistently couple ocean and atmosphere models together, and many others. Our goal is to address at least some of these problems.

Global ocean circulation models are traditionally based on the primitive hydrostatic equations (see [39]), although some efforts have been made to use a more general incompressible Navier-Stokes equations (NSEs) model (e.g. [82]). However, incompressibility assumption holds only under certain conditions on the motion-induced fluctuations of density and pressure (e.g. see [113] for analysis of Boussinesq approximation for an ideal gas), and compressibility is known to influence certain processes in the ocean, such as tsunami propagation (see [1]). Since there is a rationale that climate models have to adequately resolve mesoscale (see [95]), and with growing computational capabilities perhaps move beyond it, the most general and comprehensive physical models should be used for the ocean circulation at the highest possible resolution to resolve as many physical processes as possible, reducing the need for parametrizations. Besides that, the compressible Navier-Stokes equations are undoubtedly required for the simulation of the atmosphere, and the use of the same set of governing equations for both the atmosphere and the ocean may simplify the coupling of the two models, which is a significant problem nowadays (some simple *ad-hoc* strategies are shown to produce inconsistent results, see e.g. [75], [74]). Solving the compressible Navier-Stokes equations for the ocean is extremely challenging from the numerical point of view and requires a method that is very robust at extremely low Mach numbers. The need for extreme resolution at global scales imposes strict requirements for the parallel performance of the numerical methods in use. Although we do not hope to answers all of the open questions, we present some evidence that the proposed methodology has potential and may help address some of the issues discussed above.

The main idea of the proposed methodology is similar to Chapter 1 - an LBI factored scheme (see [12], [9], [10]) coupled with an iterative splitting error reduction strategy is used to efficiently solve the full system of governing equations implicitly. There are several differences between the algorithm presented in this chapter and the algorithm from Chapter 1. First, the governing equations are written in spherical coordinates, which leads to more involved differential operators and non-trivial direction splitting. The equations are written in non-conservative advection form, which is better suited for nearly incompressible flows than the conservative formulation. Next, nonlinear Picard iterations are used instead of the simple Taylor-based linearization which allows one to solve the full nonlinear system without computing complicated expressions for Jacobians. These iterations are combined with the splitting error reduction. The direction splitting strategy is based on the second-order Douglas-type direction splitting (see [28]) instead of the first-order direction splitting employed in Chapter 1, to increase the accuracy of the scheme. Finally, the staggered Marker-and-Cell (MAC) grid is used instead of the centered non-staggered discretization. This helps to avoid the high-frequency pressure oscillations without the introduction of any artificial dissipation terms. It is possible to do it in this case since we are not interested here in high-Mach number flows with shock waves, and thus do not need to consider Riemann problems between adjacent cells.

The rest of this chapter is organized as follows. We describe the details of the proposed algorithm in Section 3.2. Numerical experiments are described in Section 3.3, and we provide some concluding remarks in Section 3.4.

3.2 Algorithm.

In this chapter we are interested in simulation of flows at extremely low to moderate values of the Mach number, and thus we do not expect any shock waves to be present in the solution. Hence, the non-conservative formulation with pressure p $[Pa]$, velocity $\mathbf{u}$ $[m/s, 1/s, 1/s]$, and temperature T $[K]$ as unknowns may be employed, which is a more natural setting for the incompressible regime. In case of dry, stratified air, it means solving the following system of PDEs (see Appendix B for a detailed derivation of this system based on the conservative equation set for atmospheric modeling from [81]):

$$\begin{aligned}
&\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T + (\gamma-1)T\nabla\cdot\mathbf{u} - \frac{(\gamma-1)T}{p+\pi_\infty}\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla T\right) \\
&\qquad\qquad - \frac{(\gamma-1)T}{p+\pi_\infty}\nabla\mathbf{u} : \hat{\boldsymbol{\sigma}} = 0, \qquad (3.1)\\
&\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \frac{1}{\rho}\nabla p - \frac{1}{\rho}\nabla\cdot\hat{\boldsymbol{\sigma}} + \mathbf{g} + 2(\mathbf{u}\times\boldsymbol{\omega}) = 0, \qquad (3.2)\\
&\frac{\partial p}{\partial t} + \mathbf{u}\cdot\nabla p + \gamma(p+\pi_\infty)\nabla\cdot\mathbf{u} - (\gamma-1)\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla T\right) \qquad (3.3)\\
&\qquad\qquad - (\gamma-1)\nabla\mathbf{u} : \hat{\boldsymbol{\sigma}} = 0.
\end{aligned}$$

where $\boldsymbol{\omega}$ $[1/s, 1/s, 1/s]$ is the rotational velocity of the Earth, $\hat{\boldsymbol{\sigma}}$ is the viscous stress tensor given by

$$\hat{\boldsymbol{\sigma}} = \mu\left[\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) - \frac{2}{3}(\nabla\cdot\mathbf{u})\hat{\mathbf{I}}\right],$$

$\mathbf{g}$ $[m/s^2, 1/s^2, 1/s^2]$ is the sum of the true gravity and the centrifugal force, c_p $[J/(K\cdot kg)]$, c_v $[J/(K\cdot kg)]$, μ $[kg/(s\cdot m)]$, Pr, $\gamma = \dfrac{c_p}{c_V}$, π_∞ $[Pa]$ are constant for each material, and density ρ $[kg/m^3]$ is given by the Stiffened Gas Equation of State ([32]):

$$\rho = \frac{p + \pi_\infty}{c_V(\gamma - 1)T}. \tag{3.4}$$

The system (3.1)-(3.3) is solved in a part of a spherical shell similar to the Yin subdomain introduced in Chapter 2:

$$\Omega := \left\{ (r, \theta, \phi) \in [R_1, R_2] \times \left[\frac{\pi}{4}, \frac{3\pi}{4}\right] \times \left[\frac{\pi}{4}, \frac{7\pi}{4}\right] \right\}.$$

Although it was not done in this study, the domain decomposition techniques developed for Chapter 2 can be employed to extend the domain to the whole spherical shell without any changes in the algorithm.

In order to simplify the notations, we denote the vector of unknowns as $\mathbf{U} = [p, u_r, u_\theta, u_\phi, T]^T$, the gravity vector as $\boldsymbol{Gr} = [0, g, 0, 0, 0]^T$, and combine all the components of the differential operators in the corresponding directions into the $\boldsymbol{D_r}(\mathbf{U})$, $\boldsymbol{D_\theta}(\mathbf{U})$, and $\boldsymbol{D_\phi}(\mathbf{U})$ operators, and all the mixed derivatives, derivates in staggered directions, and other terms not suitable for implicit treatment by the direction splitting approach into the $\boldsymbol{D_M}(\mathbf{U})$ operator (see (C.54),(C.55), (C.56), and (C.57) for definitions of these operators). Then the system (3.1)-(3.3) can be written in a compact form as (see Appendix C for details):

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{D_r}(\mathbf{U})\mathbf{U} + \boldsymbol{D_\theta}(\mathbf{U})\mathbf{U} + \boldsymbol{D_\phi}(\mathbf{U})\mathbf{U} + \boldsymbol{D_M}(\mathbf{U})\mathbf{U} + \boldsymbol{Gr} = 0. \tag{3.5}$$

Using the Crank-Nicolson time-discretization strategy and the Picard iterations (see e.g. [73], Chapter 3), the semi-discrete version of (3.5) reads as:

$$\begin{aligned}\frac{\mathbf{U}_{k+1}^{n+1}-\mathbf{U}^{n}}{\tau}+&\frac{1}{2}\boldsymbol{D}^{n+\frac{1}{2},(k)}\mathbf{U}_{k+1}^{n+1}+\frac{1}{2}\boldsymbol{D}^{n+\frac{1}{2},(k)}\mathbf{U}^{n}+\\ &\frac{1}{2}\boldsymbol{D_M}^{n+\frac{1}{2},(k)}\mathbf{U}_{k}^{n+1}+\frac{1}{2}\boldsymbol{D_M}^{n+\frac{1}{2},(k)}\mathbf{U}^{n}+\boldsymbol{Gr}=0,\end{aligned} \tag{3.6}$$

where

$$\boldsymbol{D}(\mathbf{U})=\boldsymbol{D_r}(\mathbf{U})+\boldsymbol{D_\theta}(\mathbf{U})+\boldsymbol{D_\phi}(\mathbf{U}),$$

$$\boldsymbol{D}^{n+\frac{1}{2},(k)}=\boldsymbol{D}\left(\frac{\mathbf{U}_{k}^{n+1}+\mathbf{U}^{n}}{2}\right),$$

τ is the time-step, subindex n refers to the time level, subindex k refers to the iteration level. From now on we skip the upper subindixes of operators for brevity, assuming that $\boldsymbol{D}=\boldsymbol{D}^{n+\frac{1}{2},(k)}$. Next, similar to Section 2.3.2 a Douglas-type (see [28]) direction splitting scheme can be written for (3.5) in the following factorized form:

$$\begin{aligned}\left(I+\frac{\tau}{2}\boldsymbol{D_r}\right)\left(I+\frac{\tau}{2}\boldsymbol{D_\theta}\right)\left(I+\frac{\tau}{2}\boldsymbol{D_\phi}\right)\left(\mathbf{U}_{k+1}^{n+1}-\mathbf{U}^{n}\right)=&\\ -\tau\boldsymbol{D}\mathbf{U}^{n}-\tau\boldsymbol{Gr}-\frac{\tau}{2}\boldsymbol{D_M}\mathbf{U}_{k}^{n+1}-\frac{\tau}{2}\boldsymbol{D_M}\mathbf{U}^{n}.&\end{aligned} \tag{3.7}$$

Recalling the discussion in Sections 1.3.1 and 1.4.1 on the adverse effects of the splitting error at low values of Mach number, the error can be removed using a strategy similar to the one describe in Section 2.3.3. Indeed, we can incorporate the splitting error reduction into the Picard nonlinear iterations by adding

$$\begin{aligned}\boldsymbol{ER}\left(\mathbf{U}_{k}^{n+1}-\mathbf{U}^{n}\right)=\left(I+\frac{\tau}{2}\boldsymbol{D_r}\right)\left(I+\frac{\tau}{2}\boldsymbol{D_\theta}\right)\left(I+\frac{\tau}{2}\boldsymbol{D_\phi}\right)\left(\mathbf{U}_{k}^{n+1}-\mathbf{U}^{n}\right)-&\\ \left(I+\frac{\tau}{2}\boldsymbol{D}\right)\left(\mathbf{U}_{k}^{n+1}-\mathbf{U}^{n}\right)&\end{aligned}$$

to the right-hand-side of (3.7). After rearranging terms for conveniece and computional efficiency, we obtain the following factorized direction splitting scheme to be solved until convergence:

$$\begin{aligned}\left(I+\frac{\tau}{2}\boldsymbol{D_r}\right)\left(I+\frac{\tau}{2}\boldsymbol{D_\theta}\right)\left(I+\frac{\tau}{2}\boldsymbol{D_\phi}\right)\left(\mathbf{U}_{k+1}^{n+1}-\mathbf{U}_k^{n+1}\right) &= \\ -\left(I+\frac{\tau}{2}\boldsymbol{D}\right)\left(\mathbf{U}_k^{n+1}-\mathbf{U}^n\right)-\tau\boldsymbol{D}\mathbf{U}^n-\tau\boldsymbol{G}\boldsymbol{r}& \\ -\frac{\tau}{2}\boldsymbol{D_M}\mathbf{U}_k^{n+1}-\frac{\tau}{2}\boldsymbol{D_M}\mathbf{U}^n.&\end{aligned} \tag{3.8}$$

This approach is similar to the one used in Section 1.3.1 in the Cartesian case, and can also be viewed as a preconditioned Richardson iterative method. Note that as a more accurate direction splitting method, the Douglas scheme has a $\mathcal{O}(\tau^2)$ splitting error instead of the $\mathcal{O}(\tau)$ one, and thus it is less vulnerable to the defects at low Mach numbers than the first-order scheme from Chapter 1. Nevertheless, the splitting error reduction strategy described above adds negligible computational cost compared to the nonlinear Picard iterations without the splitting error reduction.

The system (3.8) can be solved as a sequence of three one-dimensional problems:

$$\begin{aligned}\left(I+\frac{\tau}{2}\boldsymbol{D_r}\right)\left(\boldsymbol{\eta}^{n+1}-\mathbf{U}_k^{n+1}\right) &= \\ &-\left(I+\frac{\tau}{2}\boldsymbol{D}\right)\left(\mathbf{U}_k^{n+1}-\mathbf{U}^n\right) \\ &-\tau\boldsymbol{D}\mathbf{U}^n-\tau\boldsymbol{G}\boldsymbol{r}-\frac{\tau}{2}\boldsymbol{D_M}\mathbf{U}_k^{n+1}-\frac{\tau}{2}\boldsymbol{D_M}\mathbf{U}^n,\end{aligned} \tag{3.9}$$

$$\left(I+\frac{\tau}{2}\boldsymbol{D_\theta}\right)\left(\boldsymbol{\zeta}^{n+1}-\mathbf{U}_k^{n+1}\right)=\boldsymbol{\eta}^{n+1}-\mathbf{U}_k^{n+1}, \tag{3.10}$$

$$\left(I + \frac{\tau}{2}\boldsymbol{D}_{\boldsymbol{\phi}}\right)\left(\mathbf{U}_{k+1}^{n+1} - \mathbf{U}_{k}^{n+1}\right) = \boldsymbol{\zeta}^{n+1} - \mathbf{U}_{k}^{n+1} \tag{3.11}$$

where $\boldsymbol{\eta}^{n+1}$, $\boldsymbol{\zeta}^{n+1}$, and $\mathbf{U}_{k+1}^{n+1}$ are subsequent approxomations of the exact solution at t^{n+1}. The structure of (3.9)-(3.11) is very similar to the one of (1.11)-(1.12). Each of these problems requires a solution of a block-tridiagonal linear system only, which can be performed by the block-tridiagonal extension of the Thomas algorithm for tridiagonal systems (e.g. see [31], Volume 1, pp.188-189). The parallel implementation of the Thomas algorithm using the Schur complement technique and domain decomposition, as described in [42], can be easily extended for the block-tridiagonal version of the linear solver. Weak scalability results for the method can be found in Section 3.3.2.

Note that since the staggered discretization on the MAC stencil is employed, no artificial dissipation terms are required, in contrast to the non-staggered case considered in Chapter 1.

3.3 Numerical tests.

The numerical experiments presented below confirm the accuracy of the proposed scheme in a wide range of Mach numbers ($M \in [10^{-6}, 10^{-1}]$), the correct behavior of the numerical solution in the incompressible limit, and excellent parallel performance of the method.

3.3.1 Well-prepared manufactured solution.

The following manufactured solution has been used to verify the implementation and study the properties of the algorithm:

$$\begin{aligned}
\rho =& \rho_0 = 1, \\
p =& p_0 + u_0^2 \left(1 + \sin(5t) + \cos^2(\pi r)\cos^2(4\phi)\cos^2(4\theta)\right), \\
u_r =& \frac{u_0(1+\sin t)}{2r^2} + \frac{u_0^2}{c_0}\left(1 + \sin(4t) + \sin\left(r^2\right)\cos^3(\theta)\sin^2(\phi)\right), \\
u_\theta =& \frac{u_0(1+\cos(3t+2))}{2\sin\theta} + \frac{u_0^2}{c_0}\left(1 + \sin(t) + \cos^3\left(r^2\right)\cos^2(\theta)\sin^3(\phi)\right), \\
u_\phi =& \frac{u_0(1+\sin(6+t))}{2} + \frac{u_0^2}{c_0}\left(1 + \cos(2+t) + \cos\left(r\right)\sin^3(\theta)\sin^2(\phi)\right), \\
T =& \frac{p}{c_v(\gamma-1)}.
\end{aligned}$$

Note that this solution provides well-prepared initial data, i.e. it has the correct scaling with respect to the Mach number. Thus, it can be used to study the behaviour of the scheme in the incompressible ($M_0 \to 0$) limit. Indeed, since the characteristic density $\rho_0 = 1$, the characteristic sound speed becomes $c_0 = \sqrt{\gamma p_0} \sim \sqrt{p_0}$, and the characteristic Mach number is equal to

$M_0 = \dfrac{u_0}{c_0} \sim \dfrac{u_0}{\sqrt{p_0}}$. Then, non-dimensionalized pressure is given by

$$\tilde{p} = \frac{p}{p_0} = \tilde{p}_0 + M_0^2 \tilde{p}_2(r, \theta, \phi, t),$$

and the non-dimensionalized divergence

$$\frac{\nabla \cdot \mathbf{u}}{u_0} \sim M_0,$$

which is in agreement with the results from [50] (see Section 1.2.1 for details).

Similar to Section 1.4.3, the governing equations are modified by the inclusion of source terms, computed using the manufactured solution. In all the tests presented in this chapter $p_0 = 6250$, $\gamma = \dfrac{c_p}{c_v} = 1.6$, $\mu = 1$, $Pr = 1$, $\boldsymbol{\omega} = \mathbf{0}$, $\mathbf{g} = \mathbf{0}$, the inner radius of the shell $R_1 = 1$, and the outer radius of the shell $R_2 = 2$. Two Picard iterations combined with the splitting error reduction are performed at each time step, initialized by $\mathbf{U}_0^{n+1} = \mathbf{U}^n$. Dirichlet boundary conditions are imposed for the velocity components at all the boundaries, and zero Neumann conditions are used for pressure and temperature (satisfied exactly by the manufactured solution). Different values of u_0 may be chosen to study the properties of the algorithm at different characteristic Mach numbers.

First, we examine space and time convergence properties at different values of M_0. Fugure 3.1 demonstrates the expected second order of accuracy in space for pressure (figures 3.1.a, 3.1.b, and 3.1.c) and ϕ-velocity component (figures 3.1.d, 3.1.e, and 3.1.f) for $M_0 = 10^{-2}$, $M_0 = 10^{-4}$, and $M_0 = 10^{-6}$ respectively. Although not shown here, velocity components in r and θ directions, and temperature exhibit similar convergence rates.

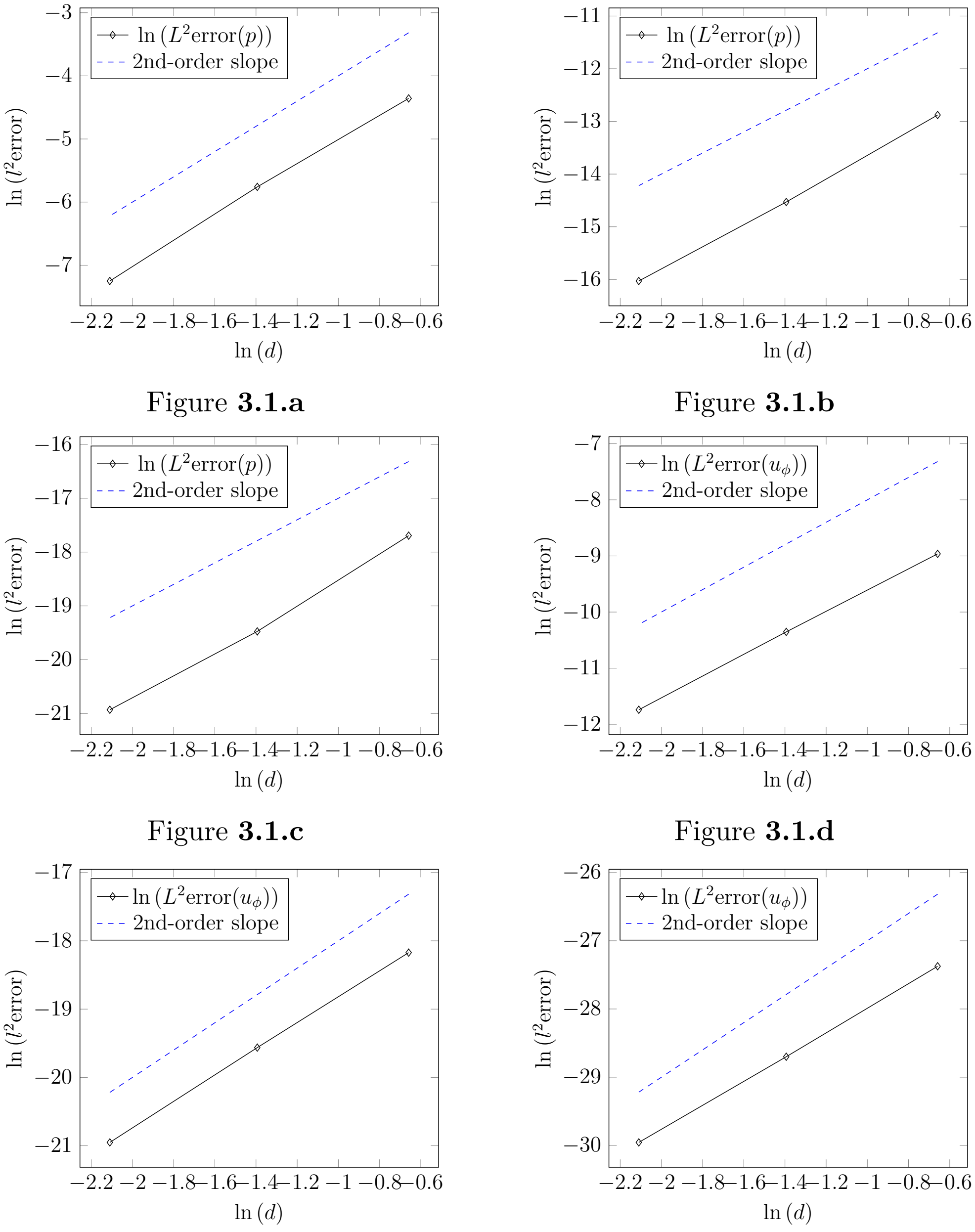

Figure **3.1.a**

Figure **3.1.b**

Figure **3.1.c**

Figure **3.1.d**

Figure **3.1.e**

Figure **3.1.f**

Figure 3.1: *log-log* plots of the discrete L^2 norm of the pressure and u_ϕ errors at $t = 10^{-3}$ ($\tau = 10^{-5}$) for $M = 10^{-2}$, $M = 10^{-4}$, and $M = 10^{-6}$ manufactured solutions.

τ	$M_0 = 10^{-2}$	$M_0 = 10^{-4}$	$M_0 = 10^{-6}$
$2 \cdot 10^{-3}$	(1.8, 1.7)	(3.2, 3.2)	(3.2, 3.2)
$1 \cdot 10^{-3}$	(1.9, 1.9)	(2.7, 2.6)	(2.7, 2.5)
$5 \cdot 10^{-4}$	(1.9, 1.9)	(1.6, 1.5)	(1.4, 1.5)

Table 3.1: Order of time convergence for (p, T) computed using the inverse Richardson extrapolation approach.

τ	$M_0 = 10^{-2}$	$M_0 = 10^{-4}$	$M_0 = 10^{-6}$
2×10^{-3}	(2.3, 2.9, 2.6)	(3.2, 2.5, 2.9)	(2.5, 2.9, 2.4)
1×10^{-3}	(2.4, 2.7, 3.2)	(2.4, 2.8, 2.6)	(2.4, 2.8, 2.3)
5×10^{-4}	(2.1, 2.4, 2.3)	(2.2, 2.6, 2.1)	(2.2, 2.6, 2.1)

Table 3.2: Order of time convergence for (u_r, u_θ, u_ϕ) computed using the inverse Richardson exptrapolation approach.

Next, we follow [88] to estimate the order of temporal accuracy using the following time convergence rate (TCR) estimate:

$$TCR(u_i, \tau) = \log_2 \left[\frac{||u_i^{\tau} - u_i^{\frac{\tau}{2}}||}{||u_i^{\frac{\tau}{2}} - u_i^{\frac{\tau}{4}}||} \right]$$

Due to the form of the TCR, spatial discretization errors cancel (i.e. the leading order truncation error is $const \cdot \tau^l$, where l is the order of accuracy in time), and $TCR \approx l$. This approach is a form of "inverse Richardson extrapolation", and allows temporal benchmarking of the algorithm without extreme grid refinement in 3D. The TCR parameters for pressure and temperature corresponding to different values of τ and M_0 are listed in Table 3.1, while Table 3.2 gives the TCR values for the velocity components. The results demonstrate that the temporal convergence is consistent with the theoretically expected second order.

Hence, the scheme retains its convergence properties in case of extremely low Mach numbers with no extra computational cost. Although the stability

n	$M_0 = 10^{-2}$	$M_0 = 10^{-3}$	$M_0 = 10^{-4}$	$M_0 = 10^{-5}$	$M_0 = 10^{-6}$
1	$3.2 \cdot 10^{-4}$	$3.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-8}$	$3.2 \cdot 10^{-10}$	$3.9 \cdot 10^{-12}$
50	$3.6 \cdot 10^{-4}$	$3.6 \cdot 10^{-6}$	$3.7 \cdot 10^{-8}$	$4.2 \cdot 10^{-10}$	$9.6 \cdot 10^{-12}$
100	$4.0 \cdot 10^{-4}$	$4.0 \cdot 10^{-6}$	$4.1 \cdot 10^{-8}$	$4.9 \cdot 10^{-10}$	$1.4 \cdot 10^{-11}$

Table 3.3: Maximum norm of relative pressure variations ($\Delta p = \frac{p-p_0}{p_0}$) after n time steps for different values of M_0. $\tau = 10^{-3}$, grid diameter: 0.12.

of the scheme has not been studied rigorously, based on our numerical experience the scheme is conditionally stable, as expected due to the presence of the advection terms. The stability restriction does not depend on the Mach number, at least based on the numerical tests performed for this chapter. Some dependence on the scaling of the problem, in particular on the value of p_0, was observed.

Furthermore, the numerical solution remains in the incompressible regime. Table 3.3 provides the maximum norm of the relative pressure fluctuations ($\Delta p = \frac{p-p_0}{p_0}$) after n time steps for different values of M_0. Theoretically predicted order of magnitude ($\Delta p \sim \mathcal{O}(M_0^2)$) is well preserved by the scheme. The method does not introduce any artificial acoustic waves ($\mathcal{O}(M_0)$ pressure fluctuations) even for extremely low values of M_0.

# cores	1(1×1×1)	32(2×4×4)	256(4×8×8)	512(4×8×16)	1024(4×16×16)
Efficiency (one core)	–	91 %	85 %	82 %	77 %
Efficiency (one node)	–	–	93 %	90 %	85 %

Table 3.4: Weak Scalability test, $3375 \cdot 10^3$ grid points per core.

3.3.2 Weak scalability.

We evaluate the parallel performance of the method by providing weak scalability results obtained using the same methodology as in Sections 1.4.4 and 2.4.2. Since we are interested in large-scale computations, we consider $3375 \cdot 10^3$ grid points per core to maximize the size of the problem given the memory limitations. Then measure the efficiency on 32 cores (1 computational node), 256 cores (8 nodes), 512 cores (16 nodes), and 1024 cores (32 nodes). The efficiency results are given in Table 3.4. The efficiency is given relative to 1 core and relative to 1 node since the drop in efficiency from 1 to 32 cores is likely caused by the need to share the memory bandwidth and cache with a smaller number of cores within the computational node, rather than scaling properties of the method (see e.g. [65], p. 152, and the discussion in Section 2.4.2). The weak scaling test demonstrates excellent parallel performance.

The scaling tests are performed using the Compute Canada Graham cluster (see https://www.computecanada.ca/) of 2.1GHz Intel $E5 - 2683$ v4 CPU cores, 32 cores per node, and each node connected via a 100 Gb/s network. The results were calculated using the wall clock time taken to simulate 10 time steps with two Picard iterations each. These computations were performed three times for each configuration, and the average wall clock time was used to compute the efficiency.

3.4 Conclusion.

The numerical experiments presented above demonstrate the effectiveness of implicit methods based on the direction splitting approach for modeling compressible flows in spherical shells in nearly incompressible and weakly compressible regimes. The proposed algorithm retains theoretically expected convergence rates and remains stable for extremely small values of the characteristic Mach number (at least as low as $M_0 = 10^{-6}$). The staggered spatial discretization on the MAC stencil, commonly used in numerical methods for incompressible Navier-Stokes equations, was found to be convenient for the discretization of the compressible Navier-Stokes equations written in the nonconservative form in terms of the primitive variables. This approach helped to avoid the high-frequency oscillations without any artificial stabilization terms. Nonlinear Picard iterations with the splitting error reduction were also implemented to allow one to obtain a solution of the fully nonlinear system of equations.

These results, alongside excellent parallel performance, prove the viability of the direction splitting approach in large-scale high-resolution high-performance simulations of atmospheric and oceanic flows. Possibilities for future studies and developments include research of monotonicity preserving properties of the scheme to evaluate the need for stabilization terms for flows under extreme conditions, such as high Reynolds numbers. The influence of the linearization error on the monotonicity and stability is worth investigation as well. The computational domain should be modified to represent realistic topography, and the adaptive mesh refinement is likely to be necessary for practical applications in oceanography and atmospheric sciences.

Conclusion

The main goal of this work was the development of all-speed numerical methods for the compressible Navier-Stokes equations, i.e. methods that remain efficient in incompressible, weakly compressible, and compressible regimes. This goal has been achieved by the proposed numerical schemes in Cartesian (Chapter 1) and spherical (Chapters 2 and 3) coordinates.

The main contributions of Chapter 1 are the development of an LBI-scheme suitable for computations of low Mach number flows (it was believed before that direction splitting schemes were not suitable for such applications, see e.g. remarks in [121], [56]), and the design of a novel low-Mach number stabilization term that is effective in eliminating the odd-even decoupling on non-staggered grids. Furthermore, the method has been extended to the multicomponent case. The proposed technique has been found to have promising parallel performance and low computational cost.

Chapter 2 introduced a second-order direction splitting method for solving the incompressible Navier-Stokes-Boussinesq system in spherical geometries. Some theoretical stability and convergence results have been proven for a linear parabolic equation, robustness and accuracy of the scheme for a more complicated case of the Boussinesq system has been demonstrated numerically. This proves that the direction splitting approach retains its appealing properties when formulated for spherical coordinates. The direction splitting scheme was coupled with the artificial compressibility regularization of the incompressible Navier-Stokes system inspired by the analysis of the low Mach number limit of the compressible NSEs.

Finally, Chapter 3 develops the ideas of the first two chapters to formulate a numerical method for nearly incompressible and weakly compressible flows in spherical shells. The algorithm is based on the direction splitting approach similar to Chapter 1 but applied to the spherical coordinates and MAC-stencil based staggered discretization similar to the one used in Chapter 2. Nonlinear Picard iterations were combined with the splitting error reduction technique to obtain a solution of the fully nonlinear system. To the best of our knowledge, this approach is completely new and has not been used before for simulations of weakly compressible flows in spherical geometries. Numerical experiments presented in Chapter 3 confirm that the scheme retains stability and convergence for extremely low values of the Mach number, preserves the incompressibility of the initial data, and has excellent parallel performance. Thus, we hope that it may serve as a foundation for the next generation of dynamical cores for weather and climate models, as well as be useful in other applications.

Bibliography

[1] A. Abdolali and J.T. Kirby. Role of compressibility on tsunami propagation. *Journal of Geophysical Research: Oceans*, 122(12):9780–9794, 2017.

[2] R. Abgrall. How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach. *Journal of Computational Physics*, 125(1):150–160, 1996.

[3] R. Abgrall and S. Karni. Computations of compressible multifluids. *Journal of Computational Physics*, 169(2):594–623, 2001.

[4] T. Alazard. Low Mach number limit of the full Navier–Stokes equations. *Archive for Rational Mechanics and Analysis*, 180(1):1–73, 2006.

[5] T. Alazard. A minicourse on the low Mach number limit. 2008.

[6] G. Allaire, S. Clerc, and S. Kokh. A five-equation model for the simulation of interfaces between compressible fluids. *Journal of Computational Physics*, 181(2):577–616, 2002.

[7] Y. Baba, K. Takahashi, T. Sugimura, and K. Goto. Dynamical core of an atmospheric general circulation model on a Yin–Yang grid. *Monthly Weather Review*, 138(10):3988–4005, 2010.

[8] J.R. Baumgardner. Three-dimensional treatment of convective flow in the Earth's mantle. *Journal of Statistical Physics*, 39(5-6):501–511, 1985.

[9] R.M. Beam and R.F. Warming. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, 22(1):87–110, 1976.

[10] R.M. Beam and R.F. Warming. An implicit factored scheme for the compressible Navier–Stokes equations. *AIAA journal*, 16(4):393–402, 1978.

[11] J. Boussinesq. *Théorie analytique de la chaleur mise en harmonic avec la thermodynamique et avec la théorie mécanique de la lumière: Tome I-[II]*, volume 2. Gauthier-Villars, 1903.

[12] W.R. Briley and H. McDonald. On the structure and use of linearized block implicit schemes. *Journal of Computational Physics*, 34(1):54–73, 1980.

[13] W.R. Briley and H. McDonald. Reflections on the evolution of implicit Navier–Stokes algorithms. *Computers & Fluids*, 41(1):15–19, 2011.

[14] S. R Chakravarthy. Euler equations–Implicit schemes and boundary conditions. *AIAA Journal(ISSN 0001-1452)*, 21:699–706, 1983.

[15] A. Chertock, S. Karni, and A. Kurganov. Interface tracking method for compressible multifluids. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(6):991–1019, 2008.

[16] G. Chesshire and W.D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64, 1990.

[17] D. Choi and C.L. Merkle. Application of time-iterative schemes to incompressible flow. *AIAA Journal*, 23(10):1518–1524, 1985.

[18] A. J. Chorin. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968.

[19] V. Coralic and T. Colonius. Finite-volume WENO scheme for viscous compressible multicomponent flows. *Journal of Computational Physics*, 274:95–121, 2014.

[20] F. Cordier, P. Degond, and A. Kumbaro. An asymptotic-preserving all-speed scheme for the Euler and Navier–Stokes equations. *Journal of Computational Physics*, 231(17):5685–5704, 2012.

[21] R. Danchin. Zero Mach number limit for compressible flows with periodic boundary conditions. *American Journal of Mathematics*, 124(6):1153–1219, 2002.

[22] R. Danchin. Low Mach number limit for viscous compressible flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(3):459–475, 2005.

[23] V. DeCaria, W. Layton, and M. McLaughlin. A conservative, second–order, unconditionally stable artificial compression method. *Computer Methods in Applied Mechanics and Engineering.*, 325:733–747, 2017.

[24] B. Desjardins and E. Grenier. Low Mach number limit of viscous compressible flows in the whole space. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 455(1986):2271–2279, 1999.

[25] B. Desjardins, E. Grenier, P.-L. Lions, and N. Masmoudi. Incompressible limit for solutions of the isentropic Navier–Stokes equations with Dirichlet boundary conditions. *Journal de Mathématiques Pures et Appliquées*, 78(5):461–471, 1999.

[26] B. Desjardins, C.-K. Lin, et al. A survey of the compressible Navier–Stokes equations. *TAIWANESE Journal of Mathematics*, 3(2):123–137, 1999.

[27] G. Dimarco, R. Loubère, and M.-H. Vignal. Study of a new asymptotic preserving scheme for the Euler system in the low Mach number limit. 2016.

[28] J. Douglas. Alternating direction methods for three space variables. *Numerische Mathematik*, 4(1):41–63, 1962.

[29] E. Feireisl, O. Kreml, V. Mácha, and Š. Nečasová. On the low Mach number limit of compressible flows in exterior moving domains. *Journal of Evolution Equations*, 16(3):705–722, 2016.

[30] E. Feireisl and A. Novotnỳ. The Oberbeck–Boussinesq approximation as a singular limit of the full Navier–Stokes–Fourier system. *Journal of Mathematical Fluid Mechanics*, 11(2):274–302, 2009.

[31] C. A.J. Fletcher. Computational techniques for fluid dynamics. Volume 1-Fundamental and general techniques. Volume 2-Specific techniques for different flow categories. In *Berlin and New York, Springer-Verlag, 1988, p. Vol. 1, 418 p.; vol. 2, 493 p.*, volume 1, 1988.

[32] H.H. Francis and A.A. Anthony. Technical report LA-4700. Technical report, 1971.

[33] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, and M.W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, 2006.

[34] J. Freund, S. Lele, P. Moin, J. Freund, S. Lele, and P. Moin. Direct simulation of a supersonic round turbulent shear layer. In *35th Aerospace Sciences Meeting and Exhibit*, page 760, 1997.

[35] Roman Frolov. An efficient algorithm for the multicomponent compressible navier–stokes equations in low-and high-mach number regimes. *Computers & Fluids*, 178:15–40, 2019.

[36] J.-F. Gerbeau, N. Glinsky-Olivier, and B. Larrouturou. Semi-implicit Roe-type fluxes for low-Mach number flows. 1997.

[37] R. Glowinski. Finite element methods for incompressible viscous flow. *Handbook of Numerical Analysis*, 9:3–1176, 2003.

[38] E. Grenier. Oscillatory perturbations of the Navier–Stokes equations. *Journal de Mathématiques Pures et Appliquées*, 76(6):477–498, 1997.

[39] S.M. Griffies and A.J. Adcroft. Formulating the equations of ocean models. *Ocean Modeling in an Eddying Regime*, 177:281–317, 2008.

[40] M.D. Griffin, E. Jones, and J.D. Anderson Jr. A computational fluid dynamic technique valid at the centerline for non-axisymmetric problems in cylindrical coordinates. *Journal of Computational Physics*, 30(3):352–360, 1979.

[41] J.-L. Guermond and P. D. Minev. High-order time stepping for the Navier–Stokes equations with minimal computational complexity. *Journal of Computational and Applied Mathematics*, 310:92–103, 2017.

[42] J.-L. Guermond and P.D. Minev. A new class of massively parallel direction splitting for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 200(23):2083–2093, 2011.

[43] J.-L. Guermond and P.D. Minev. Start-up flow in a three-dimensional lid-driven cavity by means of a massively parallel direction splitting algorithm. *International Journal for Numerical Methods in Fluids*, 68(7):856–871, 2012.

[44] J.-L. Guermond and P.D. Minev. High-order time stepping for the incompressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 37(6):A2656–A2681, 2015.

[45] J.-L. Guermond, M. Nazarov, B. Popov, and I. Tomas. Second-order invariant domain preserving approximation of the Euler equations using convex limiting. *ArXiv e-prints*, October 2017.

[46] J.-L. Guermond and B. Popov. Viscous regularization of the Euler equations and entropy principles. *SIAM Journal on Applied Mathematics*, 74(2):284–305, 2014.

[47] J.-L. Guermond and B. Popov. Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations. *Journal of Computational Physics*, 321:908–926, 2016.

[48] J.-L. Guermond and B. Popov. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 54(4):2466–2489, 2016.

[49] H. Guillard and A. Murrone. On the behavior of upwind schemes in the low Mach number limit: Ii. Godunov type schemes. *Computers & Fluids*, 33(4):655–675, 2004.

[50] H. Guillard and C. Viozat. On the behaviour of upwind schemes in the low Mach number limit. *Computers & Fluids*, 28(1):63–86, 1999.

[51] K.K. Haller, Y. Ventikos, and D. Poulikakos. Wave structure in the contact line region during high speed droplet impact on a surface: Solution of the Riemann problem for the stiffened gas equation of state. *Journal of Applied Physics*, 93(5):3090–3097, 2003.

[52] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.

[53] D. Hoff. The zero-Mach limit of compressible flows. *Communications in Mathematical Physics*, 192(3):543–554, 1998.

[54] T.Y. Hou and P.G. LeFloch. Why nonconservative schemes converge to wrong solutions: error analysis. *Mathematics of Computation*, 62(206):497–530, 1994.

[55] W. Huang and D.M. Sloan. Pole condition for singular problems: the pseudospectral approximation. *Journal of Computational Physics*, 107(2):254–261, 1993.

[56] A. Hujeirat, F.-K. Thielemann, J. Dusek, and A. Nusser. Compressed low Mach number flows in astrophysics: a nonlinear Newtonian numerical solver. *arXiv preprint arXiv:0712.3663*, 2007.

[57] H. Isozaki. Singular limits for the compressible Euler equation in an exterior domain. *Journées équations aux dérivées partielles*, pages 1–9, 1986.

[58] H. Isozaki. Singular limits for the compressible Euler equation in an exterior domain. ii. Bodies in a uniform flow. *Osaka Journal of Mathematics*, 26(2):399–410, 1989.

[59] D. Jacqmin. An energy approach to the continuum surface tension method. *AIAA paper*, (96-0858), 1996.

[60] D. Jamet, D. Torres, and J.U. Brackbill. On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method. *Journal of Computational Physics*, 182(1):262–276, 2002.

[61] E. Johnsen and T. Colonius. Numerical simulations of non-spherical bubble collapse. *Journal of Fluid Mechanics*, 629:231–262, 2009.

[62] Y. Kagei and M. Ružička. The Oberbeck–Boussinesq approximation as a constitutive limit. *Continuum Mechanics and Thermodynamics*, 28(5):1411–1419, 2016.

[63] A. Kageyama and T. Sato. "Yin–Yang grid": An overset grid in spherical geometry. *Geochemistry, Geophysics, Geosystems*, 5(9), 2004.

[64] S. Karni. Multicomponent flow calculations by a consistent primitive algorithm. *Journal of Computational Physics*, 112(1):31–43, 1994.

[65] J.W. Keating. Direction-splitting schemes for particulate flows. 2013.

[66] S. Klainerman and A. Majda. Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids. *Communications on Pure and Applied Mathematics*, 34(4):481–524, 1981.

[67] S. Klainerman and A. Majda. Compressible and incompressible fluids. *Communications on Pure and Applied Mathematics*, 35(5):629–651, 1982.

[68] R. Klein. Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics i: One-dimensional flow. *Journal of Computational Physics*, 121(2):213–237, 1995.

[69] K.-B. Kong, J.-S. Kim, and S.-O. Park. Reduced splitting error in the ADI–FDTD method using iterative method. *Microwave and Optical Technology Letters*, 50(8):2200–2203, 2008.

[70] H.-O. Kreiss, J. Lorenz, and M.J. Naughton. Convergence of the solutions of the compressible to the solutions of the incompressible Navier–Stokes equations. *Advances in Applied Mathematics*, 12(2):187–214, 1991.

[71] O.A. Ladyzhenskaya. *The mathematical theory of viscous incompressible flow*, volume 2. Gordon and Breach New York, 1969.

[72] L. Landau. A new exact solution of the Navier–Stokes equations. *Doklady Academii Nauk USSR*, 43:286–295, 1944.

[73] H. P. Langtangen. Solving nonlinear ODE and PDE problems. *Center for Biomedical Computing, Simula Research Laboratory and Department of Informatics, University of Oslo*, 2016.

[74] F. Lemarié, E. Blayo, and L. Debreu. Analysis of ocean-atmosphere coupling algorithms: consistency and stability. *Procedia Computer Science*, 51:2066–2075, 2015.

[75] F. Lemarié, P. Marchesiello, L. Debreu, and E. Blayo. Sensitivity of ocean-atmosphere coupled models to the coupling method: example of tropical cyclone Erica. 2014.

[76] L. Li, Y.Y. Li, and X. Yan. Homogeneous solutions of stationary Navier–Stokes equations with isolated singularities on the unit sphere. I. One singularity. *Archive for Rational Mechanics and Analysis.*, 227:1091–1163, 2018.

[77] P.-L. Lions and N. Masmoudi. Incompressible limit for a viscous compressible fluid. *Journal de mathématiques pures et appliquées*, 77(6):585–627, 1998.

[78] T.G. Liu, B.C. Khoo, and K.S. Yeo. Ghost fluid method for strong shock impacting on material interface. *Journal of Computational Physics*, 190(2):651–681, 2003.

[79] X.-D. Liu, R.P. Fedkiw, and S. Osher. A conservative approach to the multiphase Euler equations without spurious pressure oscillations. 1998.

[80] G.I. Marchuk and N.N. Yanenko. The solution of a multi-dimensional kinetic equation by the splitting method. In *Dokl. Akad. Nauk SSSR*, volume 157, pages 1291–1292, 1964.

[81] S. Marras, J.F. Kelly, M. Moragues, A. Müller, M.A. Kopera, M. Vázquez, F.X. Giraldo, G. Houzeaux, and O. Jorba. A review of element-based Galerkin methods for numerical weather prediction: Finite elements, spectral elements, and discontinuous Galerkin. *Archives of Computational Methods in Engineering*, 23(4):673–722, 2016.

[82] J. Marshall, A. Adcroft, C. Hill, L. Perelman, and C. Heisey. A finite-volume, incompressible Navier-Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, 102(C3):5753–5766, 1997.

[83] T. Mathew. *Domain decomposition methods for the numerical solution of partial differential equations*, volume 61. Springer Science & Business Media, 2008.

[84] B.E. Merrill, Y.T. Peet, P.F. Fischer, and J.W. Lottes. A spectrally accurate method for overlapping grid solution of incompressible Navier–Stokes equations. *Journal of Computational Physics*, 307:60–93, 2016.

[85] G. Métivier and S. Schochet. The incompressible limit of the non-isentropic Euler equations. *Archive for Rational Mechanics and Analysis*, 158(1):61–90, 2001.

[86] F. Miczek. *Simulation of low Mach number astrophysical flows.* PhD thesis, Universität München, 2013.

[87] F. Miczek, F.K. Röpke, and P.V.F. Edelmann. New numerical solver for flows at various Mach numbers. *Astronomy & Astrophysics*, 576:A50, 2015.

[88] PD Minev and C Ross Ethier. A characteristic/finite element algorithm for the 3-d navier–stokes equations using unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 178(1-2):39–50, 1999.

[89] K. Mohseni and T. Colonius. Numerical treatment of polar coordinate singularities. *Journal of Computational Physics*, 157(2):787–795, 2000.

[90] S. Noelle, G. Bispen, K. R. Arun, M. Lukáčová-Medviďová, and C.-D. Munz. A weakly asymptotic preserving low mach number scheme for the euler equations of gas dynamics. *SIAM Journal on Scientific Computing*, 36(6):B989–B1024, 2014.

[91] R.R. Nourgaliev, S. Y. Sushchikh, T.-N. Dinh, and T.G. Theofanous. Shock wave refraction patterns at interfaces. *International Journal of Multiphase Flow*, 31(9):969–995, 2005.

[92] A. Oberbeck. Über die Wärmeleitung der Flüssigkeiten bei Berücksichtigung der Strömungen infolge von Temperaturdifferenzen. *Annalen der Physik*, 243(6):271–292, 1879.

[93] N. Ohno and A. Kageyama. Visualization of spherical data by Yin–Yang grid. *Computer Physics Communications*, 180(9):1534–1538, 2009.

[94] T. Ohwada and P. Asinari. Artificial compressibility method revisited: Asymptotic numerical method for incompressible Navier–Stokes equations. *Journal of Computational Physics*, 229:1698–1723, 2010.

[95] I. Orlanski. The rationale for why climate models should adequately resolve the mesoscale. In *High resolution numerical modelling of the atmosphere and ocean*, pages 29–44. Springer, 2008.

[96] G. Perigaud and R. Saurel. A compressible flow model with capillary effects. *Journal of Computational Physics*, 209(1):139–178, 2005.

[97] K.R. Rajagopal, M. Ruzicka, and A.R. Srinivasa. On the Oberbeck–Boussinesq approximation. *Mathematical Models and Methods in Applied Sciences*, 6(08):1157–1167, 1996.

[98] C. Ronchi, R. Iacono, and P.S. Paolucci. The “cubed sphere”: a new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics*, 124(1):93–114, 1996.

[99] C. Roy, C. Ober, and T. Smith. Verification of a compressible CFD code using the method of manufactured solutions. In *32nd AIAA Fluid Dynamics Conference and Exhibit*, page 3110, 2002.

[100] V.S Ryaben’kii and S.V. Tsynkov. *A theoretical introduction to numerical analysis*. CRC Press, 2006.

[101] A.A Samarskii and Vabischevich. P.N. *Additivnie skhemi dlya zadach matematicheskoy fiziki (in Russian)*. Moskva Nauka, 1999.

[102] R. Saurel and R. Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21(3):1115–1145, 1999.

[103] N. Schaeffer, D. Jault, H.-C. Nataf, and A. Fournier. Turbulent geodynamo simulations: a leap towards Earth's core. *Geophysical Journal International*, 211(1):1–29, 2017.

[104] S. Schochet. The compressible Euler equations in a bounded domain: existence of solutions and the incompressible limit. *Communications in Mathematical Physics*, 104(1):49–75, 1986.

[105] S. Schochet. Fast singular limits of hyperbolic PDEs. *Journal of Differential Equations*, 114(2):476–512, 1994.

[106] S. Schochet. The mathematical theory of low Mach number flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(3):441–458, 2005.

[107] J. Shen. A remark on the projection-3 method. *International Journal for Numerical Methods in Fluids*, 16(3):249–253, 1993.

[108] J. Shen. On error estimates of the penalty method for unsteady Navier–Stokes equations. *SIAM Journal on Numerical Analysis*, 32(2):386–403, 1995.

[109] S. Shin and D. Juric. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *Journal of Computational Physics*, 180(2):427–470, 2002.

[110] R.K. Shukla, C. Pantano, and J.B. Freund. An interface capturing method for the simulation of multi-phase compressible flows. *Journal of Computational Physics*, 229(19):7411–7439, 2010.

[111] G.A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1–31, 1978.

[112] Y.T. Song and T.Y. Hou. Parametric vertical coordinate formulation for multiscale, Boussinesq, and non-Boussinesq ocean modeling. *Ocean Modelling*, 11(3-4):298–332, 2006.

[113] E.A. Spiegel and G. Veronis. On the Boussinesq approximation for a compressible fluid. *The Astrophysical Journal*, 131:442, 1960.

[114] G. Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

[115] P.J. Tackley. Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the Yin–Yang grid. *Physics of the Earth and Planetary Interiors*, 171(1-4):7–18, 2008.

[116] H.S. Tang, S.C. Jones, and F. Sotiropoulos. An overset-grid method for 3D unsteady incompressible flows. *Journal of Computational Physics*, 191(2):567–600, 2003.

[117] R. Temam. Une méthode d'approximation de la solution des équations de Navier–Stokes. *Bulletin de la Société Mathématique de France*, 96:115–152, 1968.

[118] R. Temam. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires (i). *Archive for Rational Mechanics and Analysis*, 32(2):135–153, 1969.

[119] K.W. Thompson. Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, 68(1):1–24, 1987.

[120] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction.* Springer Science & Business Media, 2013.

[121] E. Turkel. Preconditioning techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 31(1):385–416, 1999.

[122] E. Turkel, R. Radespiel, and N. Kroll. Assessment of preconditioning methods for multidimensional aerodynamics. *Computers & Fluids*, 26(6):613–634, 1997.

[123] S. Ukai et al. The incompressible limit and the initial layer of the compressible Euler equation. *Journal of Mathematics of Kyoto University*, 26(2):323–331, 1986.

[124] P. Vabishchevich. Additive schemes (splitting schemes) for some systems of evolutionary equations. *Mathematics of Computation*, 83(290):2787–2797, 2014.

[125] A. Wongwathanarat, E. Mueller, and H.-T. Janka. Three-dimensional simulations of core-collapse supernovae: from shock revival to shock breakout. *Astronomy & Astrophysics*, 577:A48, 2015.

[126] H. Zakerzadeh and S. Noelle. A note on the stability of implicit–explicit flux splittings for stiff hyperbolic systems. *IGPM Preprint*, (449):2–1, 2016.

[127] M. Zerroukat and T. Allen. On the solution of elliptic problems on overset/Yin–Yang grids. *Monthly Weather Review*, 140(8):2756–2767, 2012.

[128] R.K. Zeytounian. Joseph Boussinesq and his approximation: a contemporary view. *Comptes Rendus Mecanique*, 331(8):575–586, 2003.

[129] S. Zhong, M.T. Zuber, L. Moresi, and M. Gurnis. Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection. *Journal of Geophysical Research: Solid Earth*, 105(B5):11063–11082, 2000.

Appendices

Appendix A. Governing Equations and computation of Jacobians for Chapter 1.

The governing equations (compressible Navier-Stokes and advection of SG EOS coefficients) can be written as:

$$\begin{aligned}\frac{\partial \mathbf{U}}{\partial t}+\mathbf{A}\frac{\partial \mathbf{F}(\mathbf{U})}{\partial x}+\mathbf{B}\frac{\partial \mathbf{G}(\mathbf{U})}{\partial y}=\frac{\partial \mathbf{V}_1(\mathbf{U},\mathbf{U}_x)}{\partial x}+\frac{\partial \mathbf{V}_2(\mathbf{U},\mathbf{U}_y)}{\partial x}\\ +\frac{\partial \mathbf{W}_1(\mathbf{U},\mathbf{U}_x)}{\partial y}+\frac{\partial \mathbf{W}_2(\mathbf{U},\mathbf{U}_y)}{\partial y}+\mathbf{F}^{ST},\end{aligned} \tag{A.1}$$

where

$$\mathbf{U}=\begin{bmatrix}\rho & m & n & E & \alpha & \beta\end{bmatrix}^T \tag{A.2}$$

$$\mathbf{F}=\begin{bmatrix}m & \tfrac{m^2}{\rho}+p & \tfrac{mn}{\rho} & (E+p)\tfrac{m}{\rho} & \alpha & \beta\end{bmatrix}^T \tag{A.3}$$

$$\mathbf{G}=\begin{bmatrix}n & \tfrac{nm}{\rho} & \tfrac{n^2}{\rho}+p & (E+p)\tfrac{n}{\rho} & \alpha & \beta\end{bmatrix}^T \tag{A.4}$$

$$\mathbf{F}^{ST}=\begin{bmatrix}0 & \sigma\kappa\partial_x\phi & \sigma\kappa\partial_y\phi & \sigma\kappa u\partial_x\phi+\sigma\kappa v\partial_y\phi & 0 & 0\end{bmatrix}^T \tag{A.5}$$

$$\mathbf{V}_1=\begin{bmatrix}0 & \tfrac{4\mu}{3}\partial_x\tfrac{m}{\rho} & \mu\partial_x\tfrac{n}{\rho} & \tfrac{4\mu m}{3\rho}\partial_x\tfrac{m}{\rho}+\mu\tfrac{n}{\rho}\partial_x\tfrac{n}{\rho} & 0 & 0\end{bmatrix}^T \tag{A.6}$$

$$\mathbf{V}_2 = \begin{bmatrix} 0 & \frac{-2\mu}{3}\partial_y \frac{n}{\rho} & \mu \partial_y \frac{m}{\rho} & \frac{-2\mu m}{3\rho}\partial_y \frac{n}{\rho} + \mu \frac{n}{\rho}\partial_y \frac{m}{\rho} & 0 & 0 \end{bmatrix}^T \tag{A.7}$$

$$\mathbf{W}_1 = \begin{bmatrix} 0 & \mu \partial_x \frac{n}{\rho} & -\frac{2\mu}{3}\partial_x \frac{m}{\rho} & \frac{-2\mu n}{3\rho}\partial_x \frac{m}{\rho} + \mu \frac{m}{\rho}\partial_x \frac{n}{\rho} & 0 & 0 \end{bmatrix}^T \tag{A.8}$$

$$\mathbf{W}_2 = \begin{bmatrix} 0 & \mu \partial_y \frac{m}{\rho} & \frac{4\mu}{3}\partial_y \frac{n}{\rho} & \frac{\mu m}{\rho}\partial_y \frac{m}{\rho} + \frac{4\mu n}{3\rho}\partial_y \frac{n}{\rho} & 0 & 0 \end{bmatrix}^T \tag{A.9}$$

where μ $[kg/(s \cdot m)]$ is a dynamic viscosity coefficient, σ $[N/m]$ is the surface tension coeficient, ϕ is the volume of fluid function, $\kappa = -\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$ is the interfacial curvature.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{m}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{m}{\rho} \end{bmatrix} \tag{A.10}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{n}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{n}{\rho} \end{bmatrix} \tag{A.11}$$

$$p = \frac{1}{\alpha}\left(E - \frac{m^2 + n^2}{2\rho} - \beta\right). \tag{A.12}$$

After time discretization (A.1) becomes:

$$\begin{gathered}
\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\tau} + \mathbf{A}^n \frac{\partial \mathbf{F}^{n+1}(\mathbf{U})}{\partial x} + \mathbf{B}^n \frac{\partial \mathbf{G}^{n+1}(\mathbf{U})}{\partial y} = \\
\frac{\partial \mathbf{V}_1^{n+1}(\mathbf{U}, \mathbf{U}_x)}{\partial x} + \frac{\partial \mathbf{V}_2^{n}(\mathbf{U}, \mathbf{U}_y)}{\partial x} + \frac{\partial \mathbf{W}_1^{n}(\mathbf{U}, \mathbf{U}_x)}{\partial y} + \frac{\partial \mathbf{W}_2^{n+1}(\mathbf{U}, \mathbf{U}_y)}{\partial y},
\end{gathered} \tag{A.13}$$

which is linearized as:

$$\mathbf{F}^{n+1} = \mathbf{F}^n + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) \tag{A.14}$$

$$\mathbf{G}^{n+1} = \mathbf{G}^n + \left(\frac{\partial \mathbf{G}}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) \tag{A.15}$$

$$\mathbf{V}_1^{n+1} = \mathbf{V}_1^n + \left(\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) + \left(\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}_x}\right)^n (\mathbf{U}_x^{n+1} - \mathbf{U}_x^n) \tag{A.16}$$

$$\mathbf{W}_2^{n+1} = \mathbf{W}_2^n + \left(\frac{\partial \mathbf{W}_2}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) + \left(\frac{\partial \mathbf{W}_2}{\partial \mathbf{U}_y}\right)^n (\mathbf{U}_y^{n+1} - \mathbf{U}_y^n), \tag{A.17}$$

where

$$\frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ f_1 & \frac{2m}{\rho} - \frac{m}{\alpha\rho} & -\frac{n}{\alpha\rho} & \frac{1}{\alpha} & f_2 & -\frac{1}{\alpha} \\ -\frac{mn}{\rho^2} & \frac{n}{\rho} & \frac{m}{\rho} & 0 & 0 & 0 \\ f_3 & \frac{E+p}{\rho} - \frac{m^2}{\alpha\rho^2} & -\frac{mn}{\alpha\rho^2} & \frac{m}{\rho}\left(1 + \frac{1}{\alpha}\right) & f_4 & -\frac{m}{\alpha\rho} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.18}$$

where $f_1 = -\frac{m^2}{\rho^2} + \frac{1}{\alpha}\left(\frac{m^2+n^2}{2\rho^2}\right)$, $f_2 = -\frac{1}{\alpha^2}\left(E - \frac{m^2+n^2}{2\rho} - \beta\right)$,
$f_3 = -\frac{(E+p)m}{\rho^2} + \frac{m^3+mn^2}{2\alpha\rho^3}$, $f_4 = -\frac{m}{\alpha^2\rho}\left(E - \frac{m^2+n^2}{2\rho} - \beta\right)$.

$$\frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{mn}{\rho^2} & \frac{n}{\rho} & \frac{m}{\rho} & 0 & 0 & 0 \\ g_1 & -\frac{m}{\alpha\rho} & \frac{2n}{\rho} - \frac{n}{\alpha\rho} & \frac{1}{\alpha} & g_2 & -\frac{1}{\alpha} \\ g_3 & -\frac{mn}{\alpha\rho^2} & \frac{E+p}{\rho} - \frac{n^2}{\alpha\rho^2} & \frac{n}{\rho}\left(1 + \frac{1}{\alpha}\right) & g_4 & -\frac{n}{\alpha\rho} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.19}$$

where $g_1 = -\frac{n^2}{\rho^2} + \frac{1}{\alpha}\left(\frac{m^2+n^2}{2\rho^2}\right)$, $g_2 = -\frac{1}{\alpha^2}\left(E - \frac{m^2+n^2}{2\rho} - \beta\right)$,
$g_3 = -\frac{(E+p)n}{\rho^2} + \frac{n^3+nm^2}{2\alpha\rho^3}$, $g_4 = -\frac{n}{\alpha^2\rho}\left(E - \frac{m^2+n^2}{2\rho} - \beta\right)$.

$$\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{4\mu}{3}\left(\frac{\partial_x m}{\rho^2} - \frac{2m\partial_x \rho}{\rho^3}\right) & -\frac{4\mu}{3}\frac{\partial_x \rho}{\rho^2} & 0 & 0 & 0 & 0 \\ -\mu\left(\frac{\partial_x n}{\rho^2} - \frac{2n\partial_x \rho}{\rho^3}\right) & 0 & -\mu\frac{\partial_x \rho}{\rho^2} & 0 & 0 & 0 \\ v_1 & v_2 & v_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.20}$$

where $v_1 = -\frac{4\mu}{3}\left(\frac{2m\partial_x m}{\rho^3} - \frac{3m^2\partial_x \rho}{\rho^4}\right) - \mu\left(\frac{2n\partial_x n}{\rho^3} - \frac{3n^2\partial_x \rho}{\rho^4}\right)$, $v_2 = \frac{4\mu}{3}\left(\frac{\partial_x m}{\rho^2} - \frac{2m\partial_x \rho}{\rho^3}\right)$, $v_3 = \mu\left(\frac{\partial_x n}{\rho^2} - \frac{2n\partial_x \rho}{\rho^3}\right)$.

$$\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}_x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{4\mu}{3}\frac{m}{\rho^2} & \frac{4\mu}{3\rho} & 0 & 0 & 0 & 0 \\ -\frac{\mu n}{\rho^2} & 0 & \frac{\mu}{\rho} & 0 & 0 & 0 \\ -\frac{4\mu}{3}\frac{m^2}{\rho^3} - \frac{\mu n^2}{\rho^3} & \frac{4\mu}{3}\frac{m}{\rho^2} & \frac{\mu n}{\rho^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.21}$$

$$\frac{\partial \mathbf{W}_2}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\mu\left(\frac{\partial_y m}{\rho^2} - \frac{2m\partial_y \rho}{\rho^3}\right) & -\mu\frac{\partial_y \rho}{\rho^2} & 0 & 0 & 0 & 0 \\ -\frac{4\mu}{3}\left(\frac{\partial_y n}{\rho^2} - \frac{2n\partial_y \rho}{\rho^3}\right) & 0 & -\frac{4\mu}{3}\frac{\partial_y \rho}{\rho^2} & 0 & 0 & 0 \\ w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.22}$$

where $w_1 = -\frac{4\mu}{3}\left(\frac{2n\partial_y n}{\rho^3} - \frac{3n^2\partial_y \rho}{\rho^4}\right) - \mu\left(\frac{2m\partial_y m}{\rho^3} - \frac{3m^2\partial_y \rho}{\rho^4}\right)$, $w_2 = \mu\left(\frac{\partial_y m}{\rho^2} - \frac{2m\partial_y \rho}{\rho^3}\right)$,

$w_3 = \frac{4\mu}{3}\left(\frac{\partial_y n}{\rho^2} - \frac{2n\partial_y \rho}{\rho^3}\right).$

$$\frac{\partial \mathbf{W}_2}{\partial \mathbf{U}_x} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\mu\frac{m}{\rho^2} & \frac{\mu}{\rho} & 0 & 0 & 0 & 0 \\ -\frac{4\mu}{3}\frac{n}{\rho^2} & 0 & \frac{4\mu}{3}\frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{4\mu}{3}\frac{n^2}{\rho^3} - \frac{\mu m^2}{\rho^3} & \mu\frac{m}{\rho^2} & \frac{4\mu}{3}\frac{n}{\rho^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.23}$$

Appendix B. Compressible Navier-Stokes equations for dry atmosphere in primitive variables.

The dry dynamics of Earth's atmosphere can be modeled by the compressible Navier-Stokes equations written in the conservative form in terms of density ρ $[kg/m^3]$, velocity $\mathbf{u}$ $[m/s, m/s, m/s]$ – Cartesian or $[m/s, 1/s, 1/s]$ – spherical, and the total energy per unit volume E $[J/m^3]$ (see [81]):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{B.24}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p + 2\rho\left(\boldsymbol{\omega} \times \mathbf{u}\right) + \rho \mathbf{g} - \nabla \cdot \hat{\boldsymbol{\sigma}} = 0 \tag{B.25}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot ((E+p)\mathbf{u}) - \nabla \cdot \left(\frac{\mu c_p}{Pr} \nabla T + \mathbf{u} \cdot \hat{\boldsymbol{\sigma}}\right) = 0 \tag{B.26}$$

where $\boldsymbol{\omega}$ $[1/s, 1/s, 1/s]$ is the rotational velocity of the Earth, $\hat{\boldsymbol{\sigma}}$ is the viscous stress tensor given by

$$\hat{\boldsymbol{\sigma}} = \mu \left[\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \frac{2}{3} (\nabla \cdot \mathbf{u}) \hat{\mathbf{I}} \right],$$

$\mathbf{g}$ $[m/s^2, m/s^2, m/s^2]$ – Cartesian or $[m/s^2, 1/s^2, 1/s^2]$ – spherical, is the sum of the true gravity and the centrifugal force, c_p $[J/(K \cdot kg)]$, c_v $[J/(K \cdot kg)]$, μ $[kg/(s \cdot m)]$, Pr, $\gamma = \dfrac{c_p}{c_V}$, π_∞ $[Pa]$ are constant for each material. The total energy E is the sum of the internal energy ($e = c_V T + \dfrac{\pi_\infty}{\rho}$), kinetic energy, and gravitational potential energy:

$$E = \rho e + \frac{1}{2}\rho \mathbf{u} \cdot \mathbf{u} \cdot \mathbf{u} + \rho g r \tag{B.27}$$

where r $[m]$ is the radial distance from the center of the Earth. The viscous stress tensor for a Newtonian fluid is given by

$$\hat{\boldsymbol{\sigma}} = \mu \left[\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \frac{2}{3}(\nabla \cdot \mathbf{u})\hat{\mathbf{I}} \right]. \tag{B.28}$$

Pressure is given through the Stiffened Gas Equation of State:

$$p = (\gamma - 1)\rho e - \gamma \pi_\infty. \tag{B.29}$$

The goal of this appendix is to re-write equations (B.24)-(B.26) in the non-conservative form in terms of the primitive variables p $[Pa]$, $\mathbf{u}$ $[m/s, m/s, m/s]$ – Cartesian or $[m/s, 1/s, 1/s]$ – spherical, and T $[K]$. Then, density will be given by the following equation of state (equivalent to B.29):

$$\rho = \frac{p + \pi_\infty}{c_V(\gamma - 1)T}. \tag{B.30}$$

First, we work with the momentum conservation equation (B.25). Using the chain rule:

$$\frac{\partial \rho \mathbf{u}}{\partial t} = \rho \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\frac{\partial \rho}{\partial t},$$

$$\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla \cdot (\rho \mathbf{u})\mathbf{u}.$$

Hence, (B.25) becomes

$$\rho\left[\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{\rho}\nabla p-\frac{1}{\rho}\nabla\cdot\hat{\boldsymbol{\sigma}}+\mathbf{g}+2(\mathbf{u}\times\boldsymbol{\omega})\right]+\mathbf{u}\left[\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{u})\right]=0$$

Since the expression in the second brackets is zero due to the mass conservation (B.24), the momentum conservation can be re-written in the non-conservative form as:

$$\frac{\partial\mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{\rho}\nabla p-\frac{1}{\rho}\nabla\cdot\hat{\boldsymbol{\sigma}}+\mathbf{g}+2(\mathbf{u}\times\boldsymbol{\omega})=0. \tag{B.31}$$

Next, we look at the energy conservation equation (B.26). Fo convinience, we denote

$$Q=-\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla T+\mathbf{u}\cdot\hat{\boldsymbol{\sigma}}\right),$$

and then (B.26) can be re-written as

$$\frac{\partial E}{\partial t}+\nabla\cdot((E+p)\mathbf{u})+Q=0.$$

We can express the total energy as:

$$E=\rho c_V T+\pi_\infty+\frac{\rho\mathbf{u}\cdot\mathbf{u}}{2}+\rho gr=\frac{p+\pi_\infty}{\gamma-1}+\pi_\infty+\frac{\rho\mathbf{u}\cdot\mathbf{u}}{2}+\rho gr=$$
$$\frac{p}{\gamma-1}+\frac{\gamma\pi_\infty}{\gamma-1}+\frac{\rho\mathbf{u}\cdot\mathbf{u}}{2}+\rho gr.$$

Substituting the last expression for energy into (B.26) gives:

$$\left[\partial_t\left(\frac{p}{\gamma-1}\right)+\nabla\cdot\left(\frac{p}{\gamma-1}\mathbf{u}\right)+\nabla\cdot(p\mathbf{u})\right]+\left[\partial_t\left(\frac{\pi_\infty\gamma}{\gamma-1}\right)+\nabla\cdot\left(\frac{\pi_\infty\gamma}{\gamma-1}\mathbf{u}\right)\right]+$$
$$\left[\partial_t\left(\frac{\rho\mathbf{u}\cdot\mathbf{u}}{2}\right)+\nabla\cdot\left(\frac{\rho\mathbf{u}\cdot\mathbf{u}}{2}\mathbf{u}\right)\right]+\left[\partial_t(\rho gr)+\nabla\cdot(\rho gr\mathbf{u})\right]+Q=0$$

Considering all the expressions in brackets above one by one, and using (B.24), one obtains

$$\partial_t(\rho g r) + \nabla \cdot (\rho g r \mathbf{u}) = g(r\partial_t \rho + r\rho\nabla \cdot \mathbf{u} + r\nabla\rho \cdot \mathbf{u} + \rho\nabla r \cdot \mathbf{u}) =$$

$$g(r(\partial_t \rho + \nabla \cdot (\rho\mathbf{u})) + \rho\nabla r \cdot \mathbf{u}) = gr\nabla r \cdot \mathbf{u} = g\rho u_r,$$

$$\frac{1}{2}(\partial_t(\rho\mathbf{u}\cdot\mathbf{u}) + \nabla \cdot ((\rho\mathbf{u}\cdot\mathbf{u})\mathbf{u})) = \frac{1}{2}\left((\mathbf{u}\cdot\mathbf{u})\partial_t\rho + \rho\partial_t(\mathbf{u}\cdot\mathbf{u}) + (\rho\mathbf{u}\cdot\mathbf{u})\nabla\cdot\mathbf{u} + \mathbf{u}\cdot\nabla(\rho\mathbf{u}\cdot\mathbf{u})\right) =$$

$$\frac{\mathbf{u}\cdot\mathbf{u}}{2}\left[\partial_t\rho + \rho\nabla\cdot\mathbf{u} + \mathbf{u}\cdot\nabla\rho\right] + \frac{\rho}{2}\left[\partial_t(\mathbf{u}\cdot\mathbf{u}) + \mathbf{u}\cdot\nabla(\mathbf{u}\cdot\mathbf{u})\right] = \rho\mathbf{u}\cdot\left[\partial_t\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right].$$

Then, using (B.31):

$$\frac{1}{2}(\partial_t(\rho\mathbf{u}\cdot\mathbf{u}) + \nabla \cdot ((\rho\mathbf{u}\cdot\mathbf{u})\mathbf{u})) = \rho\mathbf{u}\cdot\left[-\frac{1}{\rho}\nabla p - \mathbf{g} - 2\boldsymbol{\omega}\times\mathbf{u} + \frac{1}{\rho}\nabla\cdot\hat{\boldsymbol{\sigma}}\right] =$$

$$-\mathbf{u}\cdot\nabla p - \rho\mathbf{u}\cdot\mathbf{g} - 2\rho\mathbf{u}\cdot(\boldsymbol{\omega}\times\mathbf{u}) + \mathbf{u}\cdot(\nabla\cdot\hat{\boldsymbol{\sigma}}).$$

Since $\mathbf{u}\cdot\mathbf{g} = gu_r$ and $\mathbf{u}$ is perpendicular to $\boldsymbol{\omega}\times\mathbf{u}$, (B.26) becomes:

$$\begin{gathered}\partial_t\left(\frac{p}{\gamma-1}\right) + \nabla\cdot\left(\frac{p}{\gamma-1}\mathbf{u}\right) + \nabla\cdot(p\mathbf{u}) + \partial_t\left(\frac{\pi_\infty\gamma}{\gamma-1}\right) + \\ \nabla\cdot\left(\frac{\pi_\infty\gamma}{\gamma-1}\mathbf{u}\right) - \mathbf{u}\cdot\nabla p + \mathbf{u}\cdot(\nabla\cdot\hat{\boldsymbol{\sigma}}) + Q = 0\end{gathered} \tag{B.32}$$

Let $V = \mathbf{u}\cdot(\nabla\cdot\hat{\boldsymbol{\sigma}}) + Q$. Applying the chain rule to the terms in (B.32) and re-arranging them we get:

$$\frac{1}{\gamma-1}\left[\frac{\partial p}{\partial t} + p\nabla\cdot\mathbf{u} + (\gamma-1)p\nabla\cdot\mathbf{u} + \mathbf{u}\cdot\nabla p + \pi_\infty\gamma\nabla\cdot\mathbf{u}\right] + V +$$

$$p\left[\partial_t\left(\frac{1}{\gamma-1}\right) + \mathbf{u}\cdot\nabla\left(\frac{1}{\gamma-1}\right)\right] + \left[\partial_t\left(\frac{\pi_\infty\gamma}{\gamma-1}\right) + \mathbf{u}\cdot\nabla\left(\frac{\pi_\infty\gamma}{\gamma-1}\right)\right] = 0.$$

Since equations

$$\partial_t \left(\frac{1}{\gamma - 1}\right) + \mathbf{u} \cdot \nabla \left(\frac{1}{\gamma - 1}\right) = 0$$

and

$$\partial_t \left(\frac{\pi_\infty \gamma}{\gamma - 1}\right) + \mathbf{u} \cdot \nabla \left(\frac{\pi_\infty \gamma}{\gamma - 1}\right) = 0$$

represent the advection of a material interface and thus have to be satisfied, (B.26) can be written as:

$$\frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p + \gamma(p + \pi_\infty)\nabla \cdot \mathbf{u} + (\gamma - 1)V = 0. \tag{B.33}$$

Finally, we re-write (B.24) as:

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

Since

$$\rho = \frac{p}{c_V(\gamma - 1)T} + \frac{\pi_\infty}{c_V(\gamma - 1)T},$$

we compute:

$$\begin{aligned}\frac{\partial \rho}{\partial t} =& \frac{1}{c_V(\gamma - 1)T}\frac{\partial p}{\partial t} - \frac{p}{c_v(\gamma - 1)T^2}\frac{\partial T}{\partial t} + \frac{p}{T}\partial_t \left(\frac{1}{c_V(\gamma - 1)}\right) \\ &+ \frac{1}{T}\partial_t \left(\frac{\pi_\infty}{c_V(\gamma - 1)}\right) - \frac{\pi_\infty}{c_V(\gamma - 1)T^2}\frac{\partial T}{\partial t},\end{aligned}$$

$$\begin{aligned}\nabla \rho =& \frac{1}{c_V(\gamma - 1)T}\nabla p - \frac{p}{c_v(\gamma - 1)T^2}\nabla T + \frac{p}{T}\nabla \left(\frac{1}{c_V(\gamma - 1)}\right) \\ &+ \frac{1}{T}\nabla \left(\frac{\pi_\infty}{c_V(\gamma - 1)}\right) - \frac{\pi_\infty}{c_V(\gamma - 1)T^2}\nabla T,\end{aligned}$$

and the mass conservation equation (B.24) becomes:

$$\frac{1}{c_V(\gamma-1)T}\left[\frac{\partial p}{\partial t}+\mathbf{u}\cdot\nabla p\right]-\frac{p+\pi_\infty}{c_V(\gamma-1)T^2}\left[\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T\right]+\frac{p+\pi_\infty}{c_V(\gamma-1)T}\nabla\cdot\mathbf{u}+$$
$$\frac{p}{T}\left[\frac{\partial}{\partial t}\left(\frac{1}{c_V(\gamma-1)}\right)+\mathbf{u}\cdot\nabla\left(\frac{1}{c_V(\gamma-1)}\right)\right]+\frac{1}{T}\left[\frac{\partial}{\partial t}\left(\frac{\pi_\infty}{c_V(\gamma-1)}\right)+\mathbf{u}\cdot\nabla\left(\frac{\pi_\infty}{c_V(\gamma-1)}\right)\right]=0$$

Since

$$\frac{\partial}{\partial t}\left(\frac{1}{c_V(\gamma-1)}\right)+\mathbf{u}\cdot\nabla\left(\frac{1}{c_V(\gamma-1)}\right)=0$$

and

$$\frac{\partial}{\partial t}\left(\frac{\pi_\infty}{c_V(\gamma-1)}\right)+\mathbf{u}\cdot\nabla\left(\frac{\pi_\infty}{c_V(\gamma-1)}\right)=0$$

represent the advection of a material interface and thus have to be satisfied, and $\frac{\partial p}{\partial t}+\mathbf{u}\cdot\nabla p$ can be expressed from (B.33), the mass conservation can be written in the non-conservative form as:

$$\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T+(\gamma-1)T\nabla\cdot\mathbf{u}+\frac{(\gamma-1)T}{p+\pi_\infty}V=0. \tag{B.34}$$

Recall that

$$V=-\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla T\right)-\nabla\cdot(\mathbf{u}\cdot\hat{\boldsymbol{\sigma}})+\mathbf{u}\cdot(\nabla\cdot\hat{\boldsymbol{\sigma}}).$$

Using the symmetry of the stress tensor we obtain:

$$\nabla\cdot(\mathbf{u}\cdot\hat{\boldsymbol{\sigma}})-\mathbf{u}\cdot(\nabla\cdot\hat{\boldsymbol{\sigma}})=\partial_i(u_j\sigma_{ij})-u_j\partial_i\sigma_{ji}=(\partial_iu_j)\sigma_{ij}+u_j\partial_i\sigma_{ij}-u_j\partial_i\sigma_{ji}=$$
$$(\partial_iu_j)\sigma_{ij}+u_j\partial_i\sigma ij-u_j\partial_i\sigma_{ij}=(\partial_iu_j)\sigma_{ij}=\nabla\mathbf{u}:\hat{\boldsymbol{\sigma}},$$

and thus V becomes

$$V = -\nabla \cdot \left(\frac{\mu c_p}{Pr} \nabla T\right) - \nabla \mathbf{u} : \hat{\boldsymbol{\sigma}}. \tag{B.35}$$

Substituting (B.35) into (B.33) and (B.34), we finally write the system (B.24)-(B.26) in the non-conservative form in terms of the primitive variables $(p, \mathbf{u}, T)$:

$$\begin{aligned} \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T + (\gamma - 1) T \nabla \cdot \mathbf{u} - \frac{(\gamma-1)T}{p + \pi_\infty} \nabla \cdot \left(\frac{\mu c_p}{Pr} \nabla T\right) & \\ - \frac{(\gamma-1)T}{p + \pi_\infty} \nabla \mathbf{u} : \hat{\boldsymbol{\sigma}} = 0, & \end{aligned} \tag{B.36}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p - \frac{1}{\rho} \nabla \cdot \hat{\boldsymbol{\sigma}} + \mathbf{g} + 2(\mathbf{u} \times \boldsymbol{\omega}) = 0, \tag{B.37}$$

$$\begin{aligned} \frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p + \gamma(p + \pi_\infty) \nabla \cdot \mathbf{u} - (\gamma - 1) \nabla \cdot \left(\frac{\mu c_p}{Pr} \nabla T\right) & \\ - (\gamma - 1) \nabla \mathbf{u} : \hat{\boldsymbol{\sigma}} = 0, & \end{aligned} \tag{B.38}$$

where

$$\rho = \frac{p + \pi_\infty}{c_V (\gamma - 1) T}. \tag{B.39}$$

Appendix C. Governing Equations and definition of operators for Chapter 3.

In Chapter 3 we are concerned with solving the following system of PDEs in spherical coordinates:

$$\begin{aligned}
\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T + (\gamma-1)T\nabla\cdot\mathbf{u} - \frac{(\gamma-1)T}{p+\pi_\infty}\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla T\right)& \qquad (\text{C.40})\\
- \frac{(\gamma-1)T}{p+\pi_\infty}\nabla\mathbf{u} : \hat{\boldsymbol{\sigma}} = 0,&\\
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \frac{1}{\rho}\nabla p - \frac{1}{\rho}\nabla\cdot\hat{\boldsymbol{\sigma}} + \mathbf{g} + 2(\mathbf{u}\times\boldsymbol{\omega}) = 0,& \qquad (\text{C.41})\\
\frac{\partial p}{\partial t} + \mathbf{u}\cdot\nabla p + \gamma(p+\pi_\infty)\nabla\cdot\mathbf{u} - (\gamma-1)\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla T\right)& \qquad (\text{C.42})\\
- (\gamma-1)\nabla\mathbf{u} : \hat{\boldsymbol{\sigma}} = 0.&
\end{aligned}$$

where

$$\rho = \frac{p+\pi_\infty}{c_V(\gamma-1)T}. \qquad (\text{C.43})$$

First, we provide expressions for all the spherical differential operators used in (C.40)-(C.42). Recall that the spherical transformation is given by:

$$\begin{cases} x &= r\sin\theta\cos\phi\\ y &= r\sin\theta\sin\phi\\ z &= r\cos\theta, \end{cases}$$

where $r \in [R_1, R_2]$, $\theta \in [0, \pi]$, $\phi \in [0, 2\pi]$. We also denote by $\boldsymbol{e_r}$, $\boldsymbol{e_\theta}$, and $\boldsymbol{e_\phi}$ the corresponding unit vectors in spherical coordinates, and the spherical unit tensors by $\boldsymbol{e_{zl}}$, where z, $l = r$, θ, or ϕ. Then,

$$
\begin{aligned}
\nabla \cdot \mathbf{u} =& \frac{1}{r^2}\frac{\partial\left(r^2 u_r\right)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(u_\theta \sin\theta)}{\partial\theta} + \frac{1}{r\sin\theta}\frac{\partial(u_\phi)}{\partial\phi} \\
\nabla f =& \frac{\partial f}{\partial r}\boldsymbol{e_r} + \frac{1}{r}\frac{\partial f}{\partial\theta}\boldsymbol{e_\theta} + \frac{1}{r\sin\theta}\frac{\partial f}{\partial\phi}\boldsymbol{e_\phi} \\
\nabla^2 f =& \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial f}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 f}{\partial\phi^2} \\
\nabla\mathbf{u} =& \hat{\mathbf{G}} = \frac{\partial u_r}{\partial r}\boldsymbol{e_{rr}} + \frac{\partial u_\theta}{\partial r}\boldsymbol{e_{r\theta}} + \frac{\partial u_\phi}{\partial r}\boldsymbol{e_{r\phi}} + \\
& \left(\frac{1}{r}\frac{\partial u_r}{\partial\theta} - \frac{u_\theta}{r}\right)\boldsymbol{e_{\theta r}} + \left(\frac{1}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{u_r}{r}\right)\boldsymbol{e_{\theta\theta}} + \left(\frac{1}{r}\frac{\partial u_\phi}{\partial\theta}\right)\boldsymbol{e_{\theta\phi}} + \\
& \left(\frac{1}{r\sin\theta}\frac{\partial u_r}{\partial\phi} - \frac{u_\phi}{r}\right)\boldsymbol{e_{\phi r}} + \left(\frac{1}{r\sin\theta}\frac{\partial u_\theta}{\partial\phi} - \frac{u_\phi}{r\tan\theta}\right)\boldsymbol{e_{\phi\theta}} + \\
& \left(\frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi} + \frac{u_\theta}{r\tan\theta} + \frac{u_r}{r}\right)\boldsymbol{e_{\phi\phi}} \\
\nabla^2\mathbf{u} =& \left(\nabla^2 u_r - \frac{2u_r}{r^2} - \frac{2}{r^2\sin\theta}\frac{\partial(u_\theta\sin\theta)}{\partial\theta} - \frac{2}{r^2\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right)\boldsymbol{e_r} + \\
& \left(\nabla^2 u_\theta - \frac{u_\theta}{r^2\sin^2\theta} + \frac{2}{r^2}\frac{\partial u_r}{\partial\theta} - \frac{2\cos\theta}{r^2\sin^2\theta}\frac{\partial u_\phi}{\partial\phi}\right)\boldsymbol{e_\theta} + \\
& \left((\nabla^2 u_\phi - \frac{u_\phi}{r^2\sin^2\theta} + \frac{2}{r^2\sin\theta}\frac{\partial u_r}{\partial\phi} + \frac{2\cos\theta}{r^2\sin^2\theta}\frac{\partial u_\theta}{\partial\phi}\right)\boldsymbol{e_\phi} \\
\nabla\cdot(\mu\nabla\mathbf{u}) =& \left(\nabla\cdot(\mu\nabla u_r) - \frac{2\mu u_r}{r^2}\right)\boldsymbol{e_r} + \\
& \left(-\frac{\mu}{r^2\sin\theta}\frac{\partial(u_\theta\sin\theta)}{\partial\theta} - \frac{\mu}{r^2\sin\theta}\frac{\partial u_\phi}{\partial\phi} - \frac{1}{r^2\sin\theta}\frac{\partial(\mu u_\theta\sin\theta)}{\partial\theta} - \frac{1}{r^2\sin\theta}\frac{\partial(\mu u_\phi)}{\partial\phi}\right)\boldsymbol{e_r} + \\
& \left(\nabla\cdot(\mu\nabla u_\theta) - \frac{\mu u_\theta}{r^2\sin^2\theta}\right)\boldsymbol{e_\theta} + \\
& \left(\frac{\mu}{r^2}\frac{\partial u_r}{\partial\theta} - \frac{\mu\cos\theta}{r^2\sin^2\theta}\frac{\partial u_\phi}{\partial\phi} + \frac{1}{r^2}\frac{\partial(\mu u_r)}{\partial\theta} - \frac{\cos\theta}{r^2\sin^2\theta}\frac{\partial(\mu u_\phi)}{\partial\phi}\right)\boldsymbol{e_\theta} + \\
& \left(\nabla\cdot(\mu\nabla u_\phi) - \frac{\mu u_\phi}{r^2\sin^2\theta}\right)\boldsymbol{e_\phi} + \\
& \left(\frac{\mu}{r^2\sin\theta}\frac{\partial u_r}{\partial\phi} + \frac{\mu\cos\theta}{r^2\sin^2\theta}\frac{\partial u_\theta}{\partial\phi} + \frac{1}{r^2\sin\theta}\frac{\partial(\mu u_r)}{\partial\phi} + \frac{\cos\theta}{r^2\sin^2\theta}\frac{\partial(\mu u_\theta)}{\partial\phi}\right)\boldsymbol{e_\phi}
\end{aligned}
$$

$$
\begin{aligned}
\nabla\nabla\cdot\mathbf{u} = &\left(\frac{\partial}{\partial r}\left[\frac{1}{r^2}\frac{\partial\left(r^2u_r\right)}{\partial r}\right]+\frac{\partial}{\partial r}\left[\frac{1}{r\sin\theta}\frac{\partial\left(u_\theta\sin\theta\right)}{\partial\theta}\right]+\frac{\partial}{\partial r}\left[\frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right]\right)\boldsymbol{e_r}+\\
&\left(\frac{1}{r^3}\frac{\partial^2\left(r^2u_r\right)}{\partial\theta\partial r}+\frac{1}{r^2}\frac{\partial}{\partial\theta}\left[\frac{1}{\sin\theta}\frac{\partial\left(u_\theta\sin\theta\right)}{\partial\theta}\right]+\frac{1}{r^2}\frac{\partial}{\partial\theta}\left[\frac{1}{\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right]\right)\boldsymbol{e_\theta}+\\
&\left(\frac{1}{r^3\sin\theta}\frac{\partial^2\left(r^2u_r\right)}{\partial\phi\partial r}+\frac{1}{r^2\sin^2\theta}\frac{\partial^2\left(u_\theta\sin\theta\right)}{\partial\phi\partial\theta}+\frac{1}{r^2\sin^2\theta}\frac{\partial^2u_\phi}{\partial\phi^2}\right)\boldsymbol{e_\phi}
\end{aligned}
$$

Notice that

$$
\nabla\cdot\hat{\boldsymbol{\sigma}} = \nabla\cdot\left(\mu\nabla\mathbf{u}\right)+\nabla\left(\frac{\mu}{3}\nabla\cdot\mathbf{u}\right).
$$

And finally, the stress tensor is equal to

$$
\begin{aligned}
\hat{\boldsymbol{\sigma}} =&\mu\left[2\frac{\partial u_r}{\partial r}-\frac{2}{3}\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2u_r\right)+\frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta u_\theta\right)+\frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right)\right]\boldsymbol{e_{rr}}\\
&+\mu\left[\frac{\partial u_\theta}{\partial r}+\frac{1}{r}\frac{\partial u_r}{\partial\theta}-\frac{u_\theta}{r}\right]\boldsymbol{e_{r\theta}}\\
&+\mu\left[\frac{\partial u_\phi}{\partial r}+\frac{1}{r\sin\theta}\frac{\partial u_r}{\partial\phi}-\frac{u_\phi}{r}\right]\boldsymbol{e_{r\phi}}\\
&+\mu\left[\frac{\partial u_\theta}{\partial r}+\frac{1}{r}\frac{\partial u_r}{\partial\theta}-\frac{u_\theta}{r}\right]\boldsymbol{e_{\theta r}}\\
&+\mu\left[\frac{2}{r}\frac{\partial u_\theta}{\partial\theta}+\frac{2u_r}{r}-\frac{2}{3}\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2u_r\right)+\frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta u_\theta\right)+\frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right)\right]\boldsymbol{e_{\theta\theta}}\\
&+\mu\left[\frac{1}{r}\frac{\partial u_\phi}{\partial\theta}+\frac{1}{r\sin\theta}\frac{\partial u_\theta}{\partial\phi}-\frac{u_\phi}{r\tan\theta}\right]\boldsymbol{e_{\theta\phi}}\\
&+\mu\left[\frac{\partial u_\phi}{\partial r}+\frac{1}{r\sin\theta}\frac{\partial u_r}{\partial\phi}-\frac{u_\phi}{r}\right]\boldsymbol{e_{\phi r}}\\
&+\mu\left[\frac{1}{r}\frac{\partial u_\phi}{\partial\theta}+\frac{1}{r\sin\theta}\frac{\partial u_\theta}{\partial\phi}-\frac{u_\phi}{r\tan\theta}\right]\boldsymbol{e_{\phi\theta}}\\
&+\mu\left[\frac{2}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}+\frac{2u_\theta}{r\tan\theta}+\frac{2u_r}{r}-\frac{2}{3}\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2u_r\right)+\frac{1}{r\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta u_\theta\right)+\frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial\phi}\right)\right]\boldsymbol{e_{\phi\phi}}
\end{aligned}
$$

To simplify the notations, we define the following operators:

$$
\boldsymbol{A_1}(\mathbf{u})f = \mathbf{u}\cdot\nabla f \qquad \text{(C.44)}
$$

$$\boldsymbol{A_2}(p,\mathbf{u})\mathbf{v} = \gamma\left(p+\pi_\infty\right)\nabla\cdot\mathbf{v} - \left(\gamma-1\right)\nabla\mathbf{u} : \hat{\boldsymbol{\sigma}}\left(\mathbf{v}\right) \tag{C.45}$$

$$\boldsymbol{A_3}f = -\left(\gamma-1\right)\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla f\right) \tag{C.46}$$

$$\boldsymbol{B_1}(\rho)f = \frac{1}{\rho}\nabla f \tag{C.47}$$

$$\boldsymbol{B_2}(\rho,\mathbf{u})\mathbf{v} = \mathbf{u}\cdot\nabla\mathbf{v} + 2\left(\boldsymbol{\omega}\times\mathbf{v}\right) - \frac{1}{\rho}\left(\nabla\cdot\left(\mu\nabla\mathbf{v}\right) + \nabla\left(\frac{\mu}{3}\nabla\cdot\mathbf{v}\right)\right) \tag{C.48}$$

$$\boldsymbol{C_2}(T,p,\mathbf{u})\mathbf{v} = (\gamma-1)T\nabla\cdot\mathbf{v} - \frac{\left(\gamma-1\right)T}{p+\pi_\infty}\nabla\mathbf{u} : \hat{\boldsymbol{\sigma}}\left(\mathbf{v}\right) \tag{C.49}$$

$$\boldsymbol{C_3}(T,p,\mathbf{u})f = \mathbf{u}\cdot\nabla f - \frac{\left(\gamma-1\right)T}{p+\pi_\infty}\nabla\cdot\left(\frac{\mu c_p}{Pr}\nabla f\right) \tag{C.50}$$

Thus, the system (C.40)-(C.42) can be written as

$$\frac{\partial p}{\partial t} + \boldsymbol{A_1}(\mathbf{u})p + \boldsymbol{A_2}(p,\mathbf{u})\mathbf{u} + \boldsymbol{A_3}T = 0, \tag{C.51}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{B_1}(\rho)p + \boldsymbol{B_2}(\rho,\mathbf{u})\mathbf{u} + \mathbf{g} = 0, \tag{C.52}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{C_2}(T,p,\mathbf{u})\mathbf{u} + \boldsymbol{C_3}(T,p,\mathbf{u})T = 0. \tag{C.53}$$

The operators (C.44) - (C.50) can be splitted direction-wise as following (note that the operators with the upper subindex M include mixed derivatives, derivatives in staggered directions, and other terms that cannot be naturally incorporated into the direction splitting approach implicitly):

$$\boldsymbol{A_1}f = \boldsymbol{A_1^r}f + \boldsymbol{A_1^\theta}f + \boldsymbol{A_1^\phi}f = u_r\frac{\partial f}{\partial r} + \frac{u_\theta}{r}\frac{\partial f}{\partial\theta} + \frac{u_\phi}{r\sin\theta}\frac{\partial f}{\partial\phi}$$

$$\boldsymbol{A_2}\mathbf{v} = \boldsymbol{A_2^r}\mathbf{v} + \boldsymbol{A_2^\theta}\mathbf{v} + \boldsymbol{A_2^\phi}\mathbf{v} + \boldsymbol{A_2^M}\mathbf{v}$$

$$
\begin{aligned}
\boldsymbol{A_2^r}\mathbf{v} =& \left(\gamma(p+\pi_\infty) + \frac{2\mu(\gamma-1)}{3}\left(\hat{G}_{rr}+\hat{G}_{\theta\theta}+\hat{G}_{\phi\phi}\right)\right)\frac{1}{r^2}\frac{\partial\left(r^2 v_r\right)}{\partial r} - \\
&2\mu(\gamma-1)\hat{G}_{rr}\frac{\partial v_r}{\partial r} - \mu(\gamma-1)\left(2\hat{G}_{\theta\theta}+2\hat{G}_{\phi\phi}\right)\frac{v_r}{r} \\
\boldsymbol{A_2^\theta}\mathbf{v} =& \left(\gamma(p+\pi_\infty) + \frac{2\mu(\gamma-1)}{3}\left(\hat{G}_{rr}+\hat{G}_{\theta\theta}+\hat{G}_{\phi\phi}\right)\right)\frac{1}{r\sin\theta}\frac{\partial\left(\sin\theta v_\theta\right)}{\partial \theta} - \\
&2\mu(\gamma-1)\frac{\hat{G}_{\theta\theta}}{r}\frac{\partial v_\theta}{\partial \theta} - \mu(\gamma-1)\left(\frac{2\hat{G}_{\phi\phi}}{\tan\theta} - \hat{G}_{r\theta} - \hat{G}_{\theta r}\right)\frac{v_\theta}{r} \\
\boldsymbol{A_2^\phi}\mathbf{v} =& \left(\gamma(p+\pi_\infty) + \frac{2\mu(\gamma-1)}{3}\left(\hat{G}_{rr}+\hat{G}_{\theta\theta}-2\hat{G}_{\phi\phi}\right)\right)\frac{1}{r\sin\theta}\frac{\partial v_\phi}{\partial \phi} - \\
&\mu(\gamma-1)\left(\frac{2\hat{G}_{\phi\phi}}{\tan\theta} - \frac{\hat{G}_{\phi\theta}+\hat{G}_{\theta\phi}}{\tan\theta} - \hat{G}_{r\phi} - \hat{G}_{\phi r}\right)\frac{v_\phi}{r} \\
\boldsymbol{A_2^M}\mathbf{v} =& \left[\boldsymbol{A^M_{2,(r)}}, \boldsymbol{A^M_{2,(\theta)}}, \boldsymbol{A^M_{2,(\phi)}}\right]\mathbf{v} = -\mu(\gamma-1)\left(\hat{G}_{r\theta}+\hat{G}_{\theta r}\right)\frac{\partial v_\theta}{\partial r} - \mu(\gamma-1)\left(\hat{G}_{r\phi}+\hat{G}_{\phi r}\right)\frac{\partial v_\phi}{\partial r} \\
&-\mu(\gamma-1)\left(\hat{G}_{r\theta}+\hat{G}_{\theta r}\right)\frac{1}{r}\frac{\partial v_r}{\partial \theta} - \mu(\gamma-1)\left(\hat{G}_{\theta\phi}+\hat{G}_{\phi\theta}\right)\frac{1}{r}\frac{\partial v_\phi}{\partial \theta} \\
&-\mu(\gamma-1)\left(\hat{G}_{r\phi}+\hat{G}_{\phi r}\right)\frac{1}{r\sin\theta}\frac{\partial v_r}{\partial \phi} - \mu(\gamma-1)\left(\hat{G}_{\phi\theta}+\hat{G}_{\theta\phi}\right)\frac{1}{r\sin\theta}\frac{\partial v_\theta}{\partial \phi}
\end{aligned}
$$

Let $\kappa = \frac{\mu c_p}{Pr}$. Then:

$$
\begin{aligned}
\boldsymbol{A_3}f =& \boldsymbol{A_3^r}f + \boldsymbol{A_3^\theta}f + \boldsymbol{A_3^\phi}f = -\frac{\gamma-1}{r^2}\frac{\partial}{\partial r}\left(\kappa r^2\frac{\partial f}{\partial r}\right) - \\
&\frac{\gamma-1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\kappa\sin\theta\frac{\partial f}{\partial \theta}\right) - \frac{\gamma-1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\kappa\frac{\partial f}{\partial \phi}\right) \\
\boldsymbol{B_1}f =& \boldsymbol{e_r}\left(\boldsymbol{B_1^r}f\right) + \boldsymbol{e_\theta}\left(\boldsymbol{B_1^\theta}f\right) + \boldsymbol{e_\phi}\left(\boldsymbol{B_1^\phi}f\right) = \boldsymbol{e_r}\left(\frac{1}{\rho}\frac{\partial(f)}{\partial r}\right) + \boldsymbol{e_\theta}\left(\frac{1}{\rho r}\frac{\partial f}{\partial \theta}\right) + \boldsymbol{e_\phi}\left(\frac{1}{\rho r\sin\theta}\frac{\partial f}{\partial \phi}\right)
\end{aligned}
$$

Then,

$$
\begin{aligned}
\boldsymbol{B_2}\mathbf{v} = \boldsymbol{e_r}&\left(\boldsymbol{B_2^{r,r}}\mathbf{v} + \boldsymbol{B_2^{\theta,r}}\mathbf{v} + \boldsymbol{B_2^{\phi,r}}\mathbf{v} + \boldsymbol{B_2^{M,r}}\mathbf{v}\right) + \\
\boldsymbol{e_\theta}&\left(\boldsymbol{B_2^{r,\theta}}\mathbf{v} + \boldsymbol{B_2^{\theta,\theta}}\mathbf{v} + \boldsymbol{B_2^{\phi,\theta}}\mathbf{v} + \boldsymbol{B_2^{M,\theta}}\mathbf{v}+\right) + \\
\boldsymbol{e_\phi}&\left(\boldsymbol{B_2^{r,\phi}}\mathbf{v} + \boldsymbol{B_2^{\theta,\phi}}\mathbf{v} + \boldsymbol{B_2^{\phi,\phi}}\mathbf{v} + \boldsymbol{B_2^{M,\phi}}\mathbf{v}\right)
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{B_2^{r,r}}\mathbf{v} =& u_r\frac{\partial v_r}{\partial r} - \frac{1}{\rho}\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(\mu r^2\frac{\partial v_r}{\partial r}\right)\right] - \frac{1}{3\rho}\left(\frac{\partial}{\partial r}\left[\frac{\mu}{r^2}\frac{\partial}{\partial r}\left(r^2 v_r\right)\right]\right)\\
\boldsymbol{B_2^{\theta,r}}\mathbf{v} =& \frac{u_\theta}{r}\frac{\partial v_r}{\partial \theta} - \frac{1}{\rho}\left[\frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\mu\sin\theta\frac{\partial v_r}{\partial\theta}\right)\right]\\
\boldsymbol{B_2^{\phi,r}}\mathbf{v} =& \frac{u_\phi}{r\sin\theta}\frac{\partial v_r}{\partial\phi} - \frac{1}{\rho}\left[\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\mu\frac{\partial v_r}{\partial\phi}\right)\right]\\
\boldsymbol{B_2^{M,r}}\mathbf{v} =& \left[\boldsymbol{B_{2,(r)}^{M,r}}, \boldsymbol{B_{2,(\theta)}^{M,r}}, \boldsymbol{B_{2,(\phi)}^{M,r}}\right]\mathbf{v} = -\frac{u_\theta v_\theta}{r} - \frac{u_\phi v_\phi}{r} + \frac{2\mu v_r}{\rho r^2} - 2\omega\sin\theta v_\phi\\
& - \frac{1}{3\rho}\left(\frac{\partial}{\partial r}\left[\frac{\mu}{r\sin\theta}\frac{\partial}{\partial\theta}\left(v_\theta\sin\theta\right)\right]\right) - \frac{1}{3\rho}\left(\frac{\partial}{\partial r}\left[\frac{\mu}{r\sin\theta}\frac{\partial v_\phi}{\partial\phi}\right]\right)\\
& - \frac{\mu}{\rho}\left[-\frac{2}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta v_\theta\right) - \frac{2}{r^2\sin\theta}\frac{\partial v_\phi}{\partial\phi}\right] -\\
& - \frac{1}{\rho}\left[-\frac{2}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\mu\sin\theta v_\theta\right) - \frac{2}{r^2\sin\theta}\frac{\partial(\mu v_\phi)}{\partial\phi}\right]\\
\boldsymbol{B_2^{r,\theta}}\mathbf{v} =& u_r\frac{\partial v_\theta}{\partial r} - \frac{1}{\rho}\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(\mu r^2\frac{\partial v_\theta}{\partial r}\right)\right]\\
\boldsymbol{B_2^{\theta,\theta}}\mathbf{v} =& \frac{u_\theta}{r}\frac{\partial v_\theta}{\partial\theta} - \frac{1}{\rho}\left[\frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\mu\sin\theta\frac{\partial v_\theta}{\partial\theta}\right)\right] -\\
& \frac{1}{3\rho}\left(\frac{1}{r^2}\frac{\partial}{\partial\theta}\left[\frac{\mu}{\sin\theta}\frac{\partial}{\partial\theta}\left(v_\theta\sin\theta\right)\right]\right)\\
\boldsymbol{B_2^{\phi,\theta}}\mathbf{v} =& \frac{u_\phi}{r\sin\theta}\frac{\partial v_\theta}{\partial\phi} - \frac{1}{\rho}\left[\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\mu\frac{\partial v_\theta}{\partial\phi}\right)\right]\\
\boldsymbol{B_2^{M,\theta}}\mathbf{v} =& \left[\boldsymbol{B_{2,(r)}^{M,\theta}}, \boldsymbol{B_{2,(\theta)}^{M,\theta}}, \boldsymbol{B_{2,(\phi)}^{M,\theta}}\right]\mathbf{v} = \frac{u_\theta v_r}{r} - \frac{u_\phi v_\phi}{r\tan\theta} + \frac{\mu v_\theta}{\rho r^2\sin^2\theta} - 2\omega\cos\theta v_\phi\\
& - \frac{1}{3\rho}\left(\frac{1}{r^3}\frac{\partial}{\partial\theta}\left(\mu\frac{\partial(r^2 v_r)}{\partial r}\right)\right) - \frac{1}{3\rho}\left(\frac{1}{r^2}\frac{\partial}{\partial\theta}\left(\frac{\mu}{\sin\theta}\frac{\partial v_\phi}{\partial\phi}\right)\right)\\
& - \frac{\mu}{\rho}\left[\frac{1}{r^2}\frac{\partial u_r}{\partial\theta} - \frac{\cos\theta}{r^2\sin^2\theta}\frac{\partial u_\phi}{\partial\phi}\right] - \frac{1}{\rho}\left[-\frac{1}{r^2}\frac{\partial(\mu u_r)}{\partial\theta} - \frac{\cos\theta}{r^2\sin^2\theta}\frac{\partial(\mu u_\phi)}{\partial\phi}\right]\\
\boldsymbol{B_2^{r,\phi}}\mathbf{v} =& u_r\frac{\partial v_\phi}{\partial r} - \frac{1}{\rho}\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(\mu r^2\frac{\partial v_\phi}{\partial r}\right)\right]\\
\boldsymbol{B_2^{\theta,\phi}}\mathbf{u} =& \frac{u_\theta}{r}\frac{\partial v_\phi}{\partial\theta} - \frac{1}{\rho}\left[\frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\mu\sin\theta\frac{\partial v_\phi}{\partial\theta}\right)\right]\\
\boldsymbol{B_2^{\phi,\phi}}\mathbf{v} =& \frac{u_\phi}{r\sin\theta}\frac{\partial v_\phi}{\partial\phi} - \frac{1}{\rho}\left[\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\mu\frac{\partial v_\phi}{\partial\phi}\right)\right] - \frac{1}{3\rho}\left(\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\mu\frac{\partial v_\phi}{\partial\phi}\right)\right)\\
\boldsymbol{B_2^{M,\phi}}\mathbf{v} =& \left[\boldsymbol{B_{2,(r)}^{M,\phi}}, \boldsymbol{B_{2,(\theta)}^{M,\phi}}, \boldsymbol{B_{2,(\phi)}^{M,\phi}}\right]\mathbf{v} = \frac{u_\phi v_\theta}{r\tan\theta} + \frac{u_\phi v_r}{r} + \frac{\mu v_\phi}{\rho r^2\sin^2\theta} + 2\omega\cos\theta v_\theta + 2\omega\sin\theta v_r
\end{aligned}
$$

$$
\begin{aligned}
&-\frac{1}{3\rho}\left(\frac{1}{r^3\sin\theta}\frac{\partial}{\partial\phi}\left(\mu\frac{\partial(r^2v_r)}{\partial r}\right)\right)-\frac{1}{3\rho}\left(\frac{1}{r^2\sin^2\theta}\frac{\partial}{\partial\phi}\left(\mu\frac{\partial(v_\theta\sin\theta)}{\partial\theta}\right)\right)\\
&-\frac{\mu}{\rho}\left[\frac{1}{r^2\sin\theta}\frac{\partial u_r}{\partial\phi}+\frac{\cos\theta}{r^2\sin^2\theta}\frac{\partial u_\theta}{\partial\phi}\right]-\frac{1}{\rho}\left[\frac{1}{r^2\sin\theta}\frac{\partial(\mu u_r)}{\partial\phi}+\frac{\cos\theta}{r^2\sin^2\theta}\frac{\partial(\mu u_\theta)}{\partial\phi}\right]
\end{aligned}
$$

$$
\boldsymbol{C_2}\mathbf{v}=\boldsymbol{C_2^r}\mathbf{v}+\boldsymbol{C_2^\theta}\mathbf{v}+\boldsymbol{C_2^\phi}\mathbf{v}+\boldsymbol{C_2^C}\mathbf{v}
$$

$$
\begin{aligned}
\boldsymbol{C_2^r}\mathbf{v}=&\left((\gamma-1)T+\frac{2\mu(\gamma-1)T}{3(p+\pi_\infty)}\left(\hat{G}_{rr}+\hat{G}_{\theta\theta}+\hat{G}_{\phi\phi}\right)\right)\frac{1}{r^2}\frac{\partial\left(r^2v_r\right)}{\partial r}-\\
&2\mu\frac{(\gamma-1)T}{p+\pi_\infty}\hat{G}_{rr}\frac{\partial v_r}{\partial r}-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(2\hat{G}_{\theta\theta}+2\hat{G}_{\phi\phi}\right)\frac{v_r}{r}\\
\boldsymbol{C_2^\theta}\mathbf{v}=&\left((\gamma-1)T+\frac{2\mu(\gamma-1)T}{3(p+\pi_\infty)}\left(\hat{G}_{rr}+\hat{G}_{\theta\theta}+\hat{G}_{\phi\phi}\right)\right)\frac{1}{r\sin\theta}\frac{\partial\left(\sin\theta v_\theta\right)}{\partial\theta}\\
&-2\mu\frac{(\gamma-1)T}{p+\pi_\infty}\frac{\hat{G}_{\theta\theta}}{r}\frac{\partial v_\theta}{\partial\theta}-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\frac{2\hat{G}_{\phi\phi}}{\tan\theta}-\hat{G}_{r\theta}-\hat{G}_{\theta r}\right)\frac{v_\theta}{r}\\
\boldsymbol{C_2^\phi}\mathbf{v}=&\left((\gamma-1)T+\frac{2\mu(\gamma-1)T}{3(p+\pi_\infty)}\left(\hat{G}_{rr}+\hat{G}_{\theta\theta}-2\hat{G}_{\phi\phi}\right)\right)\frac{1}{r\sin\theta}\frac{\partial v_\phi}{\partial\phi}-\\
&\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\frac{2\hat{G}_{\phi\phi}}{\tan\theta}-\frac{\hat{G}_{\phi\theta}+\hat{G}_{\theta\phi}}{\tan\theta}-\hat{G}_{r\phi}-\hat{G}_{\phi r}\right)\frac{v_\phi}{r}\\
\boldsymbol{C_2^M}\mathbf{v}=&\left[\boldsymbol{C^M_{2,(r)}},\boldsymbol{C^M_{2,(\theta)}},\boldsymbol{C^M_{2,(\phi)}}\right]\mathbf{v}=\\
&-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\hat{G}_{r\theta}+\hat{G}_{\theta r}\right)\frac{\partial v_\theta}{\partial r}-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\hat{G}_{r\phi}+\hat{G}_{\phi r}\right)\frac{\partial v_\phi}{\partial r}-\\
&-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\hat{G}_{r\theta}+\hat{G}_{\theta r}\right)\frac{1}{r}\frac{\partial v_r}{\partial\theta}-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\hat{G}_{\theta\phi}+\hat{G}_{\phi\theta}\right)\frac{1}{r}\frac{\partial v_\phi}{\partial\theta}\\
&-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\hat{G}_{r\phi}+\hat{G}_{\phi r}\right)\frac{1}{r\sin\theta}\frac{\partial v_r}{\partial\phi}-\mu\frac{(\gamma-1)T}{p+\pi_\infty}\left(\hat{G}_{\phi\theta}+\hat{G}_{\theta\phi}\right)\frac{1}{r\sin\theta}\frac{\partial v_\theta}{\partial\phi}
\end{aligned}
$$

$$
\boldsymbol{C_3}f=\boldsymbol{C_3^r}f+\boldsymbol{C_3^\theta}f+\boldsymbol{C_3^\phi}f
$$

$$
\begin{aligned}
\boldsymbol{C_3^r} f &= u_r \frac{\partial f}{\partial r} - \frac{(\gamma-1)T}{(p+\pi_\infty)r^2} \frac{\partial}{\partial r}\left(\kappa r^2 \frac{\partial f}{\partial r}\right) \\
\boldsymbol{C_3^\theta} f &= \frac{u_\theta}{r} \frac{\partial f}{\partial \theta} - \frac{(\gamma-1)T}{(p+\pi_\infty)r^2 \sin\theta} \frac{\partial}{\partial \theta}\left(\kappa \sin\theta \frac{\partial f}{\partial \theta}\right) \\
\boldsymbol{C_3^\phi} f &= \frac{u_\phi}{r\sin\theta} \frac{\partial f}{\partial \phi} - \frac{(\gamma-1)T}{(p+\pi_\infty)r^2 \sin^2\theta} \frac{\partial}{\partial \phi}\left(\kappa \frac{\partial f}{\partial \phi}\right).
\end{aligned}
$$

If we denote by U the vector of unknowns:

$$U = [p, u_r, u_\theta, u_\phi, T]^T,$$

define Gr as the gravity vector

$$Gr = [0, g, 0, 0, 0]^T,$$

and combine operators in corresponding directions by introducing the following block-operators:

$$\boldsymbol{D_r}(U) = \begin{bmatrix} \boldsymbol{A_1^r} & \boldsymbol{A_2^r} & 0 & 0 & \boldsymbol{A_3^r} \\ \boldsymbol{B_1^r} & \boldsymbol{B_2^{r,r}} & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{B_2^{r,\theta}} & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{B_2^{r,\phi}} & 0 \\ 0 & \boldsymbol{C_2^r} & 0 & 0 & \boldsymbol{C_3^r} \end{bmatrix} \tag{C.54}$$

$$\boldsymbol{D_\theta}(U) = \begin{bmatrix} \boldsymbol{A_1^\theta} & 0 & \boldsymbol{A_2^\theta} & 0 & \boldsymbol{A_3^\theta} \\ 0 & \boldsymbol{B_2^{\theta,r}} & 0 & 0 & 0 \\ \boldsymbol{B_1^\theta} & 0 & \boldsymbol{B_2^{\theta,\theta}} & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{B_2^{\theta,\phi}} & 0 \\ 0 & 0 & \boldsymbol{C_2^\theta} & 0 & \boldsymbol{C_3^\theta} \end{bmatrix} \tag{C.55}$$

$$\boldsymbol{D_\phi}(U) = \begin{bmatrix} \boldsymbol{A_1^\phi} & 0 & 0 & \boldsymbol{A_2^\phi} & \boldsymbol{A_3^\phi} \\ 0 & \boldsymbol{B_2^{\phi,r}} & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{B_2^{\phi,\theta}} & 0 & 0 \\ \boldsymbol{B_1^\phi} & 0 & 0 & \boldsymbol{B_2^{\phi,\phi}} & 0 \\ 0 & 0 & 0 & \boldsymbol{C_2^\phi} & \boldsymbol{C_3^\phi} \end{bmatrix} \tag{C.56}$$

$$\boldsymbol{D_M}(U) = \begin{bmatrix} 0 & \boldsymbol{A_{2,(r)}^M} & \boldsymbol{A_{2,(\theta)}^M} & \boldsymbol{A_{2,(\phi)}^M} & 0 \\ 0 & \boldsymbol{B_{2,(r)}^{M,r}} & \boldsymbol{B_{2,(\theta)}^{M,r}} & \boldsymbol{B_{2,(\phi)}^{M,r}} & 0 \\ 0 & \boldsymbol{B_2^{M,2,(r)}} & \boldsymbol{B_{2,(\theta)}^{M,r}} & \boldsymbol{B_{2,(\phi)}^{M,r}} & 0 \\ 0 & \boldsymbol{B_{2,(r)}^{M,r}} & \boldsymbol{B_{2,(\theta)}^{M,r}} & \boldsymbol{B_{2,(\phi)}^{M,r}} & 0 \\ 0 & \boldsymbol{C_{2,(r)}^M} & \boldsymbol{C_{2,(\theta)}^M} & \boldsymbol{C_{2,(\phi)}^M} & 0 \end{bmatrix} \tag{C.57}$$

the system (C.51)-(C.53) can be written in a compact form as:

$$\frac{\partial U}{\partial t} + \boldsymbol{D_r}(U)U + \boldsymbol{D_\theta}(U)U + \boldsymbol{D_\phi}(U)U + \boldsymbol{D_M}(U)U + Gr = 0. \tag{C.58}$$