Title
An improved all-speed projection algorithm for low Mach number flows

Author
Chaplin, Christopher Michael

Publication Date
2018

Peer reviewed|Thesis/dissertation

An improved all-speed projection algorithm for low Mach number flows

by

Christopher M. Chaplin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Mechanical Engineering

and the Designated Emphasis

in

Computational and Data Science and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Phillip Colella, Co-chair
Professor Panayiotis Papadopoulos, Co-chair
Professor Per-Olof Persson
Professor Shawn Shadden

Summer 2018

Abstract

An improved all-speed projection algorithm for low Mach number flows

by

Christopher M. Chaplin

Doctor of Philosophy in Engineering – Mechanical Engineering

Designated Emphasis in Computational and Data Science and Engineering

University of California, Berkeley

Professor Phillip Colella, Co-chair

Professor Panayiotis Papadopoulos, Co-chair

The objective of this work is to design and implement a high fidelity method for simulating low Mach number compressible flows ranging from inviscid gas dynamics to compressible Navier-Stokes with reactions. In order to achieve this objective, we introduce three goals. The first is that Δt only be constrained by the advective CFL condition. The seconds goal is that we attain accurate treatment of compressible effects such as bulk compression and expansion due to thermo-chemical processes and long wavelength acoustics. In particular, we want to capture well-known low Mach number limiting behavior. The third and final goal is that we want to use easy solvers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Low Mach number flows are found both in natural settings and in engineering applications. Incompressible hydrodynamics, anelastic hydrodynamics, stellar hydrodynamics, combustion, and plasma physics all have low Mach number cases [3], [119].

1.1 Inviscid Gas Dynamics

The simplest flow equations that reveal the challenge of low Mach number (M) flows are the Euler equations for inviscid gas dynamics:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{1.1}$$

$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + p\mathbb{I}) = 0 \tag{1.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot [\rho (E + p)] = 0 \tag{1.3}$$

$$E = \rho \left(e + \frac{1}{2} ||\boldsymbol{v}||^2 \right) \tag{1.4}$$

$$p = \rho R T. \tag{1.5}$$

These equations are composed of the conservation of mass (Equation (1.1)), momentum (Equation (1.2)), and total energy (Equation (1.3)). The last two equations define the total energy (Equation (1.4)) in terms of the internal and kinetic energies, and the equation of state for an ideal gas (Equation (1.5)). For the discussion of low Mach number flows, it is easier to examine one of the non-conservative forms of the governing equations. Transforming Equations (1.1) to (1.3) to $(p, \boldsymbol{v}, T)$ variables yields,

$$\frac{Dp}{Dt} + \gamma p \left(\nabla \cdot \boldsymbol{v}\right) = 0 \tag{1.6}$$

$$\rho \frac{D\boldsymbol{v}}{Dt} + \nabla p = 0 \tag{1.7}$$

$$\rho \frac{DT}{Dt} - \left(\frac{\gamma - 1}{\gamma}\right) \frac{Dp}{Dt} = 0, \tag{1.8}$$

where the notation $\frac{D()}{Dt}$ is the material derivative $\frac{D()}{Dt} = \frac{\partial()}{\partial t} + (\boldsymbol{v} \cdot \nabla)()$.

1.2 Low M Asymptotics

For a flow to be low Mach number, the magnitude of bulk fluid velocity ($||\boldsymbol{v}||$) must be much less than the acoustic wave propagation speed (c). The ratio of these two speeds defines the Mach number,

$$M = \frac{||\boldsymbol{v}||}{c}. \tag{1.9}$$

In practice flows are considered low Mach number when $M < 0.1$. Typically the fluid velocity is resolved at advective time scales throughout a simulation. The acoustic speed depends on the physics of the problem.

Equations (1.6) to (1.8) can be non-dimensionalized to compare the relative importance of each term. There are two different analyses that are worth exploring to expose the underlying nature of this system.

If we start by setting length $l = l^u$, where superscript u denotes the advective scale, the non-dimensionalization is given by

$$l^u = \frac{l}{l^*} \ , \ v^u = \frac{v}{v^*} \ , \ \rho^u = \frac{\rho}{\rho^*} \ , \ p^u = \frac{p}{p^*} \ , \ T^u = \frac{T}{T^*} \ , \ M^u = \frac{v^u}{\sqrt{\gamma \frac{p^u}{\rho^u}}},$$

where the superscript $*$ denotes non-dimensional values.

If we apply this non-dimensionalization to Equations (1.6) to (1.8) and assume the following asymptotic expansion in M for each variable:

$$p = p_0(x,t) + M p_1(x,t) + \gamma M^2 p_2(x,t) \tag{1.10}$$

$$v = v_0(x,t) + M v_1(x,t) + O(M^2) \tag{1.11}$$

$$T = T_0(x,t) + M T_1(x,t) + O(M^2), \tag{1.12}$$

we produce the following system of equations

$$\frac{Dp}{Dt} + \gamma p \left(\nabla \cdot \boldsymbol{v}\right) = 0 \tag{1.13}$$

$$\rho \frac{D\boldsymbol{v}}{Dt} + (\gamma M^2)^{-1} \nabla p = 0 \tag{1.14}$$

$$\rho \frac{DT}{Dt} - \left(\frac{\gamma - 1}{\gamma}\right) \frac{Dp}{Dt} = 0, \tag{1.15}$$

where the superscript $*$ notation has been dropped.

The important outcome is that pressure gradient $\nabla p = \mathcal{O}(M^{-2})$ in this analysis. We'll label this the low M asymptotic result.

If instead the characteristic length is chose to be the size of the entire domain $L \sim ct^u$, we find that $\nabla p \sim \mathcal{O}(M^2)$ [62], [95]. This can be verified from equations of dynamics along with the equation of state. Let us denote this as the long wavelength acoustic result.

We seek a method that can recover both of the limiting cases described above: low M asymptotics and long wavelength acoustics. In the low M asymptotic regime, the acoustic waves are weak and contribute little to the dynamics. The stiffness caused by these waves constrains Δt unnecessarily for a fully explicit method. The explicit time step is given by

$$\sigma \frac{\Delta t}{\Delta x} = \frac{1}{\max(|\ ||\boldsymbol{v}|| \pm c\ |)}, \tag{1.16}$$

where σ is the Courant-Friedrichs-Lewy (CFL) number.

At low Mach numbers, the time step for the explicit methods is completely defined by the rate of acoustic wave propagation c. On the other hand, fully implicit methods can relax rapidly on time scales based on c. However fully implicit methods introduce complicated non-linear solves. This motivates the use of Implicit-Explicit (IMEX) schemes that can use a Δt based on the advective time scale, but can also account for long wavelength acoustics.

1.3 Thesis Outline

The objective of this work is to design and implement a high fidelity method for simulating low Mach number compressible flows ranging from inviscid gas dynamics to compressible Navier-Stokes with reactions. In order to achieve this objective, we introduce three goals. The first is that Δt only be constrained by the advective CFL condition. The seconds goal is that we attain accurate treatment of compressible effects such as bulk compression and expansion due to thermo-chemical processes and long wavelength acoustics. In particular, we want to capture well-known low Mach number limiting behavior. The third and final goal is that we want to use easy solvers.

Our approach is to add redundant equations for the potential component of velocity and pressure. The combined system of equations is solved using an IMEX formulation, with the redundant equations solved implicitly and the discretization of the fluid equations

having a lower-triangular dependence on the redundant variables. This new approach was motivated by that taken in Colella and Pao [84]. In Colella and Pao a similar set of redundant equations are added to model inviscid gas dynamics. This work improves the algorithm in Colella and Pao in three major ways. First, we have removed the need for any variable-coefficient projection. We restructured the equations in such a way that we can leverage IMEX frameworks. Lastly, we extended the algorithm to handle viscous, thermal, and combustion effects.

The remainder of this thesis is broken into five chapters. The next chapter contains a review of the relevant literature. In Chapter 3, we describe our all-speed splitting approach. From there we move onto describing discretization methods. The final two chapters cover the results and conclusions.

Chapter 2

Numerical Methods for Low Mach Number Flows

There are a few different solution approaches available to modeling low Mach number gas dynamics. The following considerations should be taken into account for any low Mach number method:

1. Take advantage of the disparate scales between the local fluid speed, v, and acoustic speed, c.
2. Involve either a decomposition of the variables and/or a simplification of the full governing equations.
3. Reduce to the correct incompressible solution as $M \to 0$.

Recall the original system of equations:

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

$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + p\mathbb{I}) = 0 \tag{2.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot [\rho (E + p)] = 0. \tag{2.3}$$

These equations are hyperbolic and have the form,

$$\frac{\partial U}{\partial t} + \nabla \cdot (F(U)) = 0. \tag{2.4}$$

The two large classes of approaches for modeling these flows are preconditioning schemes and pressure-implicit methods.

2.1 Preconditioning Schemes

The key feature of all of these schemes is they use an artificially chosen sound speed. All of these methods are in some way related to Chorin's method of artificial compressibility [17]. In artificial compressibility, the incompressible equations are modified so that there is an evolution equation for an artificial scalar, density in the original case, in place of the usual velocity divergence constraint. The resulting auxiliary equations are then solved together and the solution is advanced in time.

Simple/PISO

SIMPLE/PISO are preconditioning methods quite similar to artificial compressibility. However, they are important enough to merit their own discussion [10], [11], [14], [19], [23], [42]–[45], [77], [79], [82], [88], [89], [107], [109], [112].

The SIMPLE (Semi-Implicit Method for Pressure Linked Equations) method was developed by Spalding and Patankar in the early 1970s [11], [89]. There are three primary features of the SIMPLE method: replacement of the continuity equation with a pressure evolution equation, definition of a coupled pressure-velocity system, and implicit treatment of the coupling terms. In practice, SIMPLE looks like a standard predictor-corrector method (define an initial state and pressure, update velocity with that pressure, solve for new pressure) with an iterative wrapper around it. The density is updated once the pressure and velocity are deemed "converged".

Although the first version of this algorithm was designed for modeling steady flows, it was quickly extended to modeling transient flows. Many improvements and comparisons between different versions have been introduced over time [19], [45], [88]. The primary drawback with this method is that within a given iteration the updated velocities and fluxes do not satisfy momentum balance, so iteration is required. However, the method is easy to apply and a single iteration is fairly cheap computationally so it is still useful today, particularly for steady-state calculations.

In the 1980s, Raad Issa set out to create a refined version of SIMPLE that did not require iteration. He named his method PISO (Pressure-Implicit with Splitting of Operators) [42], [43]. PISO is very similar to SIMPLE except that instead of adjusting the density at the end of the iteration, the density is updated at each predictor-corrector "stage" so that continuity and momentum balance are discretely satisfied. Issa then was able to show that with a small number of predictor-corrector stages (2-3), the user could get the solution error to scale as $\mathcal{O}((\Delta t)^2)$.

There is both an incompressible and a compressible version of PISO. The algorithm is popular and is used in many commercial codes today as a counterpart to SIMPLE for transient flows [Fluent [5], OpenFoam [118], Converge [105]]. The primary drawback is that the method requires iteration for complicated flows, regardless of whether the complications are caused by physics or problem geometry. It is also not clear whether the the method

extends to high-order accuracy in time. There is no special consideration for transitioning flows in terms of Mach number.

As with SIMPLE, PISO has had many extensions and improvements over the years [44], [82].

Explicit Preconditioning

Returning to Equation (2.4), there are some terms in $\nabla \cdot (F(U))$ which do not scale well as $M \to 0$. A scaling matrix may be introduced such that the following is true,

$$P^{-1}\frac{\partial U}{\partial t} + \nabla \cdot (F(U)) = 0. \tag{2.5}$$

The goal here is to define P^{-1} in such a way that the system is less stiff as $M \to 0$. One could select any number of preconditioning matrices for a steady-state flow and the stiffness can be resolved. Retaining time accuracy for transient flows requires careful selection of the preconditioning matrix along with some other tricks. The typical approach is to introduce artificial compressibility and scale the resulting equations in such a way as to shrink the maximum eigenvalue of the resulting hyperbolic system.

The initial studies in preconditioning methods were done for steady-state Euler equations [15], [114], [115]. Weiss and Smith [117] formulated one of the first explicit preconditioning methods designed to simulate time-varying flows. Their method recovered accuracy in time by sub-cycling (taking smaller time-steps) until the error in the preconditioned time-evolution was below a chosen threshold. Meister [72] reviewed some of the early time-dependent methods and developed a way to extend the method into the weakly compressible regime. Wesseling [119] has shown that it is possible to extend compressible solvers to the weakly compressible flow regime via some clever transformations of the equations.

There are some newer splitting techniques that share similarities with the explicit preconditioning schemes. Haack et al. [36] proposed a newer, more advanced explicit scheme for compressible flow at any speed. In this approach, "preconditioning" is applied to both the temporal terms and spatial terms. They introduce a splitting of the equations into slow-speed and high-speed terms via a tunable parameter. The stability analysis reveals that the time step is independent of the acoustic speed, but there is a constraint on the time step that is more stringent than desirable.

Other Preconditioning Methods

There are a wide range of methods that resemble explicit-preconditioning available [22], [34], [36], [37], [46], [51], [53], [54], [57], [60], [61], [63], [72]–[74], [78], [98]–[101], [103], [108], [110], [113].

There are hybrid implicit-explicit preconditioning methods. Instead of transforming the equations in an effort to remove the need for implicitness, these methods accept that implicitness and iteration will be necessary [49], [69]. The general approach is to start with a fully

explicit method, conservative or not. A time step is taken with this explicit method and then corrected with the implicit preconditioning step. This process of taking an explicit step with an accompanying solve of an implicit system of equations is then iteratively repeated until a desired accuracy is achieved. There have been many advances in these methods over the years [8], [15], [28], [32], [48], [49], [64], [87], [124], [125]. The primary drawback of these approaches is that they are incredibly expensive in terms of the variable preconditioning and iterative solvers.

Asymptotic methods [16], [113] are another distinct family of compressible methods. These methods are derived using the low M approximation, but the acoustic effects are not completely filtered out.

Another approach was discussed by Kwatra et al [57]. In this study, the divergence of velocity term in the pressure evolution equation is approximated by the pressure gradient term from the velocity evolution equation (and it is treated implicitly). This produces the correct elliptic pressure solve in the zero Mach number limit. The method solves the conservative system of equations and produces convincing results in the high Mach number limit. One drawback from this study was the lack of enforcing the pressure equation of state. For a relatively simple example it was shown that a small number of pressure solver iterations yielded very similar results to the equation of state. Including some sort of relaxation to the pressure solve would likely improve the results. Also, no discrete error measures or convergence rates were presented for any of the test problems. The method was recently expanded to the Navier-Stokes equations [37]. This study showed that there were instabilities in extending the Kwatra method to include viscous terms, particularly at low Mach numbers and for long time integrations. The new study ended up fixing the instabilities, but treated viscous terms explicitly. Again, no convergence studies were provided.

Mach-Uniform Schemes

Mach-uniform, or all-speed schemes seek are a special class of solvers that seek to work well over the entire range of Mach numbers. The original all-speed method [38] is called ICE (Implicit Continuous-fluid Eulerian). The framework for this method is the MAC (Marker-And-Cell) [39] incompressible flow solver. In ICE, as the Mach number approaches zero, the MAC method is recovered. As the Mach number approaches infinity, an implicit version of a standard compressible Euler solver is recovered. This method works well for a subset of problems, and the principle ideas and technologies of the method continue to prove themselves effective and useful in many newer schemes today.

Mach-uniform schemes are designed to reduce to the staggered-grid scheme in the incompressible limit as the Mach number approaches zero. As the flow grows transonic, the methods are designed to exploit many of the benefits of the existing fully compressible methods. These methods are overall very successful at modeling physics over a wide range of flow speeds [22], [24], [40], [51], [70], [116], [122], [123]. Few of these studies include high-order discretizations and there is little to no error or convergence analysis.

2.2 Pressure-Implicit Methods

For these methods, either the compressibility effects are treated implicitly or they are eliminated entirely using asmyptotics. In both cases the pressure is treated implicitly.

Barely Implicit Flux-Corrected-Transport

An approach that is somewhat similar to the explicit preconditioning methods was developed by Patnaik et al [90]. The idea was to start with the explicit Flux-Corrected Transport (FCT) method and to add an implicit correction to eliminate the small time step problem at low Mach number. The implicit correction was based on ideas in [12], wherein it was identified that certain pressure and velocity terms must be treated implicitly in order to remove the acoustic dependence on the time step. This study was confined to low-order accuracy in time and did not include and viscous or thermal effects.

Zero Mach Number Projection

The zero Mach number formulation traces its origins to work done in the 1970s and 1980s [65], [97], [111]. The start of the formulation is to non-dimensionalize the relevant system of equations and look at the low M asymptotics.

The pressure expansion in Equation (1.10) and the non-dimensional equations reveal that the pressure can be split into a time-varying "background" pressure and a space-time-varying pressure that scales as M^2. Another important outcome is that a tidy expression for the evolution of this pressure can be found:

$$\frac{Dp_0(t)}{Dt} = -\gamma p_0(\nabla \cdot \boldsymbol{v}_0). \tag{2.6}$$

This expression was originally derived in [65], but it included additional terms relating to combustion. Another way to look at Equation (2.6) is that the velocity constraint ($\boldsymbol{v}$) can be evaluated as a function of known parameters. For this simple inviscid gas dynamics case, Equation (2.6) implies that the only cause of expansion in the fluid is the change in global pressure over time. In combustion, the expansion is also affected by chemical heat release and heat conduction.

Efficient zero Mach number methods have been proposed and developed over time that leverage this analysis [2], [6], [7], [25]–[27], [29], [33], [47], [52], [55], [56], [58], [59], [65], [76], [80], [81], [85], [86], [91]–[94], [96], [97], [102], [104], [111], [126]. These methods allow the user to eliminate acoustic wave propagation entirely, via the aforementioned assumptions and pressure splitting, while still retaining compressibility effects from kinetics and thermal terms.

The seminal computational work by Majda and Sethian [65] was adapted by Lai to apply a projection formulation of the equations [58], [59]. In the projection method, the state variables are advanced initially in a predictor step during which time the velocity constraint

is not imposed. Then the projection enforces the velocity constraint and the state variables are corrected accordingly. This approach was first combined with Adaptive Mesh Refinement (AMR) in [92], and then more complicated chemistry was included in [25]. Since then, this formulation has been extended to higher-order accuracy with the use of spectrally-deferred time-integration and higher-order spatial discretizations [47], [91].

The method is efficient and accurate for flows that globally remain low Mach number and for which the complete domain is too small to support long-wavelength acoustics. For flows for which the Mach number is not small, this method will not work well since the acoustic wave effects have been removed.

Projection Methods

The projection method was first developed to solve the incompressible Navier-Stokes equations by Chorin in the 1960s [18]. Projection refers to the procedure in which a predicted velocity field, which may not satisfy the divergence constraint, is projected onto the space of divergence-free fields.

These methods have been expanded and added to greatly over time [1], [4], [9], [18], [31], [35], [58], [59], [66], [67], [75], [84], [106], [121]. Projection methods are used in many of the most advanced incompressible flow solvers and they work well for methods that are adaptive in space and time.

As mentioned earlier, Colella and Pao created a new projection method for low speed flows of varying Mach number [84]. This method was designed to be able to handle compressibility effects in inviscid gas dynamics. In [31], this projection method was expanded to work on gravitationally stratified flows. However, in neither study were viscous or thermal effects considered, high-order spatial discretizations were not considered, and adaptivity was not included.

Chapter 3

All-Speed Splitting Methods

We'll start by returning to the equations of inviscid gas dynamics. The equations are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{3.1}$$

$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v} + p\mathbb{I}) = 0 \tag{3.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot [\rho (E + p)] = 0. \tag{3.3}$$

To fully specify the system, boundary and initial conditions must be given. The domain of interest is a closed container Ω with fixed boundaries $\partial\Omega$ such that the normal velocity is zero,

$$\boldsymbol{v} \cdot \underline{\underline{\boldsymbol{n}}} = 0 \text{ on } \partial\Omega. \tag{3.4}$$

Initial conditions are required for the dependent variables

$$\rho|_{t=0} = \rho_0(x) \tag{3.5}$$

$$\boldsymbol{v}|_{t=0} = \boldsymbol{v}_0(x) \tag{3.6}$$

$$E|_{t=0} = E_0(x). \tag{3.7}$$

We'll look at the standard hyperbolic analysis of this system. Converting the conserved variables $U = (\rho, \rho\boldsymbol{v}, \rho E)$ into the primitive variables $W = (\rho, \boldsymbol{v}, p)$ yields,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{3.8}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\, \boldsymbol{v} + \frac{1}{\rho} \nabla p = 0 \tag{3.9}$$

$$\frac{\partial p}{\partial t} + (\boldsymbol{v} \cdot \nabla) p + \gamma p (\nabla \cdot \boldsymbol{v}) = 0. \tag{3.10}$$

We can express this in matrix form as

$$\frac{\partial W}{\partial t} + A_W \nabla W = 0. \tag{3.11}$$

If we substitute the following expression for the speed of sound ($c^2 = \frac{\gamma p}{\rho}$), the eigenvalues of this system are

$$\lambda = \left[(\boldsymbol{v} \cdot \underline{\underline{\boldsymbol{n}}}) \ , \ (\boldsymbol{v} \cdot \underline{\underline{\boldsymbol{n}}}) \pm c \right] D. \tag{3.12}$$

3.1 Splitting

We seek an IMEX splitting of the governing equations such that we can block-diagonalize the system into stiff and non-stiff terms with non-stiff coupling. If the splitting is successful, we can treat the high-speed terms implicitly and the low-speed terms explicitly. Expressing this as a system yields,

$$\frac{\partial U}{\partial t} = F\left(U\right) \tag{3.13}$$

$$F(U) = F^E\left(U\right) + F^I\left(U\right), \tag{3.14}$$

where U is the collection of evolved variables, $F^E\left(U\right)$ is the explicit operator acting on the non-stiff terms, and $F^I\left(U\right)$ is the implicit operator acting on the stiff terms.

Hodge Decomposition

Applying the Hodge decomposition to the governing equations helps us to block-diagonalize the system. We apply the decomposition using the projection method for smooth vector fields. This decomposition involves splitting the vector field into divergence-free and curl-free components. The variable-coefficient version is presented first,

$$\boldsymbol{w} = \boldsymbol{w}_d + \frac{1}{\rho} \nabla \phi. \tag{3.15}$$

For this particular splitting Equation (3.15), the following must hold for the decomposition to be orthogonal,

$$\int_\Omega \boldsymbol{w}_d \cdot \frac{1}{\rho} \nabla \phi \rho d\Omega = 0. \tag{3.16}$$

The curl-free vector component has the following properties:

$$\Delta_\rho \phi = \nabla \cdot \boldsymbol{w} \ \text{ in } \Omega \tag{3.17}$$

$$\frac{1}{\rho} \frac{\partial \phi}{\partial n} = 0 \ \text{ in } d\Omega, \tag{3.18}$$

where the variable density Laplacian operator is given by

$$\Delta_\rho() = \nabla \cdot \frac{1}{\rho}\nabla(). \tag{3.19}$$

The divergence-free component is described by

$$\nabla \cdot \boldsymbol{w}_d = 0 \ \text{ in } \Omega \tag{3.20}$$
$$\boldsymbol{w}_d \cdot \boldsymbol{n} = 0 \ \text{ in } d\Omega. \tag{3.21}$$

Other boundary conditions could be considered, such as a moving domain, and that would introduce a harmonic velocity to the splitting (see [31] for an example of a decomposition with a harmonic velocity).

From Equation (3.15), the two variable-coefficient (in ρ) projection operators can be defined,

$$\mathbb{Q}_\rho = \frac{1}{\rho}\nabla\left(\Delta_\rho^{-1}\right)\nabla\cdot \tag{3.22}$$
$$\mathbb{P}_\rho = \mathbb{I} - \mathbb{Q}_\rho. \tag{3.23}$$

The more familiar constant-coefficient Hodge decomposition is defined by

$$\boldsymbol{v} = \boldsymbol{v}_d + \nabla\psi, \tag{3.24}$$

where orthogonality is guaranteed if the following holds,

$$\int_\Omega \boldsymbol{v}_d \cdot \nabla\psi d\Omega = 0. \tag{3.25}$$

In Equation (3.24), the divergence-free component has the same properties as the variable-coefficient counterpart. The curl-free component satisfies the following conditions,

$$\Delta\psi = \nabla \cdot \boldsymbol{v} \ \text{ in } \Omega \tag{3.26}$$
$$\frac{\partial\psi}{\partial n} = 0 \ \text{ in } d\Omega, \tag{3.27}$$

where the standard Laplacian is given by

$$\Delta() = \nabla \cdot \nabla(). \tag{3.28}$$

The constant-coefficient projection operators are now given

$$\mathbb{Q}_0 = \nabla\left(\Delta^{-1}\right)\nabla\cdot \tag{3.29}$$
$$\mathbb{P}_0 = \mathbb{I} - \mathbb{Q}_0. \tag{3.30}$$

Approaches

Recall our three goals in this work. The first is that Δt only be constrained by the advective Courant-Friedrichs-Lewy (CFL) condition. The seconds goal is that we attain accurate treatment of compressible effects such as bulk compression and expansion due to thermo-chemical processes and long wavelength acoustics. In particular, we want to capture well-known low Mach number limiting behavior. The third and final goal is that we want to use easy solvers.

Our approach is to add redundant equations for the potential component of velocity and pressure. The combined system of equations is solved using an Implicit-Explicit (IMEX) formulation, with the redundant equations solved implicitly and the discretization of the fluid equations having a lower-triangular dependence on the redundant variables.

This method is an extension of the work done by Colella and Pao [84]. Accordingly that method will be presented first.

Colella and Pao Splitting

As recognized in [84], it is possible to split the equations of inviscid gas dynamics into stiff and non-stiff terms by using the Hodge decomposition. The projection operator is the vehicle that allows us to perform this decomposition.

The first step is to perform the constant-coefficient Hodge decomposition on the velocity variable. A suitable splitting will also be applied to the pressure variables.

$$\boldsymbol{v} = \boldsymbol{v}_d + \boldsymbol{v}_p \tag{3.31}$$
$$p = \pi + \delta \tag{3.32}$$

The vectors $\boldsymbol{v}_d$ and $\boldsymbol{v}_p$ are the divergence-free and curl-free vector fields described previously. For the pressure splitting, δ is the acoustic pressure and π is an auxiliary pressure. In the limit as $M \to 0$, the acoustic pressure goes to zero and the auxiliary pressure approaches the the incompressible pressure. The split terms satisfy the following relations for a closed, fixed domain:

$$\boldsymbol{v}_d = \mathbb{P}_o(\boldsymbol{v}) \qquad \boldsymbol{v}_p = \mathbb{Q}_o(\boldsymbol{v}) \tag{3.33}$$
$$\nabla \cdot \boldsymbol{v}_d = 0 \qquad \boldsymbol{v}_p = \nabla\varphi \tag{3.34}$$
$$\boldsymbol{v}_d \cdot \boldsymbol{n} = 0 \qquad \boldsymbol{v}_p \cdot \boldsymbol{n} = 0. \tag{3.35}$$

Applying this splitting to the non-conservative equations produces the following new system:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{3.36}$$
$$\frac{\partial \delta}{\partial t} + \frac{\partial \pi}{\partial t} + (\boldsymbol{v} \cdot \nabla) p = -\gamma p (\nabla \cdot \boldsymbol{v}_p) \tag{3.37}$$

$$\frac{\partial \boldsymbol{v}_d}{\partial t} + \boldsymbol{A_d}(\boldsymbol{v}) + \frac{1}{\rho}\nabla\pi = -\mathbb{P}_0\left(\frac{1}{\rho}\nabla\delta\right) \tag{3.38}$$

$$\frac{\partial \boldsymbol{v}_p}{\partial t} + \frac{\nabla|\boldsymbol{v}_p|^2}{2} = -\mathbb{Q}_0\left(\frac{1}{\rho}\nabla\delta\right), \tag{3.39}$$

where

$$\boldsymbol{A_d}(\boldsymbol{v}) \equiv= \boldsymbol{v}\cdot\nabla\boldsymbol{v} - \nabla\frac{|\boldsymbol{v}_p|^2}{2} \tag{3.40}$$

$$\frac{1}{\rho}\nabla\pi \equiv -\mathbb{Q}_\rho\left(\boldsymbol{A_d}(\boldsymbol{v})\right). \tag{3.41}$$

All of the terms involving $\nabla\cdot\boldsymbol{v}_p$ and $\frac{1}{\rho}\nabla\delta$ are the compressible terms that must be treated implicitly.

Ideally, we could cast these new equations directly into the IMEX formulation. But there are some challenges with the above system of equations as far as the IMEX formulation is concerned. The primary issue is that we have a time derivative for π appearing in the evolution equation for δ. Equations with multiple time derivatives are differential algebraic equations, and typically have a complicated solution procedure.

For the IMEX formulation to work, we want to ensure that each equation in the system has a single time derivative. With the equations in this form, a method of lines solution can be constructed wherein an appropriate spatial discretization is applied to the system first to produce ordinary differential equations (ODE). Then a time integration scheme integrates the resulting ODEs.

In light of this, we combine the partial time derivatives of the pressure terms in the pressure evolution equation. Furthermore, we split δ into p and π contributions in the compressible velocity,

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \boldsymbol{v}_d\\ \boldsymbol{v}_p\\ p\end{bmatrix} = \begin{bmatrix}-\nabla\cdot(\rho\boldsymbol{v})\\ -\boldsymbol{A}_d\boldsymbol{v} + \mathbb{Q}_0\left[\frac{1}{\rho}\nabla\delta\right]\\ -\frac{\nabla|\boldsymbol{v}_p|^2}{2} + \mathbb{P}_0\left[\frac{1}{\rho}\nabla\delta\right] + \frac{1}{\rho}\nabla\pi\\ -\boldsymbol{v}\cdot\nabla p\end{bmatrix} + \begin{bmatrix}0\\ -\frac{1}{\rho}\nabla p\\ -\frac{1}{\rho}\nabla p\\ -\gamma p(\nabla\cdot\boldsymbol{v}_p)\end{bmatrix}. \tag{3.42}$$

Compressible Projection Splitting

Even with the new IMEX form of the inviscid equations, there are two remaining obstacles. The first obstacle is that we have two types of projection operators that need to be applied to advance the solution. The second obstacle is this splitting does not generalize easily to viscous and reacting flows.

Therefore, a new splitting is proposed that seeks to remove both of these obstacles. First, we return to the definition of $\boldsymbol{v}_p = \mathbb{Q}_0(\boldsymbol{v})$. In [84], an ansatz was made as to the form of the

evolution equation for the curl-free velocity. That evolution equation was designed to capture all of the potential flow from the total velocity. However, the $\mathbb{Q}_0$ operator is designed to do this by definition. So, an alternative version of the compressible velocity evolution could be,

$$\frac{\partial \boldsymbol{v}_p}{\partial t} = \mathbb{Q}_0 \left(\frac{\partial \boldsymbol{v}}{\partial t} \right) \tag{3.43}$$

$$= -\mathbb{Q}_0 \left((\boldsymbol{v} \cdot \nabla) \boldsymbol{v} + \frac{1}{\rho} \nabla p \right). \tag{3.44}$$

It is relatively easy to show that this expression is equivalent to the original one.

Putting all of this together, and still retaining the enthalpy equation yields:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{3.45}$$

$$\frac{\partial p}{\partial t} + (\boldsymbol{v} \cdot \nabla) p = -\gamma p (\nabla \cdot \boldsymbol{v}_p) \tag{3.46}$$

$$\frac{\partial (\rho h)}{\partial t} + \nabla \cdot (\rho h \boldsymbol{v}) = -\gamma p (\nabla \cdot \boldsymbol{v}_p) \tag{3.47}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla) \boldsymbol{v} + \frac{1}{\rho} \nabla p = 0 \tag{3.48}$$

$$\frac{\partial \boldsymbol{v}_p}{\partial t} + \mathbb{Q}_0 \left((\boldsymbol{v} \cdot \nabla) \boldsymbol{v} + \frac{1}{\rho} \nabla p \right) = 0. \tag{3.49}$$

The curl-free $\boldsymbol{v}_p$ evolution can be amended one final time by substituting $\mathbb{Q}_0 = \mathbb{I} - \mathbb{P}_0$. Looking at the pressure and $\boldsymbol{v}_p$ equations,

$$\frac{\partial p}{\partial t} + (\boldsymbol{v} \cdot \nabla) p = -\gamma p (\nabla \cdot \boldsymbol{v}_p) \tag{3.50}$$

$$\frac{\partial \boldsymbol{v}_p}{\partial t} - \mathbb{P}_0 \left((\boldsymbol{v} \cdot \nabla) \boldsymbol{v} + \frac{1}{\rho} \nabla p \right) + (\boldsymbol{v} \cdot \nabla) \boldsymbol{v} = -\frac{1}{\rho} \nabla p, \tag{3.51}$$

the implicit terms have been placed on the right hand side. Notice that we have removed the projection operation from the implicit terms. This can be put into IMEX form,

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \boldsymbol{v} \\ \boldsymbol{v}_p \\ p \\ \rho h \end{bmatrix} = \begin{bmatrix} -\nabla \cdot (\rho \boldsymbol{v}) \\ -(\boldsymbol{v} \cdot \nabla) \boldsymbol{v} \\ \mathbb{P}_0 \left[\frac{1}{\rho} \nabla p + (\boldsymbol{v} \cdot \nabla) \boldsymbol{v} \right] - (\boldsymbol{v} \cdot \nabla) \boldsymbol{v} \\ -(\boldsymbol{v} \cdot \nabla) p \\ -\nabla \cdot (\rho h \boldsymbol{v}) - \gamma p (\nabla \cdot \boldsymbol{v}_p) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{\rho} \nabla p \\ -\frac{1}{\rho} \nabla p \\ -\gamma p (\nabla \cdot \boldsymbol{v}_p) \\ 0 \end{bmatrix} \tag{3.52}$$

The variable projection has been completely removed from the formulation. How does this new evolution equation behave in the incompressible limit? Taking the divergence of Equation (3.51), we have

$$\nabla \cdot \frac{\partial \boldsymbol{v}_p}{\partial t} + \nabla \cdot \left(\left(\boldsymbol{v} \cdot \nabla \right) \boldsymbol{v} \right) = -\nabla \cdot \left(\frac{1}{\rho} \nabla p \right). \tag{3.53}$$

In the incompressible limit, $\boldsymbol{v}_p \to 0$, so the previous expression becomes

$$\nabla \cdot \left(\left(\boldsymbol{v} \cdot \nabla \right) \boldsymbol{v} \right) = -\nabla \cdot \frac{1}{\rho} \nabla p. \tag{3.54}$$

This is the correct variable-coefficient equation for pressure.

3.2 Compressible Navier-Stokes Equations

The governing equations of compressible gas dynamics with viscous and thermal effects included are known as the Navier-Stokes equations,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{3.55}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + \left(\boldsymbol{v} \cdot \nabla \right) \boldsymbol{v} + \frac{1}{\rho} \nabla p - \frac{1}{\rho} \nabla \cdot \tau = 0 \tag{3.56}$$

$$\frac{\partial p}{\partial t} + (\boldsymbol{v} \cdot \nabla) p + \gamma p (\nabla \cdot \boldsymbol{v}) - (\gamma - 1) \left[\nabla \cdot \boldsymbol{q} + \Phi \right] = 0 \tag{3.57}$$

$$\frac{\partial (\rho h)}{\partial t} + \nabla \cdot (\rho h \boldsymbol{v}) + \Psi - \nabla \cdot \boldsymbol{q} - \Phi = 0, \tag{3.58}$$

where the viscous stress, strain rate, heat flux, dissipation, and auxiliary terms are

$$\boldsymbol{\tau} = 2\eta \boldsymbol{e} + \left(\kappa - \frac{2}{3} \eta \right) (\nabla \cdot \boldsymbol{v}) \tag{3.59}$$

$$\boldsymbol{e} = \frac{1}{2} \left(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T \right) \tag{3.60}$$

$$\boldsymbol{q} = \lambda \nabla T = \frac{\lambda}{c_p} \nabla h \tag{3.61}$$

$$\Phi = 2\eta (\boldsymbol{e} : \boldsymbol{e}) + \left(\kappa - \frac{2}{3} \eta \right) (\nabla \cdot \boldsymbol{v})^2 \tag{3.62}$$

$$\Psi = -\gamma p (\nabla \cdot \boldsymbol{v}) + (\gamma - 1) \left[\nabla \cdot \boldsymbol{q} + \Phi \right]. \tag{3.63}$$

The equation of state is the same as in the inviscid case. The coefficients for thermal conduction (λ), shear viscosity (η), and volume viscosity (κ) are functions of temperature (see Appendix D).

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \boldsymbol{v}\\ \boldsymbol{v}_p\\ p\\ \rho h\end{bmatrix} = \begin{bmatrix}-\nabla\cdot(\rho\boldsymbol{v})\\ -(\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v}-\frac{1}{\rho}\nabla p\\ \mathbb{P}_0\left[\frac{1}{\rho}\nabla p+(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}-\frac{1}{\rho}\nabla\cdot\tau\right]-(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\\ -(\boldsymbol{v}\cdot\nabla)\,p+(\gamma-1)\Phi\\ -\nabla\cdot(\rho h\boldsymbol{v})+\Psi+\Phi\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{\rho}\nabla\cdot\tau\\ \frac{1}{\rho}\nabla\cdot\tau-\frac{1}{\rho}\nabla p\\ (\gamma-1)(\nabla\cdot\boldsymbol{q})-\gamma p(\nabla\cdot\boldsymbol{v})\\ \nabla\cdot\boldsymbol{q}\end{bmatrix} \tag{3.64}$$

3.3 Combustion Equations

The final system investigated is the full compressible Navier-Stokes equations with NS number of species and combustion (see Appendix B for derivation). The governing equations in non-conservative form are:

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\boldsymbol{v})=0 \tag{3.65}$$

$$\frac{\partial(\rho Y_k)}{\partial t}+\nabla\cdot(\rho Y_k\boldsymbol{v})-\nabla\cdot(\rho\mathcal{D}_k\nabla Y_k)-\dot{\omega}_k=0 \tag{3.66}$$

$$\frac{\partial\boldsymbol{v}}{\partial t}+(\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v}+\frac{1}{\rho}\nabla p-\frac{1}{\rho}(\nabla\cdot\boldsymbol{\tau})=0 \tag{3.67}$$

$$\frac{\partial p}{\partial t}+(\boldsymbol{v}\cdot\nabla)\,p-\dot{p}^E-\dot{p}^I+\alpha\sum_{k=1}^{N_s}\left[(\beta h_k+\zeta_k)\,\dot{\omega}_k\right]=0 \tag{3.68}$$

$$\frac{\partial(\rho h)}{\partial t}+\nabla\cdot(\rho h\boldsymbol{v}-q^E-q^I)+\dot{p}^E+\dot{p}^I-\alpha\sum_{k=1}^{N_s}\left[(\beta h_k+\zeta_k)\,\dot{\omega}_k\right]-\Phi=0, \tag{3.69}$$

where Y_k is the species mass fraction, D_k is the diffusion vector, h_k is the species-specific enthalpy, $\dot{\omega}_k$ is the species generation term, q^E is the explicitly treated heat flux, q^I is the implicitly treated heat flux, p^E is the explicit update to pressure, and p^I is the implicit update to pressure (refer to Appendix B).

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho Y_k\\ \boldsymbol{v}\\ \boldsymbol{v}_p\\ p\\ \rho h\end{bmatrix} = \begin{bmatrix}-\nabla\cdot(\rho\boldsymbol{v})\\ -\nabla\cdot(\rho Y_k\boldsymbol{v})\\ -(\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v}-\frac{1}{\rho}\nabla p\\ \mathbb{P}_0\left[\frac{1}{\rho}\nabla p+(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}-\frac{1}{\rho}\nabla\cdot\tau\right]-(\boldsymbol{v}\cdot\nabla)\boldsymbol{v}\\ -(\boldsymbol{v}\cdot\nabla)\,p+\dot{p}^E\\ -\nabla\cdot(\rho h\boldsymbol{v}+q^E)+\dot{p}^E+\dot{p}^I+\Phi\end{bmatrix} + \begin{bmatrix}0\\ \nabla\cdot(\rho D_k\nabla Y_k)+\dot{\omega}_k\\ \frac{1}{\rho}\nabla\cdot\tau\\ \frac{1}{\rho}\nabla\cdot\tau-\frac{1}{\rho}\nabla p\\ \dot{p}^I-\alpha\sum\limits_{k=1}^{N_s}\left[(\beta h_k+\zeta_k)\,\dot{\omega}_k\right]\\ \nabla\cdot q^I+\alpha\sum\limits_{k=1}^{N_s}\left[(\beta h_k+\zeta_k)\,\dot{\omega}_k\right]\end{bmatrix} \tag{3.70}$$

3.4 Domain and Boundary Conditions

The domain of interest is a N^D box with unit length ($\Omega = [0,1]^D$). Along the boundary of the box $\partial\Omega$, boundary conditions must be specified. Two cases will be investigated, periodic boundary conditions and solid wall boundary conditions. For the solid wall boundary conditions, no mass may leave the domain so the normal flux along $\partial\Omega$ should be equal to zero. This is accomplished by setting the bulk fluid velocity equal to the solid wall velocity ($(\boldsymbol{v}\cdot\boldsymbol{n})|_{\partial\Omega} = u_n|_{\partial\Omega} = u_{wall}$). The wall is typically stationary, such that $u_n|_{\partial\Omega} = 0$. As for the scalars, it is assumed that there is no normal derivative across $\partial\Omega$. So this implies that $\left(\frac{\partial\phi}{\partial n}\right)|_{wall} = 0$.

Tangential boundary conditions are required for the viscous velocity terms. Typical choices for velocity include slip, no-slip, or turbulent law of the wall. No-slip will be used here($u_t|_{wall} = 0$).

Periodic boundary conditions assume the values are equal at either end of domain and that the variables are periodic ($\phi|_{l=0} = \phi|_{l=1}$).

Initial conditions are also specified for all dependent variables at time zero.

Chapter 4

Discretization

4.1 Finite-Volume Preliminaries

The general approach for discretization is that of the high-order finite-volume approach described in [20], [71]. We will begin this chapter covering the fundamental definitions required to build a high-order finite-volume discretization. The preliminary definitions include averages, interpolations, and derivatives.

Computing Finite-Volume Averages

We start by defining the various required averaging operations. The first is the conversion of cell-centered quantities to cell-averaged quantities:

$$\langle \phi \rangle_{\boldsymbol{i}} = \phi_{\boldsymbol{i}} + \mathcal{O}(h^2) \tag{4.1}$$

$$= \phi_{\boldsymbol{i}} + \frac{h^2}{24} \sum_{d=1}^{D} \left(\frac{\partial^2 \phi}{\partial x_d^2} \right)_{\boldsymbol{i}} + \mathcal{O}(h^4). \tag{4.2}$$

The second is multiplication of cell-averaged quantities:

$$\langle \phi \Psi \rangle_{\boldsymbol{i}} = \langle \phi \rangle_{\boldsymbol{i}} \langle \Psi \rangle_{\boldsymbol{i}} + \mathcal{O}(h^2) \tag{4.3}$$

$$= \langle \phi \rangle_{\boldsymbol{i}} \langle \Psi \rangle_{\boldsymbol{i}} + \frac{h^2}{12} \sum_{d=1}^{D} \left[\left(\frac{\partial \phi}{\partial x_d} \right)_{\boldsymbol{i}} \left(\frac{\partial \Psi}{\partial x_d} \right)_{\boldsymbol{i}} \right] + \mathcal{O}(h^4). \tag{4.4}$$

The third is multiplication of face-averaged quantities:

$$\langle \phi \Psi \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle \phi \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle \Psi \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \mathcal{O}(h^2) \tag{4.5}$$

$$= \langle \phi \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle \Psi \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12} \sum_{d' \neq d}^{D} \left[\left(\frac{\partial \phi}{\partial x_{d'}} \right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \left(\frac{\partial \Psi}{\partial x_{d'}} \right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right] + \mathcal{O}(h^4). \tag{4.6}$$

It is worth noting that for higher than second-order accuracy, there are additional terms to compute in these discretizations. Each of these additional terms is a product of derivatives. The derivatives need only be computed to second-order accuracy using the standard centered-difference scheme [13]. For this work, centered-differences are used to compute the transverse derivatives instead of point-wise values,

$$\left(\frac{\partial \phi}{\partial x_d}\right)_{\boldsymbol{i}} = \left(\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\right)/(2h) \tag{4.7}$$

$$\left(\frac{\partial \phi}{\partial x_{d'}}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \left(\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d+\boldsymbol{e}^{d'}} - \langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d-\boldsymbol{e}^{d'}}\right)/(2h). \tag{4.8}$$

Interpolating between Cell-Averages and Face-Averages

The next challenge is to move between cell-averaged and face-averaged quantities and vice-versa.

Here is the approach for performing the first operation:

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{2}\left[\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}}\right] + \mathcal{O}(h^2) \tag{4.9}$$

$$= \frac{1}{12}\left[7\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} + 7\langle\phi\rangle_{\boldsymbol{i}} - \langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right] + \mathcal{O}(h^4). \tag{4.10}$$

The following notation will be used for this operation,

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathcal{I}_{face}\left(\langle\phi\rangle\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.11}$$

Face-averaged quantities must also be able to be converted to cell-averaged quantities. That is accomplished by the following approximation

$$\langle\phi\rangle_{\boldsymbol{i}} = \frac{1}{2}\left[\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right] + \mathcal{O}(h^2) \tag{4.12}$$

$$= \frac{1}{16}\left[9\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + 9\langle\phi\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}-\frac{3}{2}\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+\frac{3}{2}\boldsymbol{e}^d}\right] + \tag{4.13}$$

$$\frac{h^2}{24}\left(\frac{\partial \phi}{\partial x_d}\right)_{\boldsymbol{i}} + \mathcal{O}(h^4). \tag{4.14}$$

In the previous equation, the partial derivative $\left(\frac{\partial \phi}{\partial x_d}\right)$ may be computed at second-order accuracy.

Again, a shorthand notation for this interpolation is introduced

$$\langle\phi\rangle_{\boldsymbol{i}} = \mathcal{I}_{center}\left(\langle\phi\rangle\right)_{\boldsymbol{i}}. \tag{4.15}$$

Derivatives

Typically derivatives are computed at faces in finite-volume schemes. Here is the approach to computing normal derivatives at faces:

$$\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{h}\left(\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}}\right) + \mathcal{O}(h^2) \tag{4.16}$$

$$= \frac{1}{12h}\left(5\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - 5\langle\phi\rangle_{\boldsymbol{i}} - \langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\right) + \mathcal{O}(h^4) \tag{4.17}$$

Non-normal derivatives at faces require first computing the cell-averaged value of the required derivative, then interpolating the cell-averaged quantity to a face-averaged one. Here is what that approach looks like

$$\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^{d'\neq d}} = \mathcal{I}_{face}\left(\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^{d'\neq d}}. \tag{4.18}$$

To get the derivative as a cell-averaged quantity, simply take the difference of the face-averaged values

$$\left\langle \frac{\partial \phi}{\partial x_d} \right\rangle_{\boldsymbol{i}} = \frac{1}{h}\left[\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right], \tag{4.19}$$

where the order of accuracy is solely determined by the method used to compute $\langle\phi\rangle_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}$.

Divergence and Gradient

These are pretty straightforward after the derivative is defined. The discrete divergence is defined as:

$$D\left(\langle\boldsymbol{v}\rangle\right)_{\boldsymbol{i}} = \frac{1}{h}\sum_{d=1}^{D}\left[\left\langle v^d\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \left\langle v^d\right\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right] \tag{4.20}$$

$$D\left(\langle\boldsymbol{v}\rangle\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathcal{I}_{face}\left(D\left(\langle\boldsymbol{v}\rangle\right)\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.21}$$

The divergence of a tensor field is computed in the same manner, except that a vector is produced.

The gradient of a scalar is defined as

$$G\left(\phi_{\boldsymbol{i}}\right) = \left\langle \frac{\partial \phi}{\partial \boldsymbol{x}} \right\rangle_{\boldsymbol{i}} \tag{4.22}$$

$$G\left(\phi_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\right) = \left\langle \frac{\partial \phi}{\partial \boldsymbol{x}} \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, \tag{4.23}$$

where $\boldsymbol{x} = [x_0, x_1, ..., x_D]$.

The gradient of a vector is computed in a similar manner, except with $\boldsymbol{v}^d$ in place of ϕ in the previous gradient equations. A tensor is created in this case instead of vector.

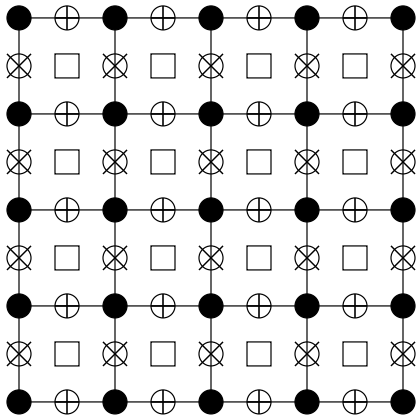


Figure 4.1: Vertex (•), Cell (□) and Face (⊕ and ⊗) Sites on a Grid [21].

4.2 Method of Lines

We recall that the system of equations we wish to solve can be given by,

$$\frac{\partial U}{\partial t} = F\left(U\right) = F^E\left(U\right) + F^I\left(U\right). \tag{4.24}$$

A semi-discrete approach is used to discretize and advance the solution. For the spatial differences, high-order finite-volume discretizations are selected. The solution is advanced in time via the method of lines with an Additive Runge-Kutta (ARK) ODE integrator.

The finite-volume method discretizes the physical domain Ω into a union of control volumes $V_{\boldsymbol{i}}$,

$$V_{\boldsymbol{i}} = [\boldsymbol{i}h, (\boldsymbol{i}+\boldsymbol{e})h] \ , \ \boldsymbol{i} \in \mathbb{Z}^D \ , \ \boldsymbol{e} = (1, 1, ..., 1), \tag{4.25}$$

where h is the isotropic unit grid spacing and $\boldsymbol{i}$ is a D-dimensional index denoting location. Values of the evolved quantities U are stored as cell-averages $\langle U \rangle$ over each $V_{\boldsymbol{i}}$, and the fluxes $\boldsymbol{F}^d$ as well as $\boldsymbol{v}_p$ are stored as averages $\left\langle \boldsymbol{F}^d \right\rangle_{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d}$ over the surface faces $A_d^{\pm}$ of each cell:

$$\langle U \rangle_{\boldsymbol{i}}(t) = \frac{1}{h^D} \int_{V_{\boldsymbol{i}}} U(\boldsymbol{x}, t) d\boldsymbol{x}, \tag{4.26}$$

$$\left\langle \boldsymbol{F}^d \right\rangle_{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d}(t) = \frac{1}{h^{D-1}} \int_{A_d^{\pm}} \boldsymbol{F}^d(\boldsymbol{x}, t) \ d\boldsymbol{x}. \tag{4.27}$$

For a graphical depiction of where values are stored, please see Figure 4.1. All of the variables are collocated at cell-centers, except for the curl-free velocity $\boldsymbol{v}_p$. The curl-free velocity is used to compute divergence and as such is centered on faces.

Applying the finite-volume discretization (Equation (4.25)) to the IMEX formulation (Equation (4.24)) yields a semi-discrete system of ordinary differential equations (ODE) in time

$$\frac{d\langle U\rangle_{\boldsymbol{i}}}{dt} = \frac{1}{h^D}\int_{V_{\boldsymbol{i}}} \left[F^{[E]}(U) + F^{[I]}(U)\right] d\boldsymbol{x}, \tag{4.28}$$

where $F^{[E]}$ is the collection of explicit divergence of fluxes and source terms for the system, and $F^{[I]}$ is the implicit counterpart.

The integration of the above system Equation (4.28) with respect to time from t^n to t^{n+1} produces the solution,

$$\langle U\rangle_{\boldsymbol{i}}^{n+1} = \langle U\rangle_{\boldsymbol{i}}^{n} + \Delta t \sum_{[E],[I]} \left\langle F^{[E,I]}\right\rangle_{\boldsymbol{i}}^{\hat{n}} \tag{4.29}$$

$$\left\langle F^{[E,I]}\right\rangle_{\boldsymbol{i}}^{\hat{n}} = \frac{1}{\Delta t}\int_{t^n}^{t^n+\Delta t} \langle F\rangle_{\boldsymbol{i}}^{[E,I]}(t)dt. \tag{4.30}$$

4.3 Time Integration

As mentioned previously, the IMEX ODE integrator requires a splitting of the governing equations into terms that act on different time scales. The stiff terms, resolved on time scales defined by acoustic wave speed, diffusion, or combustion, are treated implicitly $[I]$. The non-stiff terms, resolved on time scales defined by the bulk fluid velocity, are treated explicitly $[E]$.

The Additive Runge-Kutta Method (ARK) is selected for the IMEX ODE integrator. These methods are composed of two or more Runge-Kutta (RK) methods. Typically, one RK scheme is chosen to integrate all of the explicit terms and another RK scheme is chosen to integrate the implicit terms. Explicit Runge-Kutta (ERK) schemes are typically used for the non-stiff terms and Explicit First Stage, Diagonal Implicit Runge-Kutta (ESDIRK) schemes are used for the stiff terms. Any combination of RK methods and splittings is theoretically possible, but only two are used in practice. The two particular schemes are carefully selected such that coupling and order conditions are satisfied [50]. In practice, this leads to the selection of RK methods that share the same $b_i^{[E,I]} = b_i$ and $c_i^{[E,I]} = c_i$ coefficients in their respective Butcher Tableau. In this study the ARK4(3)6L[2]SA scheme is used, and its coefficients are provided in [50]. It is a a six-stage, fourth-order accurate, L-stable method.

The conceptual idea is to split the ODE into explicit and implicit contributors

$$\frac{d\langle U\rangle}{dt} = F^{[E]}(\langle U\rangle) + F^{[I]}(\langle U\rangle). \tag{4.31}$$

The next time value is defined by,

$$\langle U\rangle^{n+1} = \langle U\rangle^{n} + \Delta t \sum_{k=1}^{s} b_k \left[F^{[E]}\left(\langle U\rangle^{k}\right) + F^{[I]}\left(\langle U\rangle^{k}\right)\right], \tag{4.32}$$

where

$$\langle U\rangle^k = \langle U\rangle^n + \Delta t\left[\sum_{j=1}^{s} a_{kj}^{[E]} F^{[E]}\left(\langle U\rangle^j\right) + \sum_{k=1}^{s} a_{kj}^{[I]} F^{[I]}\left(\langle U\rangle^j\right)\right]. \tag{4.33}$$

The coefficients b_k, c_k, and $a_{kj}^{[E,I]}$ are given by the Butcher Tableau for each RK method. Also, $\langle U\rangle^n = \langle U\rangle(t^n)$, $\langle U\rangle^{n+1} = \langle U\rangle(t^n + \Delta t)$, and $\langle U\rangle^k = \langle U\rangle(t^n + c_k\Delta t)$.

The solution is advanced by

$$\chi^k = \Delta t \sum_{j=1}^{k-1}\left[a_{kj}^{E} F^{[E]}\left(\langle U\rangle^j\right) + a_{kj}^{I} F^{[I]}\left(\langle U\rangle^j\right)\right], \tag{4.34}$$

$$\langle U\rangle^k = \langle U\rangle^n + \chi^k + \Delta t\gamma F^{[I]}\left(\langle U\rangle^k\right)\ ,\ (k \geq 2). \tag{4.35}$$

4.4 Evaluating the Spatial Operators

The fundamentals of the spatial discretization, including high-order averaging and derivatives have already been introduced. The remainder of this section will discuss how to evaluate the remaining spatial operators belonging to Equation (3.70).

Scalar Advective Transport

The scalar advective terms $\langle\nabla\cdot(\phi\boldsymbol{v})\rangle_{\boldsymbol{i}}$ are evaluated by computing the advective fluxes at cell faces and then taking the divergence

$$\langle\nabla\cdot(\phi\boldsymbol{v})\rangle_{\boldsymbol{i}} \to \frac{1}{h}\sum_{d=1}^{D}\left[\left\langle\rho v^d\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \left\langle\rho v^d\right\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right]. \tag{4.36}$$

Again note that $\left\langle\rho v^d\right\rangle$ is not the simply the product of $\langle\rho\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ and $\left\langle v^d\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ if fourth-order accuracy is desired. The tangential derivatives must be taken into account.

The pressure evolution equation also has an advective term $\langle\boldsymbol{v}\cdot\nabla p\rangle_{\boldsymbol{i}}$, and it is computed in the following manner:

$$\langle\boldsymbol{v}\cdot\nabla p\rangle_{\boldsymbol{i}} \to \sum_{d=1}^{D}\left[\left\langle v^d\frac{\partial p}{\partial x^d}\right\rangle_{\boldsymbol{i}}\right]. \tag{4.37}$$

Advective Velocity Computation

The advective velocity is required on cell-faces for computing the explicit IMEX updates at each stage.

Interpolate $\boldsymbol{v}$ to faces and MAC project (defined later) the result before summing it with the acoustic velocity. In the Colella and Pao splitting, we interpolate $\boldsymbol{v}_d$ to faces instead of $\boldsymbol{v}$.

$$\left\langle v_d^d \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathbb{P}_o^{MAC}\left(\mathcal{I}_{face}\left(\langle \boldsymbol{v} \rangle\right)\right) \tag{4.38}$$

$$\left\langle v^d \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \left\langle v_d^d \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \left\langle v_p^d \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.39}$$

If an operator requires both normal and tangential face-averaged velocities, use simple interpolation:

$$\langle \boldsymbol{v} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathcal{I}_{face}\left(\langle \boldsymbol{v} \rangle\right) + \langle \boldsymbol{v}_p \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.40}$$

In the Compressible Projection splitting the divergence-free velocity is needed on cell-faces, but there is no evolution equation for it. This field is computed by applying the constant-coefficient projection operator to the total velocity field, hence

$$\langle \boldsymbol{v}_d \rangle_{\boldsymbol{i}} = \mathbb{P}_0^0(\boldsymbol{v}_{\boldsymbol{i}}). \tag{4.41}$$

Diffusive Transport

There are several diffusion terms in the implicit and explicit operators, including: $\langle \nabla \cdot \boldsymbol{q} \rangle_{\boldsymbol{i}}$ and $\langle \nabla \cdot \rho D_k \nabla Y_k \rangle_{\boldsymbol{i}}$. All the diffusive terms are evaluated in a similar manner. First evaluate the gradient terms $\langle \nabla Y_k \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ and $\langle \nabla h \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ on cell faces. The divergence of these gradient terms at cell-centers is all that is required. Accordingly, we only need to compute the normal derivatives on cell faces.

Once the normal derivatives are computed, scale the derivatives by their corresponding scalars to produce the final diffusive fluxes.

Viscous Tensor Term

The discretization of the viscous tensor term $\left\langle \frac{1}{\rho} \nabla \cdot \tau \right\rangle_{\boldsymbol{i}}$ is now defined. Recall the definition of τ:

$$\boldsymbol{\tau} = 2\eta \boldsymbol{e} + \left(\kappa - \frac{2}{3}\eta\right)(\nabla \cdot \boldsymbol{v}). \tag{4.42}$$

Since the divergence of τ is required, compute $\langle \tau \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ as a face-average. The discretization approach for $(\nabla \cdot \boldsymbol{v})_{\boldsymbol{i}}$ has already been discussed. The approach for computing the cell-averaged $\langle \boldsymbol{e} \rangle_{\boldsymbol{i}}$ will be provided shortly. In this context, compute the strain rate tensor and divergence as face-averages.

$$\langle \boldsymbol{e} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{2}\left(\langle \nabla \boldsymbol{v} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \langle \nabla \boldsymbol{v} \rangle^T_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\right) \tag{4.43}$$

$$\langle \nabla \cdot \boldsymbol{v} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathcal{I}_{face}\left(\langle \nabla \cdot \boldsymbol{v} \rangle\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.44}$$

A boundary condition needs to be imposed on $\langle \nabla \cdot \boldsymbol{v} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ and on the transverse derivatives. The no-slip boundary condition is selected, and the following is true in that case:

$$\langle \nabla \cdot \boldsymbol{v} \rangle|_{\partial\Omega} = \left. \frac{\partial v^n}{\partial x_n} \right|_{\partial\Omega}. \tag{4.45}$$

Projection

Projection operators are required for computing updates to evolved variables as well for synchronizing the compressible velocity to the total velocity.

Constant-Coefficient Projection

Recall that the constant-coefficient projection must be computed, regardless of which form of the splitting is used. The term $\mathbb{P}_0^h \left(\langle \boldsymbol{\Psi} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right)$ is sought, where $\boldsymbol{\Psi}$ is a face-centered vector field. We define this discrete projection operator $\mathbb{P}_0^h$ in the following manner

$$\mathbb{P}_0^h = \mathbb{I} - \mathbb{Q}_0^h, \tag{4.46}$$

where

$$\mathbb{Q}_0^h = G \left(DG \right)^{-1} D, \tag{4.47}$$

and G and D are the discrete gradient and divergence operators. Going forward we will drop the superscript h and just assume we are discussing the discrete implementation of all projection operators.

Typically a face-centered projection is desired, as opposed to a cell-centered one. The face-centered version is denoted by $\mathbb{P}_0^{MAC}$ as opposed to the cell-centered projection $\mathbb{P}_0^0$.

The divergence of the vector field $\boldsymbol{\Psi}$ is simple enough to compute, particularly because the vector field already exists on cell faces. The solve then looks like

$$DG \langle \phi \rangle_{\boldsymbol{i}} = D \left(\boldsymbol{\Psi}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right). \tag{4.48}$$

This solve produces a cell-averaged $\langle \phi \rangle_{\boldsymbol{i}}$.

The boundary condition on the scalar ϕ is the same Neumann type for the constant and variable-coefficient projection operators:

$$\left. \left\langle \frac{\partial \phi}{\partial x_n} \right\rangle \right|_{\partial\Omega} = 0. \tag{4.49}$$

The gradient of $\langle \phi \rangle_{\boldsymbol{i}}$ is easily computed afterwards. Since the interest is in $\mathbb{P}_0^{MAC}$, compute the face-averaged gradient of $\langle \phi \rangle_{\boldsymbol{i}}$. Once this gradient is computed, the $\mathbb{Q}_0^{MAC}$ computation is complete. The full projection operator, $\mathbb{P}_0^{MAC}$, simply requires negating $\mathbb{Q}_0^{MAC}$ and incrementing by the original input.

Variable-Coefficient Projection

Although this projection is not required in the Compressible Projection splitting, it is required in the Colella and Pao splitting. The term $\mathbb{P}_\rho\left(\langle \boldsymbol{A}_d(\boldsymbol{v})\rangle\right)_{\boldsymbol{i}}$ shows up in the divergence-free velocity evolution equation. Recall that $\mathbb{P}_\rho$ is defined as

$$\mathbb{P}_\rho = \mathbb{I} - \mathbb{Q}_\rho, \tag{4.50}$$

where

$$\mathbb{Q}_\rho = \frac{1}{\rho} G \left(D \frac{1}{\rho} G \right)^{-1} D. \tag{4.51}$$

The inverted density is needed both as a cell-average and face-average field,

$$\langle 1/\rho \rangle_{\boldsymbol{i}} = 1/\langle \rho \rangle_{\boldsymbol{i}} \tag{4.52}$$

$$\langle 1/\rho \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = 1/\mathcal{I}_{face}\left(\langle \rho \rangle\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.53}$$

The divergence of the vector field $\langle \boldsymbol{A}_d(\boldsymbol{v})\rangle_{\boldsymbol{i}}$ is simple enough to compute. The solve requires a little more care, and it looks like

$$D \left\langle \frac{1}{\rho} \nabla \phi \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = D \langle \boldsymbol{A}_d(\boldsymbol{v}) \rangle_{\boldsymbol{i}}. \tag{4.54}$$

This solve produces a cell-averaged $\langle \phi \rangle_{\boldsymbol{i}}$.

The boundary condition on the scalar ϕ is

$$\left\langle \frac{\partial \phi}{\partial x_n} \right\rangle \Bigg|_{\partial\Omega} = 0 \tag{4.55}$$

Once $\langle \phi \rangle_{\boldsymbol{i}}$ is known, then compute $\left\langle \frac{1}{\rho} \nabla \phi \right\rangle_{\boldsymbol{i}}$. This last term is evaluated as the product of two cell-averaged vector fields after the gradient of $\langle \phi \rangle_{\boldsymbol{i}}$ is computed. This closes out the computation for $\mathbb{Q}^0_\rho$.

Vector Advective Transport

In the Colella and Pao splitting the acoustic and advective velocity equations contain specific, distinct advective terms. Recall that the advective term in the acoustic velocity evolution is $\left\langle \nabla \left(\frac{|\boldsymbol{v}_p|^2}{2} \right) \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$. It has already been stated how to compute the face-centered gradient of a cell-centered scalar field, so all that needs to done is to describe how to compute $\langle |\boldsymbol{v}_p|^2 \rangle_{\boldsymbol{i}}$.

$$\left\langle |\boldsymbol{v}_p|^2 \right\rangle_{\boldsymbol{i}} = \mathcal{I}_{center}\left(\langle \boldsymbol{v}_p \cdot \boldsymbol{v}_p \rangle \right)_{\boldsymbol{i}}. \tag{4.56}$$

The dot product is defined as:

$$\langle \boldsymbol{v}_p \cdot \boldsymbol{v}_p \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \rightarrow \sum_{d=1}^{D} \left[\left\langle v_p^d v_p^d \right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \right]. \tag{4.57}$$

The other advective term in the Colella and Pao splitting is $\langle \boldsymbol{A}_d(\boldsymbol{v})\rangle_{\boldsymbol{i}}$. In principle this term can be described a few different ways:

$$\begin{aligned}\langle \boldsymbol{A}_d(\boldsymbol{v})\rangle_{\boldsymbol{i}} &= \langle \boldsymbol{v}\cdot\nabla\boldsymbol{v}\rangle_{\boldsymbol{i}} - \left\langle \nabla\left(|\boldsymbol{v}_p|^2/2\right)\right\rangle_{\boldsymbol{i}} \\ &= \langle (\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v}_d\rangle_{\boldsymbol{i}} + \langle (\boldsymbol{v_d}\cdot\nabla)\,\boldsymbol{v}_p\rangle_{\boldsymbol{i}} \\ &= \langle \nabla\cdot(\boldsymbol{v}\boldsymbol{v}_d + \boldsymbol{v}_d\boldsymbol{v}_p)\rangle_{\boldsymbol{i}} - \langle \boldsymbol{v}_d\,(\nabla\cdot\boldsymbol{v}_p)\rangle_{\boldsymbol{i}}\end{aligned} \tag{4.58}$$

The final expression is the one used here. The second term is computed as a standard product of two cell-averaged quantities. Expanding the divergence term yields

$$\langle \nabla\cdot(\boldsymbol{v}\boldsymbol{v}_d + \boldsymbol{v}_d\boldsymbol{v}_p)\rangle_{\boldsymbol{i}} \to \frac{1}{h}\sum_{d=1}^{D}\left[\left\langle v^d\boldsymbol{v}_d + v_d^d\boldsymbol{v}_p\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \left\langle v^d\boldsymbol{v}_d + v_d^d\boldsymbol{v}_p\right\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right]. \tag{4.59}$$

The normal velocities $\left\langle v^d\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ and $\left\langle v_d^d\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ are required on faces, and the total velocity vectors on faces for $\langle \boldsymbol{v}_d\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ and $\langle \boldsymbol{v}_p\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ are also needed.

In the Compressible Projection splitting the term $\langle \boldsymbol{v}\cdot\nabla\boldsymbol{v}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is needed, both at cell faces and centers.

One approach would be to do the following,

$$\langle \boldsymbol{v}\cdot\nabla\boldsymbol{v}\rangle_{\boldsymbol{i}} = \langle \nabla\cdot(\boldsymbol{v}\boldsymbol{v})\rangle_{\boldsymbol{i}} - \langle \boldsymbol{v}\,(\nabla\cdot\boldsymbol{v})\rangle_{\boldsymbol{i}}\,, \tag{4.60}$$

where the expansion of the divergence term is similar to Equation (4.59) above.

The approach used herein is to combine the advective operators computed in the Colella and Pao splitting such that,

$$\langle \boldsymbol{v}\cdot\nabla\boldsymbol{v}\rangle_{\boldsymbol{i}} = \langle \boldsymbol{A}_d(\boldsymbol{v})\rangle_{\boldsymbol{i}} + \mathcal{I}_{center}\left(\left\langle \nabla\left(\frac{|\boldsymbol{v}_p|^2}{2}\right)\right\rangle\right)_{\boldsymbol{i}} \tag{4.61}$$

$$\langle \boldsymbol{v}\cdot\nabla\boldsymbol{v}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathcal{I}_{face}\left[\langle \boldsymbol{A}_d(\boldsymbol{v})\rangle\right]_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \left\langle \nabla\left(\frac{|\boldsymbol{v}_p|^2}{2}\right)\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \tag{4.62}$$

Viscous Dissipation

There are two apparent manners in which to compute this term. The first is to simply deconvolve the cell-averaged variables to cell centers and compute all the terms at cell centers. Then convolve the final solution back to cell-averages. For extremely complicated non-linear terms, this is likely the best approach. One may also go ahead and compute all intermediate terms as cell-averages and use the formulae presented earlier for products.

The dissipation term is given by:

$$\boldsymbol{e} = \frac{1}{2}\left(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T\right) \tag{4.63}$$

$$\Phi = 2\eta(\boldsymbol{e}:\boldsymbol{e}) + \left(\kappa - \frac{2}{3}\eta\right)(\nabla\cdot\boldsymbol{v})^2. \tag{4.64}$$

First, define the velocity gradient terms in the rate of strain tensor

$$\langle \nabla \boldsymbol{v} \rangle_{\boldsymbol{i}}^{d} = \left\langle \frac{\partial v^d}{\partial \boldsymbol{x}} \right\rangle_{\boldsymbol{i}} \tag{4.65}$$

$$\left(\langle \nabla \boldsymbol{v} \rangle_{\boldsymbol{i}}^{d} \right)^T = \left\langle \frac{\partial \boldsymbol{v}}{\partial x_d} \right\rangle_{\boldsymbol{i}} . \tag{4.66}$$

Complete the specification by defining the tensor contraction $\boldsymbol{e} : \boldsymbol{e}$:

$$\langle \boldsymbol{e} : \boldsymbol{e} \rangle_{\boldsymbol{i}} \to \sum_{i=1}^{D} \sum_{j=1}^{D} \left\langle e^{ij} e^{ij} \right\rangle_{\boldsymbol{i}} . \tag{4.67}$$

Species Generation

This collection of terms is highly non-linear. However, it is a point-wise computation such that no spatial derivatives appear. The approach taken here is to fully deconvolve to cell-centered values, compute the generation terms, and then convolve to cell-averages. This is the most robust way of performing the computation, particularly since this computation is typically performed by third party software that does not have high-order finite-volume machinery built in. A discussion of how to compute this term is given in Appendix C.

Boundary Conditions

For each of these standard operations, boundary conditions must be considered. The approach taken here is to use the standard stencils described above, but to provide ghost-cell values when the stencils reach outside of the valid domain.

For periodic functions, merely copy the corresponding value(s) from the opposite side of the problem domain into the ghost cells. So if $\boldsymbol{i}$ is in the left-most d-column of the valid domain, then the values in the ghost cells to the left would be

$$\langle \phi \rangle_{\boldsymbol{i}-\boldsymbol{e}^d} = \langle \phi \rangle_{\boldsymbol{i}+(N-1)\boldsymbol{e}^d} \tag{4.68}$$

$$\langle \phi \rangle_{\boldsymbol{i}-2\boldsymbol{e}^d} = \langle \phi \rangle_{\boldsymbol{i}+(N-2)\boldsymbol{e}^d} , \tag{4.69}$$

where N is the number of cells in the d-direction of the valid domain.

For solid wall boundary conditions, values must be computed corresponding to $\langle \phi \rangle_{\boldsymbol{i}-\boldsymbol{e}^d}$ and $\langle \phi \rangle_{\boldsymbol{i}-2\boldsymbol{e}^d}$. Furthermore, there are different computations depending on the particular boundary condition being enforced.

For scalar fields, it is assumed that there is zero normal derivative at the domain boundary

$$\left. \frac{\partial \phi}{\partial x_n} \right|_{\partial \Omega} = 0. \tag{4.70}$$

This condition may be used on the derivative along with valid domain values to build an extrapolating polynomial to any order of accuracy. Evaluating the polynomial at the ghost cells, the following stencils are produced

$$\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} = \langle\phi\rangle_{\boldsymbol{i}} + \mathcal{O}(h^2) \tag{4.71}$$

$$= \frac{1}{11}\left[9\langle\phi\rangle_{\boldsymbol{i}} + 3\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right] + \mathcal{O}(h^4) \tag{4.72}$$

$$\langle\phi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d} = \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} + \mathcal{O}(h^2) \tag{4.73}$$

$$= \frac{1}{11}\left[-30\langle\phi\rangle_{\boldsymbol{i}} + 56\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - 15\langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right] + \mathcal{O}(h^4). \tag{4.74}$$

For velocity fields, the solid wall boundary condition (with no-slip) requires that all the components are zero at the boundary

$$\boldsymbol{v}|_{\partial\Omega} = 0. \tag{4.75}$$

The extrapolated ghost cell values are then:

$$\langle\boldsymbol{v}\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} = -\langle\boldsymbol{v}\rangle_{\boldsymbol{i}} + \mathcal{O}(h^2) \tag{4.76}$$

$$= \frac{1}{3}\left[-13\langle\boldsymbol{v}\rangle_{\boldsymbol{i}} + 5\langle\boldsymbol{v}\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\boldsymbol{v}\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right] + \mathcal{O}(h^4) \tag{4.77}$$

$$\langle\boldsymbol{v}\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d} = -\langle\boldsymbol{v}\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} + \mathcal{O}(h^2) \tag{4.78}$$

$$= \frac{1}{3}\left[-70\langle\boldsymbol{v}\rangle_{\boldsymbol{i}} + 32\langle\boldsymbol{v}\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - 7\langle\boldsymbol{v}\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right] + \mathcal{O}(h^4). \tag{4.79}$$

4.5 Organizing ARK and Solvers

All of the equations are discretized in space according to the descriptions provided in the previous section. The resulting ODEs are integrated in time using the ARK framework. However, we still need to describe the method of solving for the implicit update at each ARK stage.

Since an ESDIRK method is used for integrating the implicit operators, each new stage value involves the solution of a set of equations. The functional form of the solve is given by

$$\left(\mathbb{I} - F^{[I]}\right)\langle U\rangle^k = \langle U\rangle^n + \chi^k \tag{4.80}$$

$$\chi^k = \Delta t \sum_{j=1}^{k-1}\left[a^E_{kj}F^{[E]}\left(\langle U\rangle^j\right) + a^I_{kj}F^{[I]}\left(\langle U\rangle^j\right)\right], \tag{4.81}$$

where we have already discussed in detail how to compute each term in χ^k.

Inviscid Gas Dynamics Solver

Recall the IMEX form the inviscid gas dynamics (Equation (3.52)). Extracting the pressure and curl-free evolution equations because of their implicit dependence upon one another leads to,

$$\langle p \rangle^k = \tilde{p}^k - \Delta t_k \gamma \tilde{p}^k \left[D(\boldsymbol{v}_p)^k \right] \tag{4.82}$$

$$\boldsymbol{v}_p^k = \tilde{\boldsymbol{v}_p}^k - \frac{\Delta t_k}{\tilde{\rho}^k} G(p^k), \tag{4.83}$$

where

$$\tilde{\phi}^k = \phi^k + \chi_\phi(\phi)^k, \tag{4.84}$$

and χ_ϕ is the component of χ in Equation (4.81) corresponding to ϕ.

It should be noted that the scaling pressure attached to the divergence of $\boldsymbol{v}_p$ is lagged in the right hand side of Equation (4.82). Following the the ARK stage solve context Equation (4.80) and gathering the implicit terms reveals,

$$\begin{bmatrix} \mathbb{I} & \Delta t_k \gamma \tilde{p}^k D \\ -\frac{\Delta t_k}{\tilde{\rho}^k} G & \mathbb{I} \end{bmatrix} \begin{bmatrix} p^k \\ \boldsymbol{v}_p \end{bmatrix} = \begin{bmatrix} \tilde{p}^k \\ \tilde{\boldsymbol{v}_p}^k \end{bmatrix}$$

.

Substituting the $\boldsymbol{v}_p$ expression into the pressure equation produces a Helmholtz type of equation for the new pressure $\langle p \rangle^k$,

$$\left[\mathbb{I} - (\Delta t_k)^2 \gamma \tilde{p}^k D \frac{1}{\tilde{\rho}^k} G \right] [p^k] = \tilde{p}^k - \Delta t_k \gamma \tilde{p}^k \left[D(\tilde{\boldsymbol{v}}_p^k) \right], \tag{4.85}$$

with the boundary condition,

$$\left. \frac{\partial p}{\partial x_n} \right|_{\partial \Omega} = 0. \tag{4.86}$$

This is the standard boundary condition used for the Helmholtz solves going forward.

Then the pressure scaling term can be divided through,

$$\left[\frac{1}{\gamma \tilde{p}^k} \mathbb{I} - (\Delta t_k)^2 D \frac{1}{\tilde{\rho}^k} G \right] [p^k] = \frac{1}{\gamma} \mathbb{I} - \Delta t_k \left[D(\tilde{\boldsymbol{v}}_p^k) \right]. \tag{4.87}$$

This equation is linear and may be solved via a variety of standard methods. We elect to use the multigrid method [68]. Then the curl-free velocity may be updated by back-substitution,

$$\boldsymbol{v}_p^k = \tilde{\boldsymbol{v}}_p^k - \frac{\Delta t_k}{\tilde{\rho}^k} G(p^k). \tag{4.88}$$

The total velocity is updated by interpolating the new scaled pressure gradient term from faces to centers,

$$\boldsymbol{v}^k = \tilde{\boldsymbol{v}}^k - \mathcal{I}_{center} \left(\frac{\Delta t_k}{\tilde{\rho}^k} G(p^k) \right). \tag{4.89}$$

Colella and Pao Splitting

The Colella and Pao splitting has the same solve as the Compressible Projection splitting for the equations of inviscid gas dynamics. The difference between the two methods arises in computing the various spatial operators and in the enforcement of constraints (Equation (3.42)).

Navier-Stokes Solver

Returning to the IMEX form of the Navier-Stokes equations (Equation (3.64)), we remark that there are several more implicit terms and couplings to deal with than with the inviscid equations (Equation (3.52)).

The first step is to solve for the enthalpy update, since the implicitness is self-dependent,

$$(\rho h)^k = (\widetilde{\rho h})^k + \Delta t_k D\left(\frac{\tilde{\lambda}^k}{\tilde{c_p}^k} G(h^k)\right). \tag{4.90}$$

This equation is rearranged to produce a Helmholtz solve for the new enthalpy,

$$\left[\rho^k - \Delta t_k D \frac{\tilde{\lambda}^k}{\tilde{c_p}^k} G\right][h^k] = (\widetilde{\rho h})^k. \tag{4.91}$$

This linear equation has a Neumann boundary condition, and is solved using the multigrid method.

The total velocity is also relatively simple to rearrange and solve for, expanding the viscous term

$$\left[\mathbb{I} - \frac{\Delta t_k}{\tilde{\rho}^k} D\left\{\eta\left(G + G^T\right) + \left(\kappa - \frac{2}{3}\eta\right) D\right\}\right][\boldsymbol{v}^k] = \tilde{\boldsymbol{v}}^k. \tag{4.92}$$

This equation is solved subject to the following boundary conditions,

$$\boldsymbol{v}|_{\partial\Omega} = \bar{0} \tag{4.93}$$

$$\nabla \cdot (\boldsymbol{v})|_{\partial\Omega} = \left.\frac{\partial v^n}{\partial x_n}\right|_{\partial\Omega} \tag{4.94}$$

$$\left.\frac{\partial v^d}{\partial x_{d' \neq d}}\right|_{\partial\Omega} = 0. \tag{4.95}$$

This velocity system is also linear and solved using multigrid.

From here, the pressure and curl-free velocity equations must be updated. Focusing again on the pressure and curl-free evolution equations,

$$\langle p \rangle^k = \tilde{p}^k + \Delta t_k (\gamma - 1) D(\frac{\lambda}{c_p} G(h))^k - \Delta t_k \gamma \tilde{p}^k \left[D(\boldsymbol{v}_p)^k\right] \tag{4.96}$$

$$\boldsymbol{v}_p^k = \tilde{\boldsymbol{v}}_p^{\,k} - \frac{\Delta t_k}{\tilde{\rho}^k} G(p^k) + \frac{\Delta t_k}{\tilde{\rho}^k} D(\boldsymbol{\tau})^k. \tag{4.97}$$

The heat flux term and the viscous stress term have already been computed, so they can be absorbed into the right hand sides for $\tilde{p}^k$ and $\boldsymbol{v}_p^k$ respectively. The remainder of the solve is identical to the inviscid gas dynamics case.

Combustion Solver

The IMEX form of the combustion equations (Equation (3.70)) introduces more evolution equations and stiff terms.

The enthalpy and mass fraction updates are computed first. Focusing on these updates reveals,

$$(\rho h)^k = \left(\widetilde{\rho h}\right)^k + \Delta t_k \left[D\left(q^I\right)\right]^k + \Delta t_k \left(\alpha \sum_{k=1}^{N_s} \left[\left(\beta h_k + \eta_k\right)\dot{\omega}_k\right]\right)^k \tag{4.98}$$

$$(\rho Y_k)^k = \left(\widetilde{\rho Y_k}\right)^k + \Delta t_k \left[D(\rho D_k \nabla Y_k) + (\dot{\omega_k})\right]^k. \tag{4.99}$$

This system may be rearranged in the following manner,

$$-\Delta t_k \left(\alpha \sum_{k=1}^{N_s} \left[\left(\beta h_k + \eta_k\right)\dot{\omega}_k\right]\right)^k - \Delta t_k \left[D\frac{\lambda}{c_p}\nabla h\right]^k + (\rho h)^k = \left(\widetilde{\rho h}\right)^k \tag{4.100}$$

$$\Delta t_k \left[D(\rho D_k \nabla Y_k) + (\dot{\omega_k})\right]^k + (\rho Y_k)^k = \left(\widetilde{\rho Y_k}\right)^k. \tag{4.101}$$

These coupled diffusion-reaction equations (Equations (4.100) to (4.101)) are non-linear. We have two possible approaches to solve the system: either we can try to solve the coupled equations together, or we must use a splitting approach to separate the diffusion and reaction terms. Regardless of which approach we chose we will need a non-linear solver to deal with the reaction terms. We used CVODE [41] to solve systems of non-linear equations in this work.

If we elect to solve the complete coupled system, we must pass everything into CVODE and solve the coupled system. If we decide to split the diffusion and reaction terms, we can proceed as follows,

$$(\rho h)^{k_1} = \left(\widetilde{\rho h}\right)^k + \frac{\Delta t_k}{2}\left(\alpha \sum_{k=1}^{N_s} \left[\left(\beta h_k + \eta_k\right)\dot{\omega}_k\right]\right)^{k_1} \tag{4.102}$$

$$(\rho Y_k)^{k_1} = \left(\widetilde{\rho Y_k}\right)^k + \frac{\Delta t_k}{2}\left[(\dot{\omega_k})\right]^{k_1}. \tag{4.103}$$

Notice that the resulting fields are evaluated at a half time step. What we have done is ignored the diffusion terms to compute an initial update based on the reaction terms. The system Equations (4.102) to (4.103) is solved using CVODE [41].

Now we can compute the diffusive update, and then correct the result with a final reaction solve.

$$(\rho h)^{k_2} = (\rho h)^{k_1} + \Delta t_k \left[\nabla \cdot \left(\frac{\lambda}{c_p}\nabla h\right)\right]^{k_2} \tag{4.104}$$

$$(\rho Y_k)^{k_2} = (\rho Y_k)^{k_1} + \Delta t_k \left[\nabla \cdot (\rho D_k \nabla Y_k)\right]^{k_2}. \tag{4.105}$$

The equations above are linear and of Helmholtz type.

And then the final correction may be made, again using CVODE

$$(\rho h)^{k} = (\rho h)^{k_2} + \frac{\Delta t_k}{2}\left(\alpha \sum_{k=1}^{N_s} \left[(\beta h_k + \eta_k)\,\dot{\omega}_k\right]\right)^{k} \tag{4.106}$$

$$(\rho Y_k)^{k} = (\rho Y_k)^{k_2} + \frac{\Delta t_k}{2}\left[(\dot{\omega}_k)\right]^{k}. \tag{4.107}$$

The remaining implicit terms are solved for via the same procedure described in the previous two sections.

4.6 Enforcing Constraints

For each system of equations (Equation (3.52), Equation (3.64), Equation (3.70)), we have introduced redundant equations. In effect, having redundant equations imposes an initial value constraint on the system. These constraints must be reimposed at regular intervals so that all of the equations are satisfied.

For the most part these constraints are enforced at the end of a time step. Hence the enforcement of these two constraints lives outside of the IMEX framework.

Pressure Constraint

We have a pressure evolution equation and an equation of state. The evolved pressure may drift from the thermodynamic pressure computed by the equation of state. The pressure equation of state for single-component gas dynamics is

$$p_0 = \rho R T, \tag{4.108}$$

and the equation of state for mixtures is

$$p_0 = \rho \mathcal{R} T \sum_{k=1}^{Ns} \frac{Y_k}{W_k}. \tag{4.109}$$

Instead of correcting the pressure every time a new stage value is computed, the pressure is corrected at the end of a complete time step. Three equations are affected by this computation, namely the pressure evolution equation, the curl-free velocity evolution, and the enthalpy evolution. The required corrections to each variable are denoted: (δp , $\delta \boldsymbol{v}_p$, $\delta(\rho h)$).

Single-Species Pressure Correction

Here is the system of equations that must be solved:

$$\delta p = \Delta t \xi \left(p_0 - p\right) - \gamma p \Delta t \left[\nabla \cdot \left(\delta \boldsymbol{v}_p\right)\right] \tag{4.110}$$

$$\delta \boldsymbol{v}_p = -\frac{\Delta t}{\rho} \nabla \delta p \tag{4.111}$$

$$\delta(\rho h) = \frac{\gamma - 1}{\gamma} \delta p. \tag{4.112}$$

The first two form a Helmholtz equation for the pressure,

$$\left[\frac{1}{\gamma p}\mathbb{I} - \left(\Delta t\right)^2 D \frac{1}{\rho} G\right] \left[\delta p\right] = \Delta t \frac{\xi}{\gamma p} \left(p_0 - p\right). \tag{4.113}$$

These equations may be rearranged to produce another Helmholtz solve,

$$\left[\mathbb{I} - \left(\Delta t\right)^2 D \frac{1}{\rho} G\right] \left[\delta p\right] = \Delta t \xi \left(p_0 - p\right) \tag{4.114}$$

Once δp is known, both $\delta \boldsymbol{v}_p$ and $\delta(\rho h)$ may be found via substitution.
The final values of (p^{n+1}, $\boldsymbol{v}_p^{n+1}$, $(\rho h)^{n+1}$) are:

$$p^{n+1} = p + \delta p \tag{4.115}$$

$$\boldsymbol{v}_p^{n+1} = \boldsymbol{v}_p + \delta \boldsymbol{v}_p \tag{4.116}$$

$$(\rho h)^{n+1} = (\rho h) + \delta(\rho h). \tag{4.117}$$

Multi-Species Pressure Correction

Although the idea is the same as the single-species correction, the form of the equations is slightly different based on the evolution equation for the pressure as well as the equation of state for the pressure.

Velocity Constraint

Both of the velocity splittings discussed lead to a velocity constraint. For the Colella and Pao splitting, we have the following constraints on the two evolved velocity fields,

$$\boldsymbol{v}_d = \mathbb{P}_0 \left(\boldsymbol{v}\right), \tag{4.118}$$

$$\boldsymbol{v}_p = \mathbb{Q}_0 \left(\boldsymbol{v}\right). \tag{4.119}$$

There is another relation that must be mentioned, and that is the total velocity given by

$$\boldsymbol{v} = \boldsymbol{v}_d + \boldsymbol{v}_p, \tag{4.120}$$

so at the end of each time step the divergence- and curl-free velocities are projected.

In the Compressible Splitting case, we have total velocity evolution equation and a curl-free velocity evolution equation. We want to ensure that all of the divergence in the total velocity is coming from $\boldsymbol{v}_p$, so at the end of the time-step we project the total velocity to remove the divergence. Then we increment this now divergence-free velocity with the curl-free velocity.

Multi-species Transport Constraints

There are some additional constraints, but they are much easier to deal with. Substitutions can be made directly for these constraints to make sure all of the equations are simultaneously satisfied at all times. These constraints are described and motivated by Day and Bell [25].

Density

The density discrepancy arises from the first two gas dynamics equations. Total density is defined by its own conservation equation, but it is also defined as the sum of the density-scaled mass fractions. This issue is reconciled in the advective term by making the substitution:

$$\langle \nabla \cdot (\rho \boldsymbol{v}) \rangle_{\boldsymbol{i}} \rightarrow \sum_{k=1}^{N_s} \langle \nabla \cdot (\rho Y_k \boldsymbol{v}) \rangle_{\boldsymbol{i}} \tag{4.121}$$

This ensures that both expressions are satisfied.

In order for mass to be conserved, the sum of the diffusive fluxes as well as the species generation terms must be zero

$$\sum_{k=1}^{N_S} \rho \mathcal{D}_k \nabla Y_k = 0 \tag{4.122}$$

$$\sum_{k=1}^{N_S} \dot{\omega}_k = 0 \tag{4.123}$$

The species generation terms are easy to specify such that the total mass will be conserved. However, the diffusive fluxes are more difficult. In the formulation, the explicit simplification that

$$\nabla \cdot (\rho Y_k \boldsymbol{V}_k) \rightarrow -\nabla \cdot (\rho \mathcal{D}_k \nabla Y_k), \tag{4.124}$$

has been made.

The thermomolecular theory states that to first order the following is true (ignoring Soret effects)

$$\boldsymbol{V}_k = -\frac{\mathcal{D}_k}{X_k}\boldsymbol{d}_k + \boldsymbol{V}_{cor} \tag{4.125}$$

$$\boldsymbol{d}_k = \nabla X_k + (X_k - Y_k)\frac{\nabla p}{p}. \tag{4.126}$$

If Fick's law holds for the diffusion then

$$\widetilde{\boldsymbol{V}}_k = -\mathcal{D}_k \frac{\nabla Y_k}{Y_k} \tag{4.127}$$

$$\boldsymbol{V}_k = \widetilde{\boldsymbol{V}}_k + \widetilde{\boldsymbol{V}}_{cor}, \tag{4.128}$$

where $\widetilde{\boldsymbol{V}}_{cor}$ is the amount by which the computed diffusion velocity is not mass conserving. This flux must be included in the mass fraction PDE at some point in order to conserve. The typical approach is to compute an initial update to the mass fractions based upon $\widetilde{\boldsymbol{V}}_k$. Then $\widetilde{\boldsymbol{V}}_{cor}$ is computed and the initial mass fractions are corrected.

Enthalpy

For the case of enthalpy, there are three expressions to account for. The first is an evolution equation for the specific enthalpy The second is an expression for the specific enthalpy of the mixture as a mass- weighted sum of individual species-specific enthalpies. And lastly there is an analytic expression for the individual species-specific enthalpies. The analytical expression typically requires a lookup table. The three types of expressions are provided for reference,

$$\frac{\partial(\rho h)}{\partial t} + \nabla\cdot(\rho h\boldsymbol{v}) + \frac{Dp}{Dt} = \nabla\cdot\boldsymbol{q} + \Phi \tag{4.129}$$

$$h = \sum_{k=1}^{N_s} Y_k h_k \tag{4.130}$$

$$h_k = h_k(T). \tag{4.131}$$

The right hand side of the second expression, $h = \sum_{k=1}^{N_s} Y_k h_k$, should be used anywhere h needs to be evaluated. In making the above substitution, h_k must be evaluated for each species over each finite volume. Most terms in the evolution equation do not pose a problem. The divergence of the heat flux needs to be expounded on. Looking back on the heat flux, it would be better if everything was transformed into enthalpy variables:

$$\nabla \cdot \boldsymbol{q} = \nabla \cdot \lambda \nabla T + \sum_{k=1}^{N_s} \nabla \cdot (\rho \mathcal{D}_k h_k \nabla Y_k) \tag{4.132}$$

$$= \nabla \cdot \frac{\lambda}{c_p} \nabla h - \sum_{k=1}^{N_s} \nabla \cdot \frac{\lambda}{c_p} h_k \nabla Y_k + \sum_{k=1}^{N_s} \nabla \cdot (\rho \mathcal{D}_k h_k \nabla Y_k) \tag{4.133}$$

$$= \nabla \cdot \frac{\lambda}{c_p} \nabla h - \sum_{k=1}^{N_s} \nabla \cdot \left[\left(\frac{\lambda}{c_p} - \rho \mathcal{D}_k \right) h_k \nabla Y_k \right] \tag{4.134}$$

This is the formulation used for evaluating the heat flux in the enthalpy equation.

$$\boldsymbol{q} = q^I + q^E \tag{4.135}$$

$$q^I = \frac{\lambda}{c_p} \nabla h \tag{4.136}$$

$$q^E = \left[\left(\frac{\lambda}{c_p} - \rho \mathcal{D}_k \right) h_k \nabla Y_k \right]. \tag{4.137}$$

Temperature

Temperature is needed to evaluate the equation of state, fluid properties, heat flux, and combustion generation terms. So one can either use the temperature evolution form for these, or compute it using data already at hand. However, seeing as the link between enthalpy and temperature is being leveraged so much already, it makes sense to use enthalpy to define temperature where it is needed. The temperature can be found through using $h, Y_k,$ and h_k.

4.7 Complete Single-Level Algorithm

The complete single-level algorithm is summarized in Figure 4.2.

function TIMESTEP(l , t^l , Δt^l)

(1.) Initialize the state and increment:

$$\langle U \rangle^1 = \langle U \rangle^n \tag{4.138}$$

$$\boldsymbol{k}^1 = \Delta t \left[F^{[E]}(\langle U \rangle^1) + F^{[I]}(\langle U \rangle^1) \right] \tag{4.139}$$

(2.) Compute the subsequent states and increments:

for $s = 2$ **to** nStages **do**

Compute the new state:

$$(\mathbb{I} + \gamma \Delta t_k F^{[I]}) \langle U \rangle^s = \langle U \rangle^n + \chi^s \tag{4.140}$$

$$\chi^s = \Delta t \left(\sum_{j=1}^{s-1} a_{s,j}^{[E]} \left[F^{[E]}(\langle U \rangle^j) \right] + \sum_{j=1}^{s-1} a_{s,j}^{[I]} \left[F^{[I]}(\langle U \rangle^j) \right] \right) \tag{4.141}$$

Compute the new update:

$$\boldsymbol{k}^s = \Delta t \left[F^{[E]}(\langle U \rangle^s) + F^{[I]}(\langle U \rangle^s) \right] \tag{4.142}$$

end for

(3.) Update the solution:

$$\langle U \rangle^{n+1} = \langle U \rangle^n + \sum_{s=1}^{\text{nStages}} b_s \boldsymbol{k}^s \tag{4.143}$$

$$\tag{4.144}$$

(4.) Correct for volume discrepancy and synchronize Velocities:

$$\langle U \rangle^{n+1} = \langle U \rangle^{n+1} + \left\langle \delta U^{n+1} \right\rangle \tag{4.145}$$

$$\langle \boldsymbol{v} \rangle^{n+1} = \mathbb{P}_0 \left(\langle \boldsymbol{v} \rangle^{n+1} \right) + \mathcal{I}_{center} \left(\langle \boldsymbol{v}_p \rangle^{n+1} \right). \tag{4.146}$$

end function

Figure 4.2: Single-Level Algorithm.

Chapter 5

Results

5.1 Removing the Variable-Coefficient Projection

Here we will show success in removing the variable-coefficient projection from the solution procedure for the equations of inviscid gas dynamics.

Single Vortex

The single vortex in a box problem has been studied in multiple contexts, and was included in [84]. The velocity field is divergence-free and produces a single vortex in the middle. The density is a function of vertical position. The initial pressure is varied to achieve different Mach numbers.

$$\boldsymbol{v}(\boldsymbol{x}) = [2\sin^2(\pi x_0)\sin(\pi x_1)\cos(\pi x_1), \tag{5.1}$$

$$- 2\sin^2(\pi x_1)\sin(\pi x_0)\cos(\pi x_0)] \tag{5.2}$$

$$\rho(\boldsymbol{x}) = 1 - 0.5\tanh(10 * (x_1 - 0.5)) \tag{5.3}$$

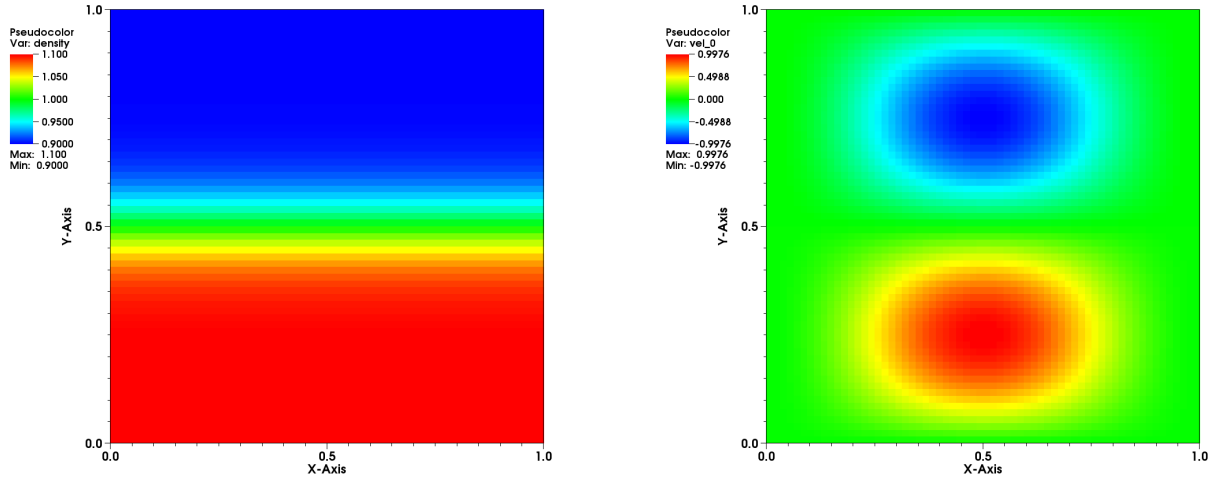
$$p(\boldsymbol{x}) = P0\ ,\ P0 = 10^0, 10^1, 10^2, 10^3, 10^4. \tag{5.4}$$

Here are the initial conditions for the density and velocity for the case when $P0 = 10^6$, which corresponds to $M \sim 0.002$ (Figures 5.1a to 5.1b).

The first results for the single vortex test consist of a qualitative comparison between the Colella and Pao splitting and the new Compressible Projection splitting after many time steps. We used the same fourth-order discretization for both splittings. The difference between the splittings is how the right hand side is computed.

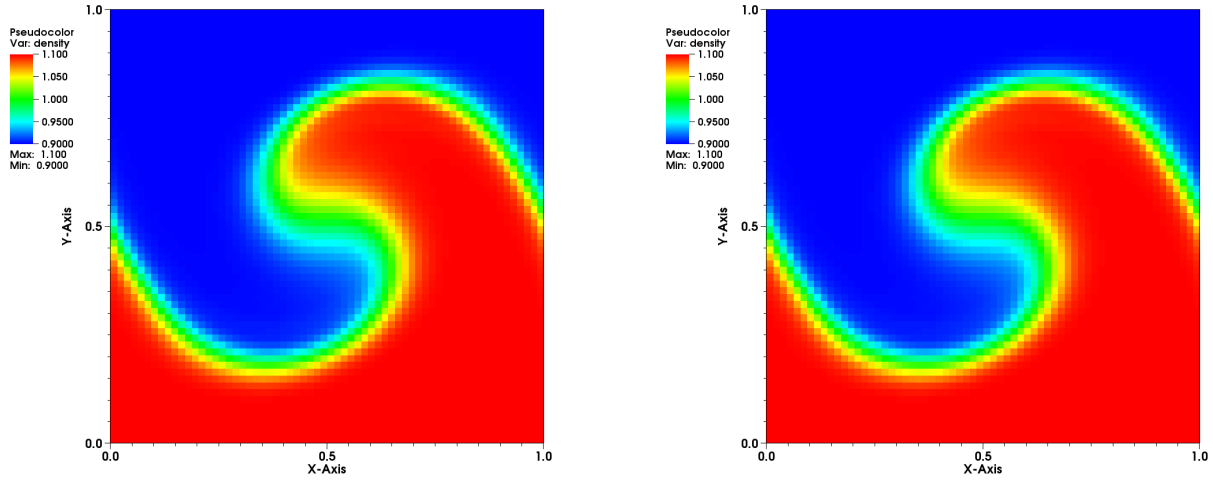
We'll start by comparing the density results of both splittings at $t = 0.5$ (Figure 5.2). We see excellent agreement between both methods for this scalar field.

Now let's take a look at the divergence-free and curl-free velocity fields (Figure 5.3). Here again we see very good agreement between the two methods. The velocities are qualitatively



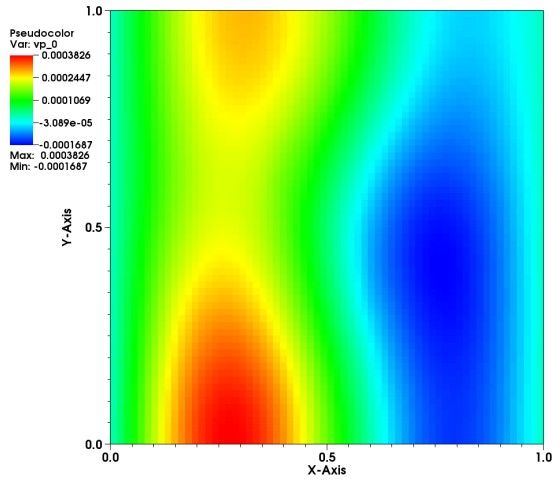
(a) Initial Density (ρ). **(b)** Initial Velocity (v^0).

Figure 5.1: Initial Conditions for Single Vortex Test.

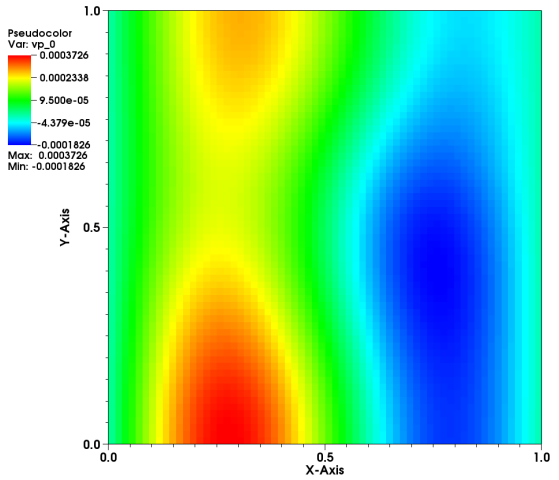


(a) Colella and Pao Splitting (ρ). **(b)** Compressible Projection Splitting (ρ).

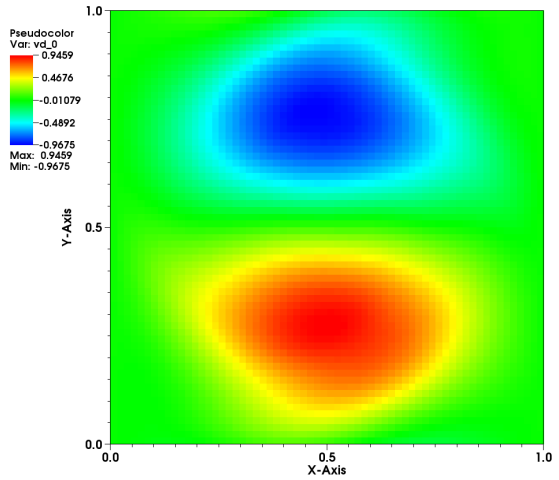
Figure 5.2: Comparison of Splitting Methods on ρ at $t = 0.5$.



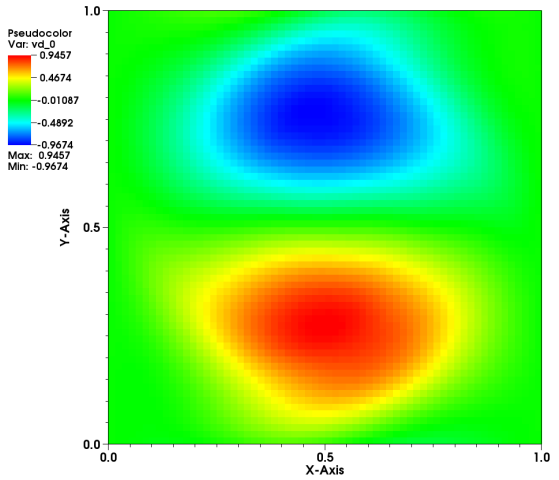
(a) Colella and Pao Splitting (v_p^0).



(b) Compressible Projection Splitting (v_p^0).

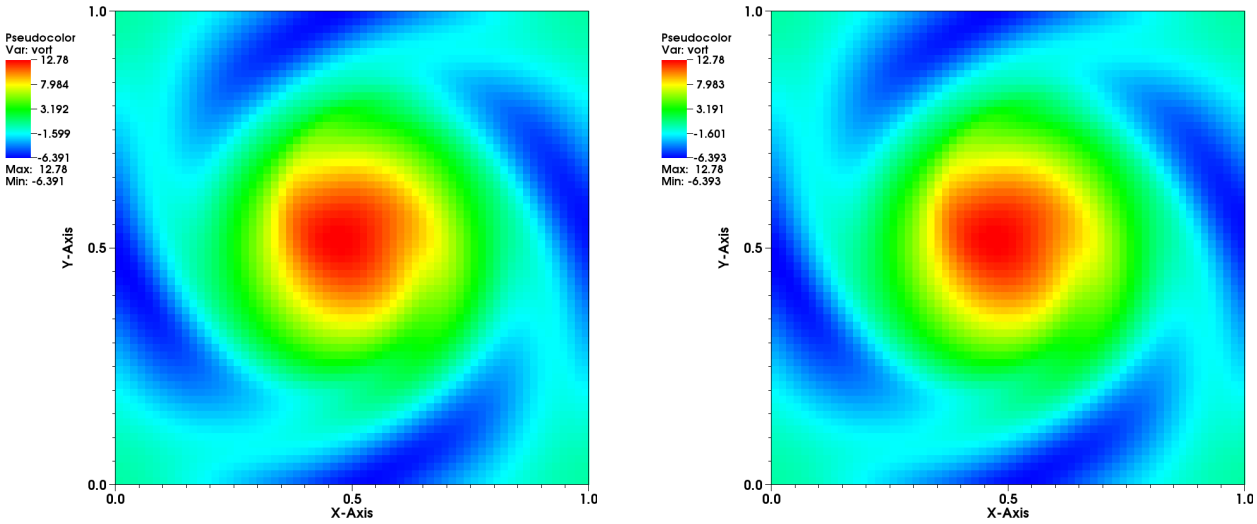


(c) Colella and Pao Splitting (v_d^0).



(d) Compressible Projection Splitting (v_d^0).

Figure 5.3: Comparison of Splitting Methods on (v_p^0, v_d^0) at $t = 0.5$.



(a) Colella and Pao Splitting (w). **(b)** Compressible Projection Splitting (w).

Figure 5.4: Vorticity (w) at $t = 0.5$.

indistinguishable. The max and min are the same out to four digits, which is fine for this coarse grid.

Baroclinic Generation of Vorticity

One of the primary motivating factors for the variable-coefficient projection is the desire to accurately capture baroclinic generation of vorticity. If the splitting developed in this work is to replace the existing one, it also needs to accurately model this vorticity. The vorticity of the two methods using the single vortex initial condition is nearly indistinguishable (Figure 5.4),

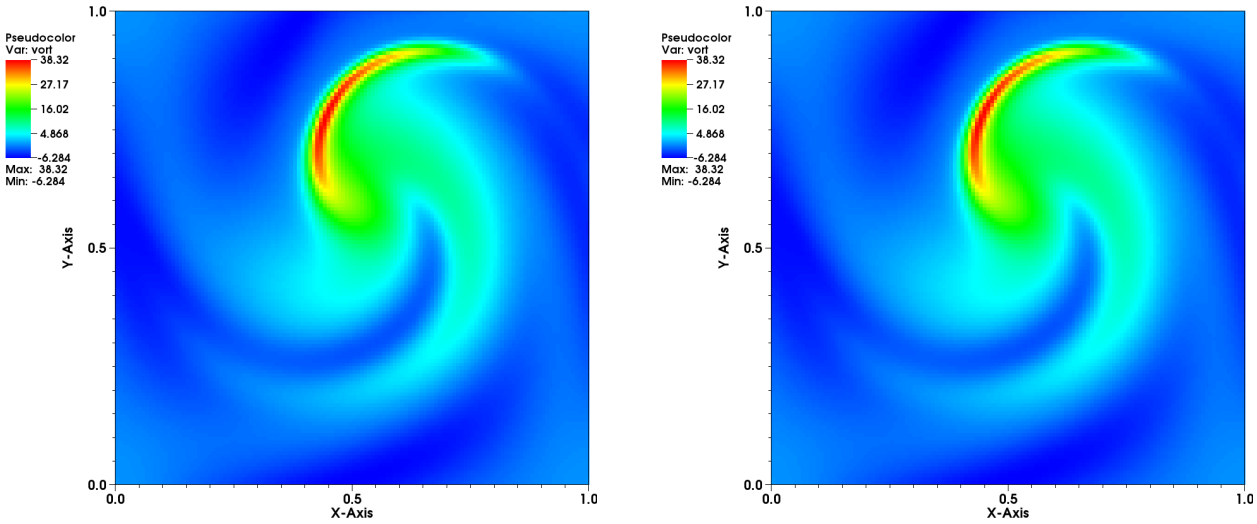
However, in the example given the density variation was pretty mild. We also ran this single vortex test except with the following density field so as to generate additional vorticity to test the new method,

$$\rho(\boldsymbol{x}) = 0.1 - 0.075 * \tanh(10 * (x[1] - 0.5)). \tag{5.5}$$

Figure 5.5 shows the vorticity field generated by the single vortex with this new density field (Equation (5.5)). Again, we notice that the two methods produce comparable results.

Acoustic Pulse

The acoustic pulse test involves simulating a radially symmetric potential flow [71]. The velocity is initially zero. The density is set to a function that varies radially from the center. The pressure is computed assuming that the fluid is polytropic.



(a) Colella and Pao Splitting (w). **(b)** Compressible Projection Splitting (w).

Figure 5.5: Vorticity (w) with Larger Density Variation at $t = 0.5$.

$$\boldsymbol{v}(\boldsymbol{x}) = [0, 0] \tag{5.6}$$

$$r = \sqrt{x_0^2 + x_1^2} \tag{5.7}$$

$$\rho(\boldsymbol{x}) = 1.4 + 0.14 \cos^6(\pi r) \exp(-16r^2) \tag{5.8}$$

$$p(\boldsymbol{x}) = (\rho(\boldsymbol{x}))^\gamma \tag{5.9}$$

The initial density for the acoustic pulse test is shown in (Figure 5.6). As before, we'll compare the scalar fields and the vector fields at a later time for both splittings. The density for the acoustic pulse at $t = 0.5$ is shown in (Figure 5.7). A comparison between the divergence-free and curl-free velocities can be seen in (Figure 5.8).

Again, for all fields there is excellent agreement between the two methods.

5.2 Long Wavelength Acoustics

The purpose of this study is to show that our algorithm can capture the long wavelength acoustics generated by vorticity.

The test consists of a vortex sheet (Section 5.2) aligned along the vertical axis that rotates and rolls up over time.

Vortex Sheet

This test is inspired by the vortex sheet test found in [83]. The initial condition we used for the discrete vortex sheet is given in Equation (5.10). As in the previous test, the velocity

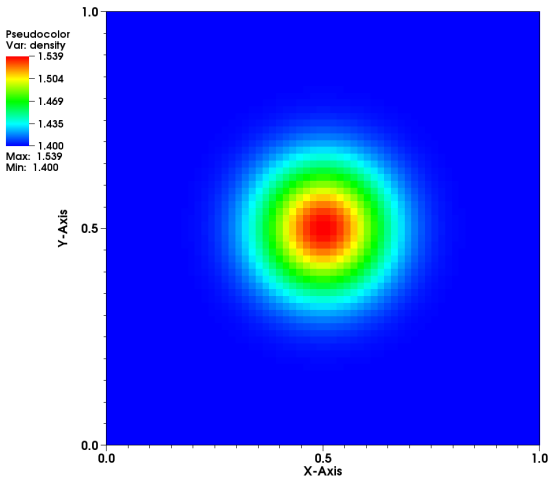
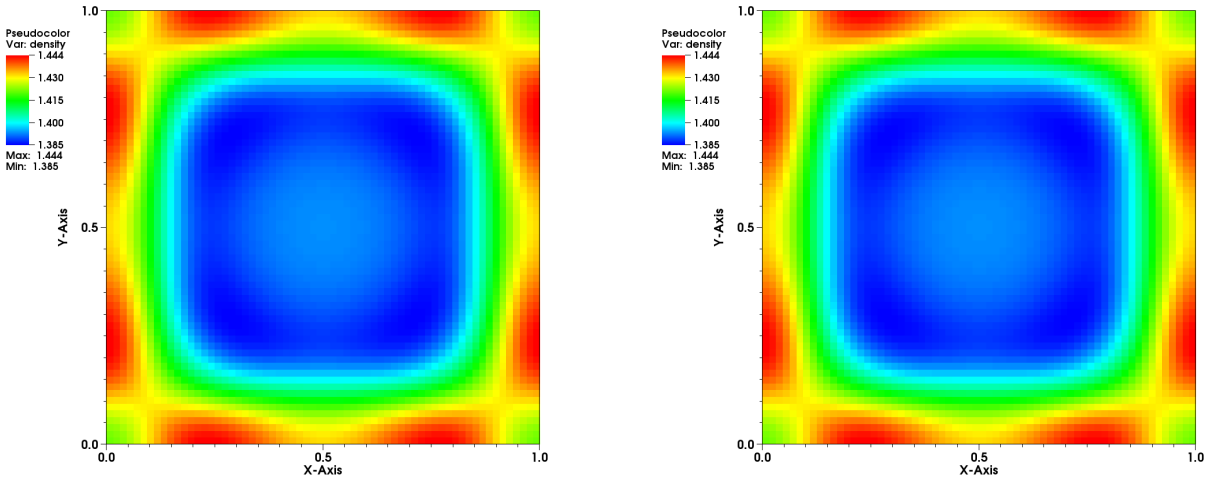
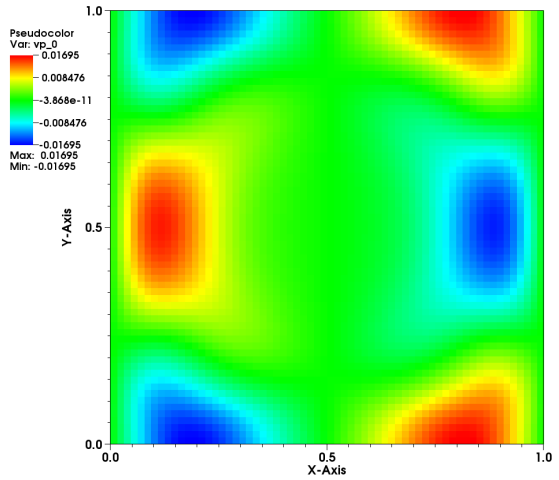


Figure 5.6: Initial Density (ρ) for Acoustic Pulse.

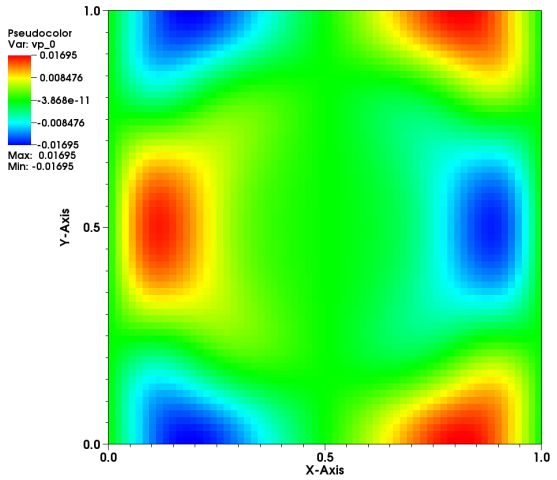


(a) Colella and Pao Splitting (ρ). **(b)** Compressible Projection Splitting (ρ).

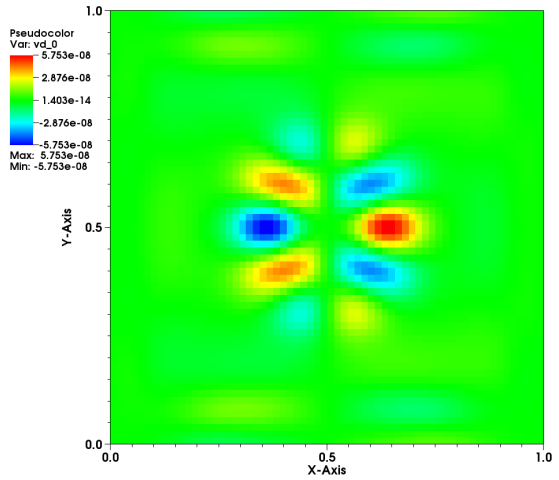
Figure 5.7: Comparison of Splitting Methods on ρ at $t = 0.5$.



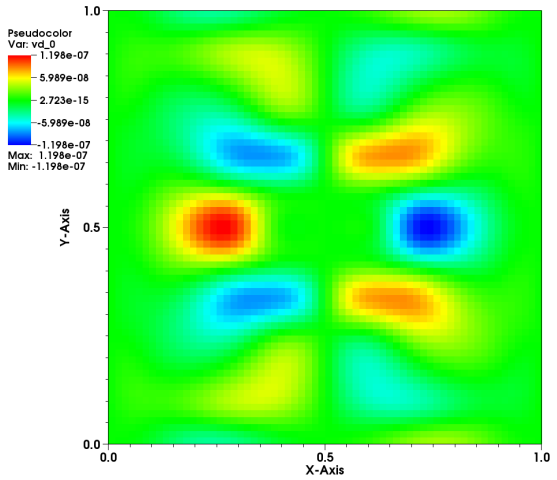
(a) Colella and Pao Splitting (v_p^0).



(b) Compressible Projection Splitting (v_p^0).



(c) Colella and Pao Splitting (v_d^0).



(d) Compressible Projection Splitting (v_d^0).

Figure 5.8: Comparison of Splitting Methods on (v_p^0, v_d^0) at $t = 0.5$.

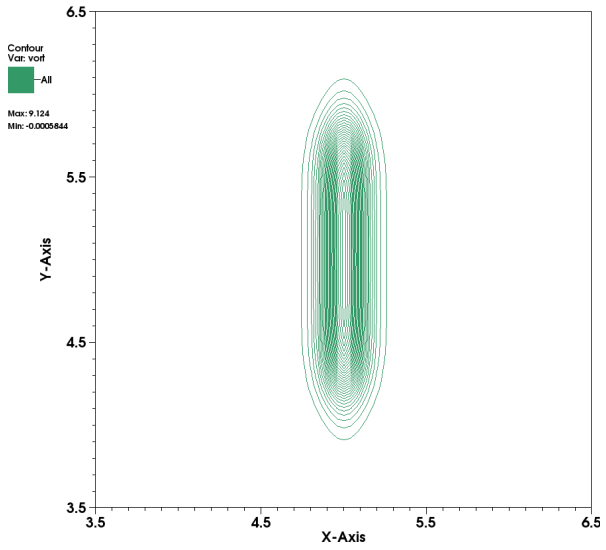


Figure 5.9: Initial Vorticity for Long Wavelength Test.

is projected and the solution is evolved from that point on. The goal of this test is to show continual generation of acoustic signal by the dynamics of vorticity in the center of the domain. This test also begins with $M \sim 0.1$. In the following formulation $l = 10$, $x_c = 5$, and $\delta p = 0.75$,

$$w(x, l, x_c, \delta p) = \begin{cases} \sigma(x, l, x_c, \delta p, +) & \text{if } x < x_c \\ \sigma(x, l, x_c, \delta p, -) & \text{if } x > x_c \end{cases} \tag{5.10}$$

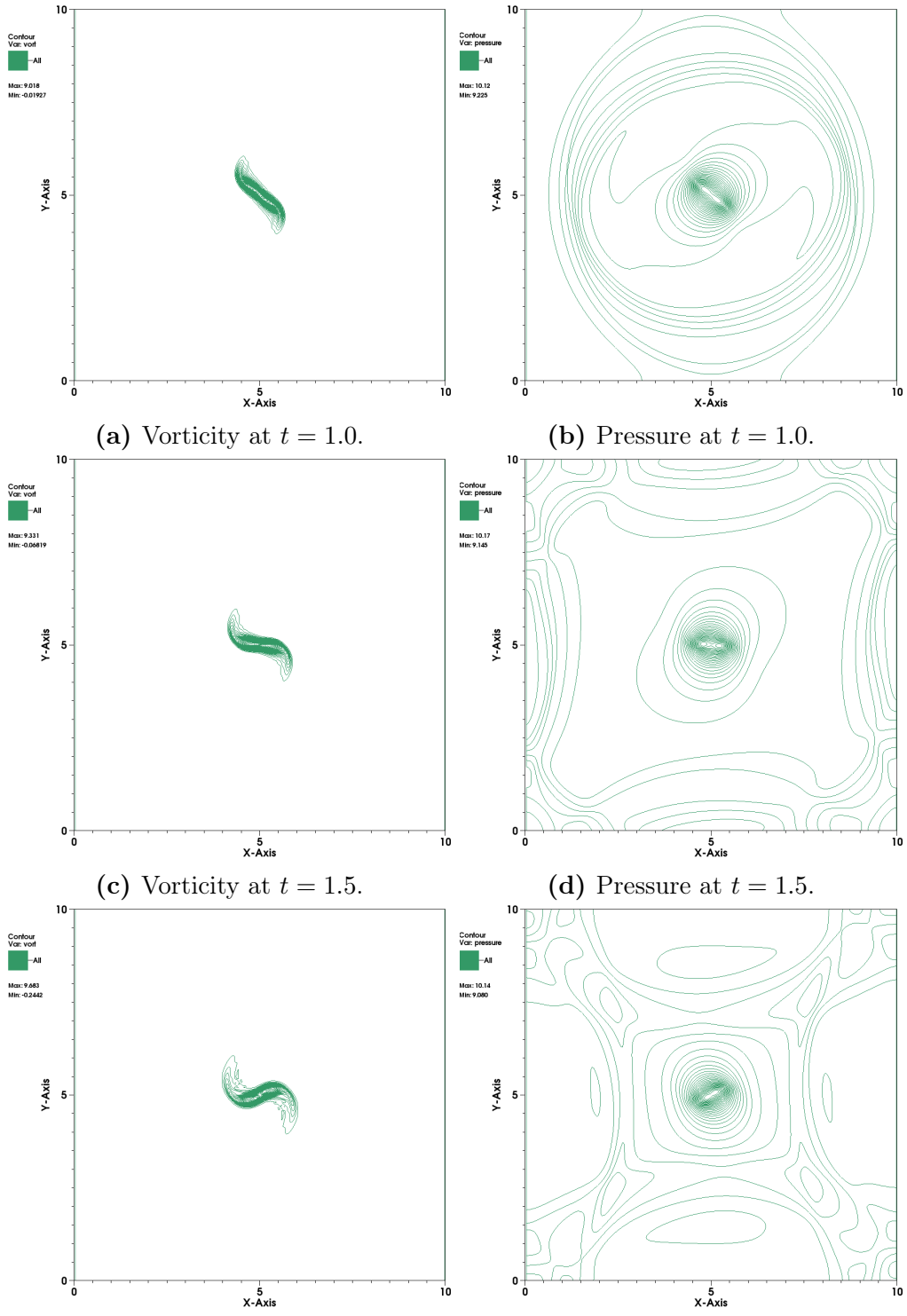
$$\sigma(x, l, x_c, \delta p, \pm) = 1/(1 + \exp(\mp l(x - x_c \mp \delta p))) \tag{5.11}$$

$$\boldsymbol{v} = [0, w(x_1, l, x_c, \delta p) * \tanh((x_0 - x_c)/(\delta p))] \tag{5.12}$$

$$p = 100 \tag{5.13}$$

$$\rho = 1. \tag{5.14}$$

For this test, we used a solid wall boundary condition in the horizontal dimension and periodic boundary condition in the vertical dimension. Figure 5.10 provides vorticity and pressure contours for the vortex sheet problem at various times. There are two sources generating the acoustic waves in this example. The first source is the initial condition. The constant pressure field is perturbed by the vortex sheet at $t = 0$. The second source of acoustic wave generation is the motion of the vortex sheet. Figure 5.10 shows acoustic wave generation due to both of these sources. The important outcome is that the algorithm computes the dynamics of the acoustic waves stably and accurately for both sources.



(a) Vorticity at $t = 1.0$. **(b)** Pressure at $t = 1.0$.

(c) Vorticity at $t = 1.5$. **(d)** Pressure at $t = 1.5$.

(e) Vorticity at $t = 2.0$. **(f)** Pressure at $t = 2.0$.

Figure 5.10: Long Wavelength Vortex Sheet Results.

5.3 Viscous Effects

The code was also tested on the compressible Navier-Stokes system of equations. It is important to demonstrate the effectiveness of the algorithm on systems involving diffusive terms without combustion. The test problem selected for this was the periodic shear layer. All of the scalar variables were initially set to constants. The velocity field has the following form,

$$\boldsymbol{v}(\boldsymbol{x}) = \{ \tanh((x_1 - 0.25)/(\delta\rho)) \quad \text{if} x_1 \leq 0.5 \tag{5.15}$$

$$\tanh((0.75 - x_1)/(\delta\rho)) \quad \text{otherwise}, \tag{5.16}$$

$$\delta \sin(2\pi x_0)\}, \tag{5.17}$$

where $\delta\rho$ is 1/30 and δ is 0.05.

Figure 5.11 shows a selection of contour plots of the vorticity at different Reynolds numbers. We can see that as the viscosity is increased there is indeed a diffusion of the vorticity.

5.4 Combustion

We looked at several combustion problems to evaluate the efficacy of the algorithm.

0D Combustion

This test is designed to ensure that the algorithm can model reaction mechanisms with the combined ARK4 and CVODE integrators.

We modeled the combustion of methane in oxygen with the following initial conditions:

$$\boldsymbol{v}(\boldsymbol{x}) = [0, 0] \text{ m/s}, \tag{5.18}$$

$$\rho(\boldsymbol{x}) = 1.0 \text{ kg/m}^3, \tag{5.19}$$

$$p(\boldsymbol{x}) = 250 \text{ kPa}, \tag{5.20}$$

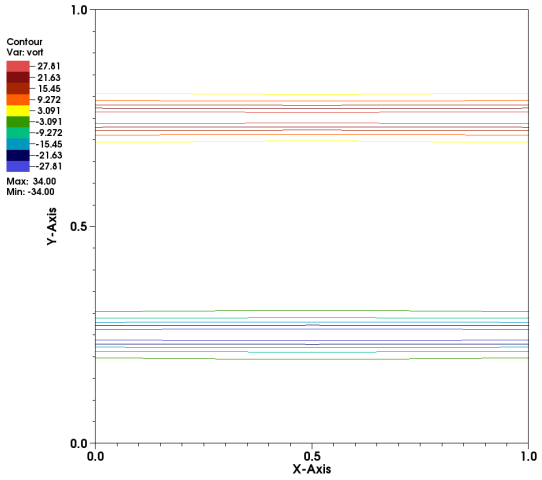
$$Y_{CH_4} = 0.21, \tag{5.21}$$

$$Y_{O_2} = 0.79, \tag{5.22}$$

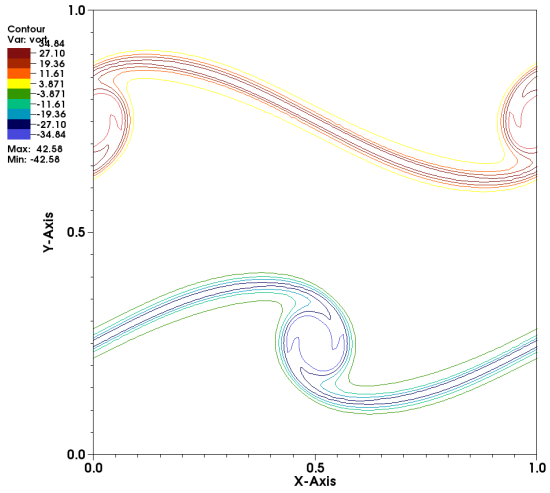
$$Y_{CO_2} = 0.0, \tag{5.23}$$

$$Y_{H_2O} = 0.0. \tag{5.24}$$

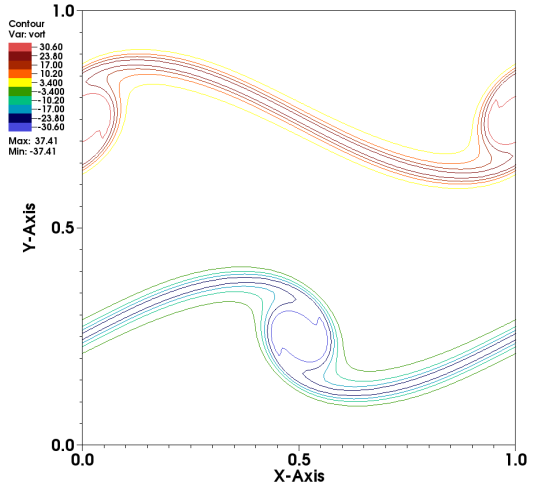
We expect that the reactants will be consumed and the products created over time until a steady state is reached. We also expect to see the pressure rising over time until a constant state is reached. Time plots for the mass fractions as well as the pressure are provided (Figure 5.12). We see the pressure slowly rising until it hits a critical value. At this point the combustion occurs rapidly.



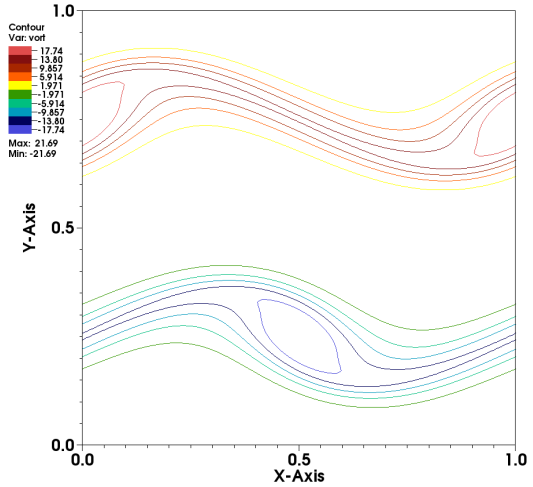
(a) Vorticity Initial Condition (w).



(b) Vorticity with $\eta = 0$.

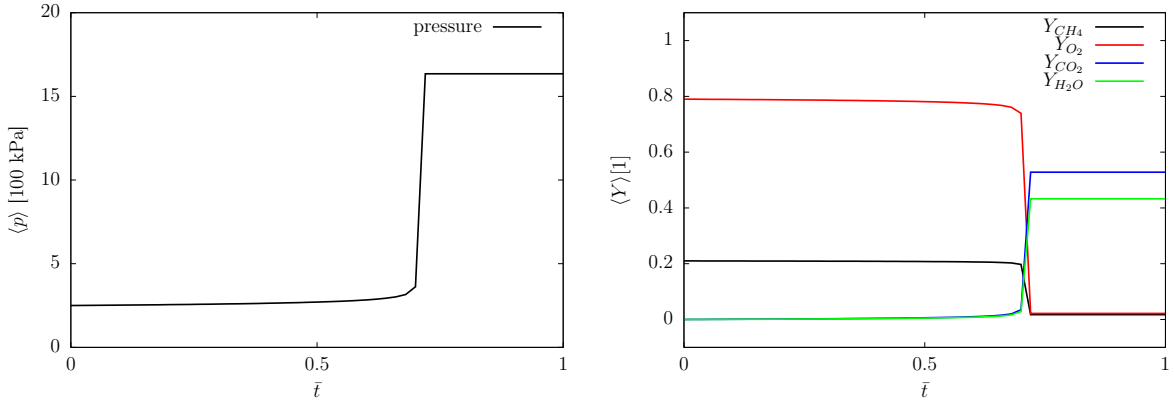


(c) Vorticity with $\eta = 1.e-4$.



(d) Vorticity with $\eta = 1.e-3$.

Figure 5.11: Shear Layer Vorticity at $t = 0.75$ for Varying Viscosities.



(a) 0D Pressure over Time.

(b) 0D Mass Fractions over Time.

Figure 5.12: 0D Combustion Results.

1D Flame

The next combustion test includes spatial variation, but only along a single dimension. For this test, we use a similar one-dimensional test as in [58]. The solution starts with products and reactants separated at the middle of the domain. The temperature of the products is high, and the temperature of the reactants is low. Here is the initial condition:

$$Y_{CH_4}(\boldsymbol{x}) = \begin{cases} 0.22 & \text{if } x_0 < 0.5, \\ 0 & \text{otherwise} \end{cases}, \tag{5.25}$$

$$Y_{O_2}(\boldsymbol{x}) = \begin{cases} 0.78 & \text{if } x_0 < 0.5, \\ 0 & \text{otherwise} \end{cases}, \tag{5.26}$$

$$Y_{CO_2}(\boldsymbol{x}) = \begin{cases} 0.52 & \text{if } x_0 > 0.5, \\ 0 & \text{otherwise} \end{cases}, \tag{5.27}$$

$$Y_{H_2O}(\boldsymbol{x}) = \begin{cases} 0.48 & \text{if } x_0 > 0.5, \\ 0 & \text{otherwise} \end{cases}, \tag{5.28}$$

$$T(\boldsymbol{x}) = \begin{cases} 300 & \text{if } x_0 < 0.5, \\ 1500 & \text{otherwise} \end{cases}, \tag{5.29}$$

$$\rho(\boldsymbol{x}) = 1.0, \tag{5.30}$$

$$\boldsymbol{v}(\boldsymbol{x}) = [0, 0]. \tag{5.31}$$

We should see a clear flame front after a period of time when the fuel and oxygen reach a high enough temperature that combustion may occur. Figure 5.13 contains a plot of the total pressure at the flame front after a time. In [58], it was noted that the thermodynamic pressure is uniform and scales as $\mathcal{O}(M^2)$. In our case the evolved pressure carries compressibility

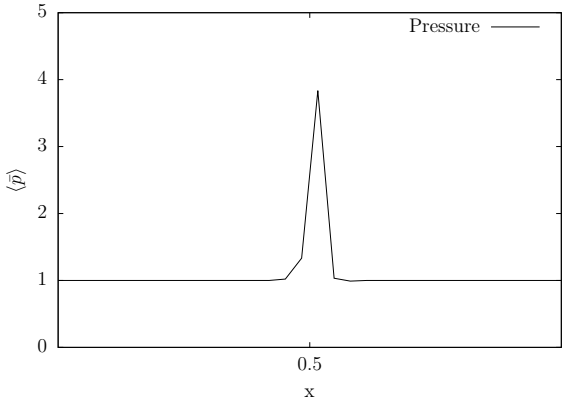


Figure 5.13: Pressure at the Flame Front for 1D Combustion Example.

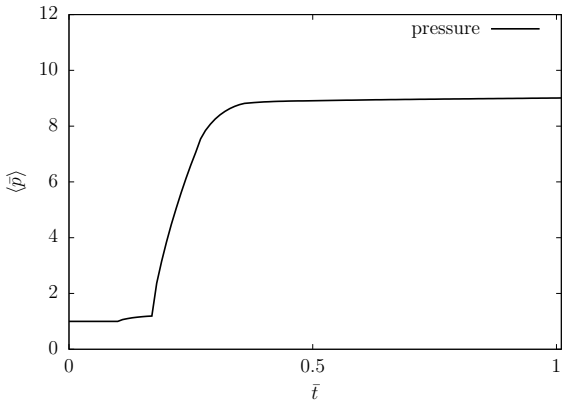


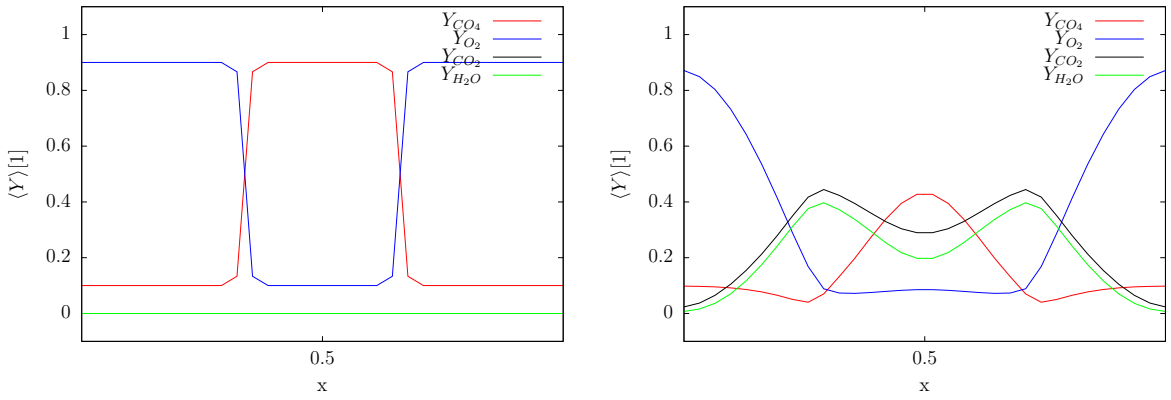
Figure 5.14: Pressure Near the Flame Front for 1D Combustion Example.

effects and generates the non-zero velocity at the flame front. But we still expect to see the pressure rise as the combustion occurs. Figure 5.14 confirms that the pressure is rising in this case.

2D Flame

This test case is similar to the 1D flame, except that we now have advection and diffusion in multiple dimensions. We modify that initial condition in the following way: we start off with nearly all of the fuel in a circular centered region of a 2D domain, and all of the oxygen outside of that circle. The fuel resides with a radius $R_0 <= 0.2$. The temperature in the oxygen-rich region is high, whereas it is low in the fuel-rich region. There are no products at the start.

Cross sections of the mass fractions at different times are show in Figure 5.15.



(a) $t = 0$. **(b)** $t = 1.0$.

Figure 5.15: Mass Fractions for 2D Flame.

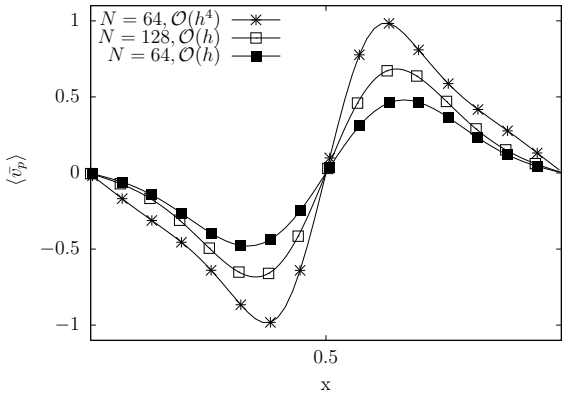


Figure 5.16: Time-Integration Order Effects on Compressible Velocity.

5.5 High-Order Accuracy

Time-Integration

One of the identified issues with the algorithm in [84] was that it used a first-order accurate time-integration scheme for the compressible variables. This low-order scheme produced poor results at finite values of M.

Introducing a fourth-order integration scheme for the compressible variables produces much better results. In Figure 5.16, we are comparing the first component of the curl-free velocity using the original first-order time-integration and the new fourth-order time-integration.

Table 5.1: Error in $\langle U \rangle$ for Inviscid Gas Dynamics [$\sigma = 0.01,\ L_\infty$].

Var	**N=16**	**Rate**	**N=32**	**Rate**	**N=64**	**Rate**	**N=128**
ρ	2.1e-06	3.97	1.33e-07	3.99	8.35e-09	4.00	5.23e-10
p	1.86e-06	3.97	1.19e-07	3.99	7.44e-09	4.00	4.66e-10
ρh	1.86e-06	3.97	1.19e-07	3.99	7.44e-09	4.00	4.66e-10
v^0	2.14e-05	3.98	1.36e-06	4.00	8.50e-08	4.00	5.32e-09
v^1	2.14e-05	3.98	1.36e-06	4.00	8.50e-08	4.00	5.32e-09
v_p^0	1.46e-06	3.80	1.05e-07	4.00	6.56e-09	4.10	3.82e-10
v_p^1	1.46e-06	3.80	1.05e-07	4.00	6.56e-09	4.10	3.82e-10
v_d^0	2.15e-05	3.98	1.36e-06	3.99	8.55e-08	4.00	5.35e-09
v_d^1	2.15e-05	3.98	1.36e-06	3.99	8.55e-08	4.00	5.35e-09

Table 5.2: Error in $\langle U \rangle$ for Inviscid Gas Dynamics [$\sigma = 0.5,\ L_\infty$].

Var	**N=16**	**Rate**	**N=32**	**Rate**	**N=64**	**Rate**	**N=128**
ρ	2.1e-05	3.74	1.61e-06	3.99	1.01e-07	4.00	6.69e-09
p	1.92e-05	3.74	1.43e-06	4.03	8.81e-08	4.01	5.47e-09
ρh	3.29e-05	3.72	2.49e-06	4.03	1.53e-07	4.01	9.50e-09
v^0	3.41e-05	3.87	2.33e-06	3.82	1.65e-07	3.96	1.06e-08
v^1	3.41e-05	3.87	2.33e-06	3.82	1.65e-07	3.96	1.06e-08
v_p^0	6.87e-05	3.77	5.03e-06	3.99	3.17e-07	3.99	2.00e-08
v_p^1	6.87e-05	3.77	5.03e-06	3.99	3.17e-07	3.99	2.00e-08
v_d^0	7.52e-05	3.76	5.55e-06	3.94	3.60e-07	3.98	2.28e-08
v_d^1	7.52e-05	3.76	5.55e-06	3.94	3.60e-07	3.98	2.28e-08

Convergence

We also provide errors and convergence rates. Both of these were tabulated using Richardson's error estimation.

First, let us examine the solution accuracy for small time-steps. Table 5.1 demonstrates the accuracy of the solution with $\sigma \sim 0.01$.

The test in Table 5.1 uses a small enough time step that we can safely assume that the leading-order error source is the spatial-differencing scheme. For each scalar and vector component, we see the expected fourth-order accuracy.

As we increase σ, we still retain fourth-order accuracy (Table 5.2).

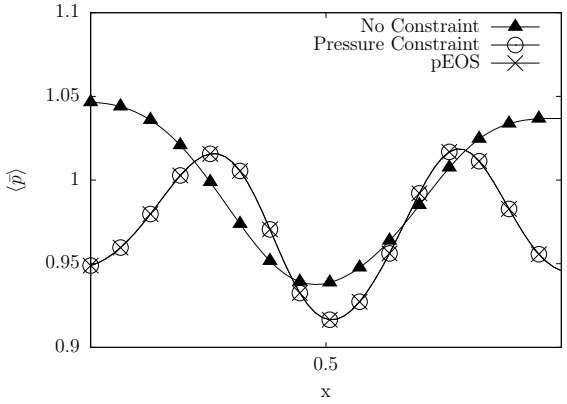


Figure 5.17: Effectiveness of Pressure Constraint along the Midline.

5.6 Enforcing Constraints

Pressure Constraint

The pressure constraint is required if the equation of the state is to be satisfied. Without the constraint, the evolved pressure will drift farther away from the equation of state pressure over time.

Here is the affect of this constraint for the single vortex test with $M \sim 0.28$ at $t = 0.5$ (Figure 5.17). The pressure constraint simulation used two iterations of the constraint solver per time advance. The data for this comparison were taken along the centerline parallel to the x-axis We see that the constraint is driving the evolved pressure towards the equation of state. In fact, the constrained evolved pressure is in nearly perfect alignment with the equation of state. Looking at the evolved pressures with and without the constraint, the phase and shape of the fields differ noticeably and the magnitudes are off by up to ten percent. These discrepancies are unacceptable in combustion since the equation of state plays a more prominent role in the simulation.

Looking at the effects of the constraint over time at a fixed location yields some additional insights (Figure 5.18). Again, we remark that the pressure field without the constraint is out of phase with the equation of state and the magnitude of the field is also diverging. With a single iteration of the pressure constraint solve we get much better agreement between the equation of state and the constrained pressure. With two iterations, the constrained pressure satisfies the equation of state.

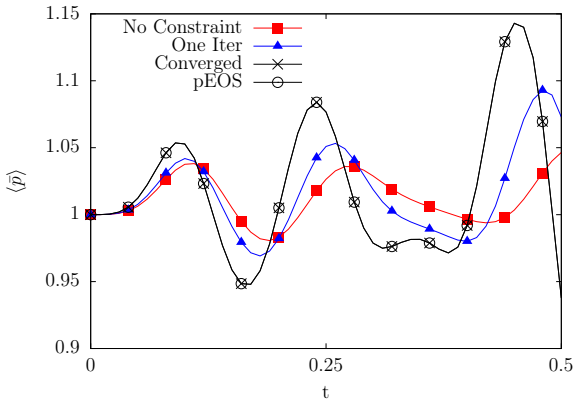


Figure 5.18: Effectiveness of Pressure Constraint over Time.

Chapter 6

Conclusions

The compressible pressure-velocity splitting proposed and developed in this thesis leads to an algorithm that works well across a variety of systems of equations and conditions. The algorithm effectiveness has been demonstrated for multiple settings ranging from inviscid gas dynamics to viscous combustion. We show that the new splitting algorithm produces identical results to the previous algorithm for inviscid gas dynamics, adequately represents baroclinic generation of vorticity, captures long wavelength acoustics, extends to viscous and combustion systems, and attains high-order accuracy.

There are three notable issues with the algorithm as presented: intra-stage synchronizations, lack of implemented robustness technologies, and lack of demonstrated AMR results.

There is at least one synchronization step required to advance the velocity at each stage. At the start of each stage computation, the total velocity is projected and interpolated onto cell faces. In some limited cases, we must also synchronize the total velocity and the potential velocity at the end of each stage computation. Ideally, any velocity synchronization and projection would be delayed until the end of the total time advance.

We did not implement some of the standard robustness technologies in this algorithm, such as flux-limiting. Furthermore, since not all of the modeled equations are in conservation form, additional care will have to be taken if one wanted to extend this algorithm to supersonic flows.

There are numerous next steps that can be taken. Embedded boundaries can be introduced to model flow in more complicated, static domains. Front-tracking can be used with embedded boundaries to model moving surfaces, such as a flat piston moving in a cylindrical domain. Mapped multi-block grids can be incorporated to model a non-flat piston. It would potentially be worth hybridizing the method with some sort of fully-conservative representation for high Mach number flows. More advanced chemistry models may be investigated as well. Alternative high-order discretizations may be considered, particularly for the time advance. We could also take the ideas developed here and apply them to other physics problems that are governed by similar evolution equations and constraints. Although no AMR results are presented herein, extending this algorithm to AMR is sensible.

Bibliography

[1] S. Abarbanel, P. Duth, and D. Gottlieb, "Splitting methods for low Mach number Euler and Navier-Stokes equations," *Computers & Fluids*, vol. 17, no. 1, pp. 1–12, 1989.

[2] T. Alazard, "Low Mach number limit of the full Navier-Stokes equations," *Archive for Rational Mechanics and Analysis*, vol. 180, no. 1, pp. 1–73, 2006.

[3] A. S. Almgren, J. B. Bell, C. A. Rendleman, and M. Zingale, "Low Mach number modeling of type Ia supernovae. I. Hydrodynamics," *The Astrophysical Journal*, vol. 637, no. 2, p. 922, 2006.

[4] A. S. Almgren, J. B. Bell, and W. G. Szymczak, "A numerical method for the incompressible Navier-Stokes equations based on an approximate projection," *SIAM Journal on Scientific Computing*, vol. 17, no. 2, pp. 358–369, 1996.

[5] ANSYS Fluent [Compute Software], ANSYS, Canonsburg, PA.

[6] J. B. Bell, M. S. Day, A. S. Almgren, M. J. Lijewski, and C. A. Rendleman, "A parallel adaptive projection method for low Mach number flows," *International Journal for Numerical Methods in Fluids*, vol. 40, no. 1-2, pp. 209–216, 2002.

[7] J. Bell, "AMR for low Mach number reacting flow," *Adaptive Mesh Refinement-Theory and Applications. Lecture Notes in Computational Science and Engineering*, vol. 41, Springer Berlin Heidelberg, 2005, pp. 203–221.

[8] W. R. Briley, L. K. Taylor, and D. L. Whitfield, "High-resolution viscous flow simulations at arbitrary Mach number," *Journal of Computational Physics*, vol. 184, no. 1, pp. 79–105, 2003.

[9] D. L. Brown, R. Cortez, and M. L. Minion, "Accurate projection methods for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 168, no. 2, pp. 464–499, 2001.

[10] L. S. Caretto, R. M. Curr, and D. B. Spalding, "Two numerical methods for three-dimensional boundary layers," *Computer Methods in Applied Mechanics and Engineering*, vol. 1, no. 1, pp. 39–57, 1972.

[11] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding, "Two calculation procedures for steady, three-dimensional flows with recirculation," *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, 1973, pp. 60–68.

[12] V. Casulli and D. Greenspan, "Pressure method for the numerical solution of transient, compressible fluid flows," *International Journal for Numerical Methods in Fluids*, vol. 4, no. 11, pp. 1001–1012, 1984.

[13] C. Chaplin and P. Colella, "A single-stage flux-corrected transport algorithm for high-order finite-volume methods," *Communications in Applied Mathematics and Computational Science*, vol. 12, no. 1, pp. 1–24, 2017.

[14] Z. Chen and A. Przekwas, "A coupled pressure-based computational method for incompressible/compressible flows," *Journal of Computational Physics*, vol. 229, no. 24, pp. 9150–9165, 2010.

[15] D. Choi and C. L. Merkle, "Application of time-iterative schemes to incompressible flow," *AIAA Journal*, vol. 23, no. 10, pp. 1518–1524, 1985.

[16] Y.-H. Choi and C. L. Merkle, "The application of preconditioning in viscous flows," *Journal of Computational Physics*, vol. 105, no. 2, pp. 207–223, 1993.

[17] A. J. Chorin, "A numerical method for solving viscous incompressible flow problems," *Journal of Computational Physics*, vol. 2, no. 1, pp. 12–26, 1967.

[18] ——, "Numerical solution of the Navier-Stokes equations," *Mathematics of Computation*, vol. 22, no. 104, pp. 745–762, 1968.

[19] W. K. Chow and Y. L. Cheung, "Comparison of the algorithms PISO and SIMPLER for solving pressure-velocity linked equations in simulating compartmental fire," *Numerical Heat Transfer, Part A: Applications*, vol. 31, no. 1, pp. 87–112, 1997.

[20] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin, "High-order, finite-volume methods in mapped coordinates," *Journal of Computational Physics*, vol. 230, no. 8, pp. 2952–2976, 2011.

[21] P. Colella, D. T. Graves, T. J. Ligocki, D. F. Martin, D. Modiano, D. B. Serafini, and B. Van Straalen, "Chombo software package for AMR applications - design document," Lawrence Berkeley National Laboratory, Technical Report LBNL-6616E, 2013.

[22] F. Cordier, P. Degond, and A. Kumbaro, "An asymptotic-preserving all-speed scheme for the Euler and Navier–Stokes equations," *Journal of Computational Physics*, vol. 231, no. 17, pp. 5685–5704, 2012.

[23] M. Darwish and F. Moukalled, "A fully coupled Navier-Stokes solver for fluid flow at all speeds," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 65, no. 5, pp. 410–444, 2014.

[24] M. Darwish, F. Moukalled, and B. Sekar, "A robust multi-grid pressure-based algorithm for multi-fluid flow at all speeds," *International Journal for Numerical Methods in Fluids*, vol. 41, no. 11, pp. 1221–1251, 2003.

[25] M. S. Day and J. B. Bell, "Numerical simulation of laminar reacting flows with complex chemistry," *Combustion Theory and Modelling*, vol. 4, no. 4, pp. 535–556, 2000.

[26] S. Dellacherie, "Analysis of Godunov type schemes applied to the compressible Euler system at low Mach number," *Journal of Computational Physics*, vol. 229, no. 4, pp. 978–1016, 2010.

[27] 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*, vol. 78, no. 5, pp. 461–471, 1999.

[28] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch, "High order conservative finite difference scheme for variable density low Mach number turbulent flows," *Journal of Computational Physics*, vol. 227, no. 15, pp. 7125–7159, 2008.

[29] J. Doom, Y. Hou, and K. Mahesh, "A numerical method for DNS/LES of turbulent reacting flows," *Journal of Computational Physics*, vol. 226, no. 1, pp. 1136–1151, 2007.

[30] A. Ern and V. Giovangigli, "Fast and accurate multicomponent transport property evaluation," *Journal of Computational Physics*, vol. 120, no. 1, pp. 105–116, 1995.

[31] C. Gatti-Bono and P. Colella, "An anelastic allspeed projection method for gravitationally stratified flows," *Journal of Computational Physics*, vol. 216, no. 2, pp. 589–615, 2006.

[32] B. E. Griffith, "An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7565–7595, 2009.

[33] J.-L. Guermond and A. Salgado, "A splitting method for incompressible flows with variable density based on a pressure Poisson equation," *Journal of Computational Physics*, vol. 228, no. 8, pp. 2834–2846, 2009.

[34] H. Guillard and C. Viozat, "On the behaviour of upwind schemes in the low Mach number limit," *Computers & Fluids*, vol. 28, no. 1, pp. 63–86, 1999.

[35] B. Gustafsson and H. Stoor, "Navier-Stokes equations for almost incompressible flow," *SIAM Journal on Numerical Analysis*, vol. 28, no. 6, pp. 1523–1547, 1991.

[36] J. Haack, S. Jin, and J. Liu, "An all-speed asymptotic-preserving method for the isentropic Euler and Navier-Stokes equations," *Communications in Computational Physics*, vol. 12, no. 4, pp. 955–980, 2012.

[37] N. Happenhofer, H. Grimm-Strele, F. Kupka, B. Löw-Baselli, and H. Muthsam, "A low Mach number solver: Enhancing applicability," *Journal of Computational Physics*, vol. 236, no. 1, pp. 96–118, 2013.

[38] F. H. Harlow and A. A. Amsden, "A numerical fluid dynamics calculation method for all flow speeds," *Journal of Computational Physics*, vol. 8, no. 2, pp. 197–213, 1971.

[39] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *Physics of Fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

[40] D. van der Heul, C. Vuik, and P. Wesseling, "A conservative pressure-correction method for flow at all speeds," *Computers & Fluids*, vol. 32, no. 8, pp. 1113–1132, 2003.

[41] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "Sundials: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software*, vol. 31, no. 3, pp. 363–396, 2005.

[42] R. I. Issa, "Solution of the implicitly discretised fluid flow equations by operator-splitting," *Journal of Computational Physics*, vol. 62, no. 1, pp. 40–65, 1986.

[43] ——, "The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme," *Journal of Computational Physics*, vol. 62, no. 1, pp. 66–82, 1986.

[44] R. I. Issa, B. Ahmadi-Befrui, K. R. Beshay, and A. D. Gosman, "Solution of the implicitly discretised reacting flow equations by operator-splitting," *Journal of Computational Physics*, vol. 93, no. 2, pp. 388–410, 1991.

[45] D. S. Jang, R. Jetli, and S. Acharya, "Comparison of the PISO, SIMPLER, and SIMPLEC algorithms for the treatment of the pressure-velocity coupling in steady flow problems," *Numerical Heat Transfer*, vol. 10, no. 3, pp. 209–228, 1986.

[46] H. Jing, L. Ru, H. Yaling, and Q. Zhiguo, "Solutions for variable density low Mach number flows with a compressible pressure-based algorithm," *Applied Thermal Engineering*, vol. 27, no. 11-12, pp. 2104–2112, 2007.

[47] S. Y. Kadioglu, R. Klein, and M. L. Minion, "A fourth-order auxiliary variable projection method for zero-Mach number gas dynamics," *Journal of Computational Physics*, vol. 227, no. 3, pp. 2012–2043, 2008.

[48] S. Y. Kadioglu and M. Sussman, "Adaptive solution techniques for simulating underwater explosions and implosions," *Journal of Computational Physics*, vol. 227, no. 3, pp. 2083–2104, 2008.

[49] S. Y. Kadioglu, M. Sussman, S. Osher, J. P. Wright, and M. Kang, "A second order primitive preconditioner for solving all speed multi-phase flows," *Journal of Computational Physics*, vol. 209, no. 2, pp. 477–503, 2005.

[50] C. A. Kennedy and M. H. Carpenter, "Additive Runge-Kutta schemes for convection-diffusion-reaction equations," *Applied Numerical Mathematics*, vol. 44, no. 1, pp. 139–181, 2003.

[51] I. J. Keshtiban, F. Belblidia, and M. F. Webster, "Compressible flow solvers for low Mach number flows - a review," Institute of Non-Newtonian Fluid Mechanics, Department of Computer Science, University of Wales, Swansea, SA2 8PP, UK., Tech. Rep., 2003.

[52] S. Klainerman and A. Majda, "Compressible and incompressible fluids," *Communications on Pure and Applied Mathematics*, vol. 35, no. 5, pp. 629–651, 1982.

[53] R. Klein, "Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I: One-dimensional flow," *Journal of Computational Physics*, vol. 121, no. 2, pp. 213–237, 1995.

[54] R. Klein, N. Botta, T. Schneider, C.-D. Munz, S. Roller, A. Meister, L. Hoffmann, and T. Sonar, "Asymptotic adaptive methods for multi-scale problems in fluid mechanics," *Journal of Engineering Mathematics*, vol. 39, no. 1, pp. 261–343, 2001.

[55] O. M. Knio, H. N. Najm, and P. S. Wyckoff, "A semi-implicit numerical scheme for reacting flow: II. Stiff, operator-split formulation," *Journal of Computational Physics*, vol. 154, no. 2, pp. 428–467, 1999.

[56] 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*, vol. 12, no. 2, pp. 187–214, 1991.

[57] N. Kwatra, J. Su, J. T. Grétarsson, and R. Fedkiw, "A method for avoiding the acoustic time step restriction in compressible flow," *Journal of Computational Physics*, vol. 228, no. 11, pp. 4146–4161, 2009.

[58] M. F. Lai, "A projection method for reacting flow in the zero Mach number limit," PhD thesis, University of California, Berkeley, 1994.

[59] M. F. Lai, J. B. Bell, and P. Colella, "A projection method for combustion in the zero Mach number limit," *Proceedings of the Eleventh AIAA Computational Fluid Dynamics Conference*, 1993, pp. 776–783.

[60] D. Lee, "The design of local Navier-Stokes preconditioning for compressible flow," *Journal of Computational Physics*, vol. 144, no. 2, pp. 460–483, 1998.

[61] X.-S. Li and C.-W. Gu, "An all-speed Roe-type scheme and its asymptotic analysis of low Mach number behaviour," *Journal of Computational Physics*, vol. 227, no. 10, pp. 5144–5159, 2008.

[62] M. J. Lighthill, "On sound generated aerodynamically I. General theory," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 211, no. 1107, pp. 564–587, 1952.

[63] M.-S. Liou, "A sequel to AUSM, Part II: AUSM+-up for all speeds," *Journal of Computational Physics*, vol. 214, no. 1, pp. 137–170, 2006.

[64] K. Liu and R. H. Pletcher, "A fractional step method for solving the compressible Navier–Stokes equations," *Journal of Computational Physics*, vol. 226, no. 2, pp. 1930–1951, 2007.

[65] A. Majda and J. Sethian, "The derivation and numerical solution of the equations for zero Mach number combustion," *Combustion Science and Technology*, vol. 42, no. 3-4, pp. 185–205, 1985.

[66] D. F. Martin and P. Colella, "A cell-centered adaptive projection method for the incompressible Euler equations," *Journal of Computational Physics*, vol. 163, no. 2, pp. 271–312, 2000.

[67] D. F. Martin, P. Colella, and D. Graves, "A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions," *Journal of Computational Physics*, vol. 227, no. 3, pp. 1863–1886, 2008.

[68] D. F. Martin and K. L. Cartwright, "Solving Poisson's equation using adaptive mesh refinement," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M96/66, 1996.

[69] I. Mary, P. Sagaut, and M. Deville, "An algorithm for low Mach number unsteady flows," *Computers & Fluids*, vol. 29, no. 2, pp. 119–147, 2000.

[70] ——, "An algorithm for unsteady viscous flows at all speeds," *International Journal for Numerical Methods in Fluids*, vol. 34, no. 5, pp. 371–401, 2000.

[71] P. McCorquodale and P. Colella, "A high-order finite-volume method for conservation laws on locally refined grids," *Communications in Applied Mathematics and Computational Science*, vol. 6, no. 1, pp. 1–25, 2011.

[72] A. Meister, "Asymptotic based preconditioning technique for low Mach number flows," *ZAMM-Journal of Applied Mathematics and Mechanics*, vol. 83, no. 1, pp. 3–25, 2003.

[73] A. G. Merzhanov, W. A. Sirignano, and L. De Luca, "Numerical simulation of unsteady combustion," *Advances in Combustion Science: In Honor of Ya. B. Zeldovich*, American Institute of Aeronautics and Astronautics, 1997, pp. 179–194.

[74] F. Miczek, F. K. Röpke, and P. V. F. Edelmann, "New numerical solver for flows at various Mach numbers," *Astronomy & Astrophysics*, vol. 576, no. A50, pp. 1–16, 2015.

[75] M. L. Minion, "A projection method for locally refined grids," *Journal of Computational Physics*, vol. 127, no. 1, pp. 158–178, 1996.

[76] E. Motheau and J. Abraham, "A high-order numerical algorithm for DNS of low-Mach-number reactive flows with detailed chemistry and quasi-spectral accuracy," *Journal of Computational Physics*, vol. 313, pp. 430–454, 2016.

[77] F. Moukalled and M. Darwish, "A high-resolution pressure-based algorithm for fluid flow at all speeds," *Journal of Computational Physics*, vol. 168, no. 1, pp. 101–130, 2001.

[78] B. Müller, "Low-Mach-number asymptotics of the Navier-Stokes equations," *Journal of Engineering Mathematics*, vol. 34, no. 1-2, pp. 97–109, 1998.

[79] C.-D. Munz, S. Roller, R. Klein, and K. J. Geratz, "The extension of incompressible flow solvers to the weakly compressible regime," *Computers & Fluids*, vol. 32, no. 2, pp. 173–196, 2003.

[80] H. N. Najm, P. S. Wyckoff, and O. M. Knio, "A semi-implicit numerical scheme for reacting flow I. Stiff chemistry," *Journal of Computational Physics*, vol. 143, no. 2, pp. 381–402, 1998.

[81] F. Nicoud, "Conservative high-order finite-difference schemes for low-Mach number flows," *Journal of Computational Physics*, vol. 158, no. 1, pp. 71–97, 2000.

[82] J. Oliveira and R. I. Issa, "An improved PISO algorithm for the computation of buoyancy-driven flows," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 40, no. 6, pp. 473–493, 2001.

[83] K. Pao and C. Anderson, "Computation of flow noise in two dimensions," Los Alamos National Laboratory, Technical Report LA-UR-94-1354, 1994.

[84] K. Pao and P. Colella, "A projection method for low speed flows," *Journal of Computational Physics*, vol. 149, no. 2, pp. 245–269, 1999.

[85] S. Paolucci, "Filtering of sound from the Navier-Stokes equations," Sandia National Labs., Livermore, CA (USA), Technical Report SAND-82-8257, 1982.

[86] F. Paravento, "A robust numerical model for premixed flames with high density ratios based on new pressure correction and IMEX schemes," *Journal of Computational Physics*, vol. 229, no. 12, pp. 4613–4647, 2010.

[87] H. Park, R. R. Nourgaliev, R. C. Martineau, and D. A. Knoll, "On physics-based preconditioning of the Navier-Stokes equations," *Journal of Computational Physics*, vol. 228, no. 24, pp. 9131–9146, 2009.

[88] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. New York: Hemisphere Publishing Corporation, 1980.

[89] S. V. Patankar and D. B. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International Journal of Heat and Mass Transfer*, vol. 15, no. 10, pp. 1787–1806, 1972.

[90] G. Patnaik, R. H. Guirguis, J. P. Boris, and E. S. Oran, "A barely implicit correction for flux-corrected transport," *Journal of Computational Physics*, vol. 71, no. 1, pp. 1–20, 1987.

[91] W. Pazner, A. Nonaka, J. B. Bell, M. S. Day, and M. L. Minion, "A high-order spectral deffered correction strategy for low Mach number flow with complex chemistry," *Combustion Theory and Modelling*, vol. 20, no. 3, pp. 521–547, 2016.

[92] R. Pember, L. Howell, J. B. Bell, P. Colella, W. Crutchfield, W. Fiveland, and J. Jessee, "An adaptive projection method for unsteady, low-Mach number combustion," *Combustion Science and Technology*, vol. 140, no. 1, pp. 123–168, 1998.

[93] R. B. Pember, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. F. Lai, "A higher-order projection method for the simulation of unsteady turbulent nonpremixed combustion in an industrial burner," *In Proceedings of the 8th International Symposium on Transport Phenomena in Combustion*, 1995, pp. 1200–1211.

[94] Y. Penel, S. Dellacherie, and B. Després, "Coupling strategies for compressible-low Mach number flows," *Mathematical Models and Methods in Applied Sciences*, vol. 25, no. 6, pp. 1045–1089, 2015.

[95] A. Powell, "Theory of vortex sound," *The Journal of the Acoustical Society of America*, vol. 36, no. 1, pp. 177–195, 1964.

[96] P. Rauwoens, J. Vierendeels, E. Dick, and B. Merci, "A conservative pressurecorrection scheme for transient simulations of reacting flows," *Journal of Computational and Applied Mathematics*, vol. 234, no. 7, pp. 2311–2318, 2010.

[97] R. Rehm and H. Baum, "The equations of motion for thermally buoyant flows," *Journal of Research of the National Bureau of Standards*, vol. 83, no. 3, pp. 297–308, 1978.

[98] F. Rieper, "A low-Mach number fix for Roe's approximate Riemann solver," *Journal of Computational Physics*, vol. 230, no. 13, pp. 5263–5287, 2011.

[99] S. Roller and C.-D. Munz, "A low Mach number scheme based on multi-scale asymptotics," *Computing and Visualization in Science*, vol. 3, no. 1-2, pp. 85–91, 2000.

[100] Ronald Fedkiw, "A survey of chemically reacting, compressible flows," PhD thesis, University of California, Los Angeles, 1997.

[101] C.-C. Rossow, "Efficient computation of compressible and incompressible flows," *Journal of Computational Physics*, vol. 220, no. 2, pp. 879–899, 2007.

[102] C. Safta, J. Ray, and H. N. Najm, "A high-order low-Mach number AMR construction for chemically reacting flows," *Journal of Computational Physics*, vol. 229, no. 24, pp. 9299–9322, 2010.

[103] T. Schneider, N. Botta, K. Geratz, and R. Klein, "Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flows," *Journal of Computational Physics*, vol. 155, no. 2, pp. 248–286, 1999.

[104] S. Schochet, "The mathematical theory of low Mach number flows," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 39, no. 3, pp. 441–458, 2005.

[105] P. K. Senecal, K. J. Richards, E. Pomraning, T. Yang, M. Z. Dai, R. M. McDavid, M. A. Patterson, S. Hou, and T. Shethaji, "A new parallel cut-cell cartesian CFD code for rapid grid generation applied to in-cylinder diesel engine simulations," *SAE Technical Paper*, SAE International, 2007, pp. 1–32.

[106] J. Sesterhenn, B. Müller, and H. Thomann, "Flux-vector splitting for compressible low mach number flow," *Computers & Fluids*, vol. 22, no. 4, pp. 441–451, 1993.

[107] E. A. Sewall and D. K. Tafti, "A time-accurate variable property algorithm for calculating flows with large temperature variations," *Computers & Fluids*, vol. 37, no. 1, pp. 51–63, 2008.

[108] E. Shima and K. Kitamura, "Parameter-free simple low-dissipation AUSM-family scheme for all speeds," *AIAA Journal*, vol. 49, no. 8, pp. 1693–1709, 2011.

[109] K. S. Shterev and S. K. Stefanov, "Pressure based finite volume method for calculation of compressible viscous gas flows," *Journal of Computational Physics*, vol. 229, no. 2, pp. 461–480, 2010.

[110] J. S. Shuen, K. H. Chen, and Y. H. Choi, "A time-accurate algorithm for chemical non-equilibrium viscous flows at all speeds," *28th Joint Propulsion Conference and Exhibit, Joint Propulsion Conferences*, 1992, pp. 1–14.

[111] G. I. Sivashinsky, "Hydrodynamic theory of flame propagation in an enclosed volume," *Acta Astronautica*, vol. 6, no. 5-6, pp. 631–645, 1979.

[112] D.-L. Sun, Z.-G. Qu, Y.-L. He, and W.-Q. Tao, "Performance analysis of IDEAL algorithm for three-dimensional incompressible fluid flow and heat transfer problems," *International Journal for Numerical Methods in Fluids*, vol. 61, no. 10, pp. 1132–1160, 2009.

[113] A. G. Tomboulides, J. C. Y. Lee, and S. A. Orszag, "Numerical simulation of low Mach number reactive flows," *Journal of Scientific Computing*, vol. 12, no. 2, pp. 139–167, 1997.

[114] E. Turkel, "Preconditioning techniques in computational fluid dynamics," *Annual Review of Fluid Mechanics*, vol. 31, no. 1, pp. 385–416, 1999.

[115] E. Turkel, "Preconditioned methods for solving the incompressible and low speed compressible equations," *Journal of Computational Physics*, vol. 72, no. 2, pp. 277–298, 1987.

[116] C. Wall, C. D. Pierce, and P. Moin, "A semi-implicit method for resolution of acoustic waves in low Mach number flows," *Journal of Computational Physics*, vol. 181, no. 2, pp. 545–563, 2002.

[117] J. M. Weiss and W. A. Smith, "Preconditioning applied to variable and constant density flows," *AIAA Journal*, vol. 33, no. 11, pp. 2050–2057, 1995.

[118] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, vol. 12, no. 6, pp. 620–631, 1998.

[119] P. Wesseling, "Unified methods for computing incompressible and compressible flow," *Principles of Computational Fluid Dynamics*, vol. 29, Berlin, Heidelberg: Springer, 2009, pp. 567–601.

[120] F. White, *Viscous Fluid Flow*, 3rd ed. McGraw-Hill Education, 2005.

[121] P. T. Williams and A. J. Baker, "Incompressible computational fluid dynamics and the continuity constraint method for the three-dimensional Navier-Stokes equations," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 29, no. 2, pp. 137–273, 1996.

[122] F. Xiao, "Unified formulation for compressible and incompressible flows by using multi-integrated moments I: One-dimensional inviscid compressible flow," *Journal of Computational Physics*, vol. 195, no. 2, pp. 629–654, 2004.

[123] F. Xiao, R. Akoh, and S. Ii, "Unified formulation for compressible and incompressible flows by using multi-integrated moments II: Multi-dimensional version for compressible and incompressible flows," *Journal of Computational Physics*, vol. 213, no. 1, pp. 31–56, 2006.

[124] T. Yabe and P.-Y. Wang, "Unified numerical procedure for compressible and incompressible fluid," *Journal of the Physical Society of Japan*, vol. 60, no. 7, pp. 2105–2108, 1991.

[125] S. Y. Yoon and T. Yabe, "The unified simulation for incompressible and compressible flow by the predictor-corrector scheme based on the CIP method," *Computer Physics Communications*, vol. 119, no. 2-3, pp. 149–158, 1999.

[126] R. Yu, J. Yu, and X.-S. Bai, "An improved high-order scheme for DNS of low Mach number turbulent reacting flows based on stiff chemistry solver," *Journal of Computational Physics*, vol. 231, no. 16, pp. 5504–5521, 2012.

Appendix A

Variables

Here is a complete list of variables, symbols, and units for reference below:

Symbol	Variable	Units
N_s	Total Species	[1]
R_u	Gas Constant	$ML^2/(T^2 mol\Theta)$
η	Shear Viscosity	M/LT
κ	Volume Viscosity	M/LT
$\mathcal{D}_k$	Approximate Diffusion Vector	L^2/T
$\mathcal{D}_{ij}$	Binary Diffusion Matrix	L^2/T
λ	Thermal Conductivity	$ML/(T^3\Theta)$
ρ	Total Density	M/L^3
p	Total Pressure	M/LT^2
T	Temperature	Θ
H	Enthalpy	$1/(LT^2)$
e	Specific Energy	L^2/T^2
M_k	Molar Mass	M/mol
Y_k	Mass Fraction	[1]
X_k	Mole Fraction	[1]
h_k	Specific Enthalpy	L^2/T^2
h_k^{st}	Enthalpy of Formation	L^2/T^2
C_p	Specific Heat at Constant p	$L^2/(\Theta T^2)$
C_v	Specific Heat at Constant v	$L^2/(\Theta T^2)$
γ	Ratio of Specific Heats	[1]
$\dot{\omega}_k$	Combustion Generation Rate	M/L^3T
$\dot{\rho}_k^s$	Spray Generation Rate	M/L^3T
$\dot{\rho}^s$	Total Spray Generation Rate	M/L^3T
$\dot{Q}^s$	Spray Energy Source	$M/(LT^3)$
Φ	Viscous Dissipation Term	$M/(LT^3)$

$\boldsymbol{v}$	Bulk Velocity	L/T
$\boldsymbol{V}_k$	Diffusion Velocity	L/T
$\boldsymbol{F}^b$	Body Force	L/T^2
$\boldsymbol{F}^s$	Spray Momentum Source	L/T^2
$\boldsymbol{q}$	Heat Flux	M/T^3
$\boldsymbol{\tau}$	Shear Stress Tensor	M/LT^2
$\boldsymbol{e}$	Strain-Rate Tensor	$[1]/T$

Variables with subscript "k" are specific to their particular gaseous species. Theta "Θ" denotes absolute temperature (Kelvin), and "(M, L, T, mol)" denote mass, length, time, and moles respectively.

Appendix B

Governing Equations

Gas-Phase Combustion

The conservation equations that govern multicomponent reacting systems for gas-phase combustion are provided below:

$$\frac{\partial(\rho Y_k)}{\partial t} + \nabla \cdot (\rho Y_k \boldsymbol{v}) = \nabla \cdot (\rho \mathcal{D}_k \nabla Y_k) + \dot{\omega}_k + \dot{\rho}_k^s \tag{B.1}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = \dot{\rho}^s \tag{B.2}$$

$$\frac{\partial(\rho \boldsymbol{v})}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} \otimes \boldsymbol{v}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{F}^b + \boldsymbol{F}^s \tag{B.3}$$

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho e \boldsymbol{v}) = -p[\nabla \cdot \boldsymbol{v}] + \Phi + \nabla \cdot \boldsymbol{q} \tag{B.4}$$

$$p = \rho R_u T \sum_{k=1}^{N_s} \frac{Y_k}{M_k} \tag{B.5}$$

$$\boldsymbol{\tau} = 2\eta \boldsymbol{e} + \left(\kappa - \frac{2}{3}\eta\right)(\nabla \cdot \boldsymbol{v}) \tag{B.6}$$

$$\boldsymbol{e} = \frac{1}{2}\left(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T\right) \tag{B.7}$$

$$\Phi = 2\eta(\boldsymbol{e} : \boldsymbol{e}) + \left(\kappa - \frac{2}{3}\eta\right)(\nabla \cdot \boldsymbol{v})^2 \tag{B.8}$$

$$\boldsymbol{q} = q^E + q^I \tag{B.9}$$

$$q^E = \sum_{k=1}^{NS} \left(\frac{\lambda}{c_p} - \rho D_k\right) h_k \nabla Y_k \tag{B.10}$$

$$q^I = \frac{\lambda}{c_p} \nabla h \tag{B.11}$$

Assumptions

We are going to ignore the contributions from spray sources and the body force. So all terms with superscript "s" ($\dot{\rho}_k^s$,$\dot{\rho}^s$,$\boldsymbol{F}^s$) as well as $\boldsymbol{F}^b$ are dropped moving forward. Soret and Dufour effects have been neglected. Energy sources from reaction rates are included in the definition of "e", hence we are evolving energy and not "sensible" energy.

A description of how to compute the reaction terms is provided in Appendix C. A description of how to compute the fluid properties is provided in Appendix D.

Alternative Forms of Governing Equations

We described the system of PDE initially in terms of the specific energy e and the total momentum $\rho\boldsymbol{v}$. From these we can deduce additional evolution equations. First we will define the material derivative for compactness of notation

$$\frac{D(a)}{Dt} = \frac{\partial(a)}{\partial t} + (\boldsymbol{v}\cdot\nabla)(a). \tag{B.12}$$

The alternative evolution equations are:

Velocity

$$\frac{D\boldsymbol{v}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}(\nabla\cdot\boldsymbol{\tau}). \tag{B.13}$$

Enthalpy

$$h = e + \frac{p}{\rho} \tag{B.14}$$

$$\frac{Dh}{Dt} = \frac{De}{Dt} + \frac{D(p/\rho)}{Dt} \tag{B.15}$$

$$= \frac{De}{Dt} + \frac{1}{\rho}\frac{Dp}{Dt} + \frac{p}{\rho}(\nabla\cdot\boldsymbol{v}) \tag{B.16}$$

$$\rho\frac{Dh}{Dt} = \rho\frac{De}{Dt} + \frac{Dp}{Dt} + p(\nabla\cdot\boldsymbol{v}) \tag{B.17}$$

$$= \frac{Dp}{Dt} + \Phi + \nabla\cdot\boldsymbol{q}. \tag{B.18}$$

We'll call the contribution of the pressure derivative on the right hand side Ψ so that,

$$\rho\frac{Dh}{Dt} = \Psi + \Phi + \nabla\cdot\boldsymbol{q}. \tag{B.19}$$

Total Enthalpy

$$H = h + \frac{\boldsymbol{v}\cdot\boldsymbol{v}}{2} \tag{B.20}$$

$$\rho\frac{DH}{Dt} = \rho\frac{Dh}{Dt} + \rho\frac{D[(\boldsymbol{v}\cdot\boldsymbol{v})/2]}{Dt} \tag{B.21}$$

$$= \frac{\partial p}{\partial t} + \nabla\cdot(\boldsymbol{v}\boldsymbol{\tau}) + \nabla\cdot\boldsymbol{q}. \tag{B.22}$$

Temperature

$$h_k(T) = h_k^{st} + \int_{T_0}^{T} C_{p,k}(T)dT \tag{B.23}$$

$$h = \sum_{k=1}^{N_s} Y_k h_k \tag{B.24}$$

$$c_p = \sum_{k=1}^{N_s} Y_k \frac{dh_k}{dT} \tag{B.25}$$

$$\rho\frac{D\left(\sum_{k=1}^{N_s} Y_k h_k\right)}{Dt} = \Psi + \Phi + \nabla\cdot\boldsymbol{q} \tag{B.26}$$

$$\sum_{k=1}^{N_s}\rho\left[Y_k\frac{Dh_k}{Dt} + h_k\frac{DY_k}{Dt}\right] = \Psi + \Phi + \nabla\cdot\boldsymbol{q} \tag{B.27}$$

$$\sum_{k=1}^{N_s}\left[\rho Y_k c_{p,k}\frac{DT}{Dt} + \rho h_k\frac{DY_k}{Dt}\right] = \Psi + \Phi + \nabla\cdot\boldsymbol{q} \tag{B.28}$$

$$\rho\frac{DT}{Dt}\sum_{k=1}^{N_s} c_{p,k}Y_k + \sum_{k=1}^{N_s} h_k\rho\frac{DY_k}{Dt} = \Psi + \Phi + \nabla\cdot\boldsymbol{q} \tag{B.29}$$

$$\rho c_p\frac{DT}{Dt} + \sum_{k=1}^{N_s} h_k\left(\nabla\cdot\rho D_k\nabla Y_k + \dot{\omega}_k\right) = \Psi + \Phi + \nabla\cdot\boldsymbol{q} \tag{B.30}$$

$$\rho c_p\frac{DT}{Dt} = \Psi + \Phi + \nabla\cdot\lambda\nabla T - \sum_{k=1}^{N_s}\left[h_k\left(\nabla\cdot\rho D_k\nabla Y_k + \dot{\omega}_k\right) - \nabla\cdot\rho D_k h_k\nabla Y_k\right] \tag{B.31}$$

$$\left(\rho c_p\right)\frac{DT}{Dt} = \Psi + \Phi + \nabla\cdot(\lambda\nabla T) - \sum_{k=1}^{N_s} h_k\dot{\omega}_k + \sum_{k=1}^{N_s}\left[\rho D_k\nabla Y_k\cdot\nabla h_k\right] \tag{B.32}$$

$$\rho c_p\frac{DT}{Dt} = \Psi+\Phi+\nabla\cdot\frac{\lambda}{c_p}\nabla h-\sum_{k=1}^{N_s}\left[h_k\left(\nabla\cdot\rho D_k\nabla Y_k + \dot{\omega}_k\right) - \nabla\cdot\left(\rho D_k + \frac{\lambda}{c_p}\right)h_k\nabla Y_k\right] \tag{B.33}$$

$$\rho c_p \frac{DT}{Dt} = \Psi + \Phi + \nabla \cdot \frac{\lambda}{c_p} \nabla h - \sum_{k=1}^{N_s} \left[h_k \dot{\omega}_k + \xi(Y_k) \right], \tag{B.34}$$

where we have introduced the following expression,

$$\xi(Y_k) = -\nabla \cdot \left(\rho D_k + \frac{\lambda}{c_p} \right) h_k \nabla Y_k + h_k \nabla \cdot \rho D_k \nabla Y_k. \tag{B.35}$$

Pressure (Single-Species Ideal Gas)

$$p = \rho R T \tag{B.36}$$

$$\frac{Dp}{Dt} = \left. \frac{\partial p}{\partial \rho} \right|_T \frac{D\rho}{Dt} + \left. \frac{\partial p}{\partial T} \right|_\rho \frac{DT}{dt} \tag{B.37}$$

$$\frac{Dp}{Dt} = RT \frac{D\rho}{Dt} + \rho R \frac{DT}{Dt}. \tag{B.38}$$

$$\frac{Dp}{Dt} = -\rho R T \left(\nabla \cdot \boldsymbol{v} \right) + \frac{\rho R}{\rho c_p} \left[\frac{Dp}{Dt} + \Phi + \nabla \cdot \boldsymbol{q} \right] \tag{B.39}$$

$$\left(1 - \frac{R}{c_p} \right) \frac{Dp}{Dt} = -p \left(\nabla \cdot \boldsymbol{v} \right) + \frac{R_\mu}{c_p} \left[\Phi + \nabla \cdot \boldsymbol{q} \right]. \tag{B.40}$$

$$\frac{Dp}{Dt} = -\gamma p \left(\nabla \cdot \boldsymbol{v} \right) + (\gamma - 1) \left[\Phi + \nabla \cdot \boldsymbol{q} \right]. \tag{B.41}$$

Pressure (Multi-Species Ideal Gas with Combustion)

$$p = \rho R_\mu T \sum_k \frac{Y_k}{W_k}. \tag{B.42}$$

$$\frac{Dp}{Dt} = \left. \frac{\partial p}{\partial \rho} \right|_{T,Y_k} \frac{D\rho}{Dt} + \left. \frac{\partial p}{\partial T} \right|_{\rho,Y_k} \frac{DT}{dt} + \left. \frac{\partial p}{\partial Y_k} \right|_{\rho,T} \frac{DY_k}{Dt}, \tag{B.43}$$

$$\frac{Dp}{Dt} = R_\mu T \sum_k \frac{Y_k}{W_k} \frac{D\rho}{Dt} + \rho R_\mu \sum_k \frac{Y_k}{W_k} \frac{DT}{Dt} + \rho R_\mu T \sum_k \frac{1}{W_k} \frac{DY_k}{Dt}. \tag{B.44}$$

$$\begin{aligned} \frac{Dp}{Dt} = & -\rho R T \sum_{k=1}^{N_s} \frac{Y_k}{W_k} \left(\nabla \cdot \boldsymbol{v} \right) + \rho R_\mu \sum_{k=1}^{N_s} \frac{Y_k}{W_k} \frac{1}{\rho c_p} \left[\Psi + \Phi + \nabla \cdot \frac{\lambda}{c_p} \nabla h - \sum_{k=1}^{N_s} \left[h_k \dot{\omega}_k + \xi(Y_k) \right] \right] \\ & + \rho R_\mu T \sum_{k=1}^{N_s} \frac{1}{W_k} \frac{1}{\rho} \left[\nabla \cdot \rho D_k \nabla Y_k + \dot{\omega}_k \right]. \end{aligned} \tag{B.45}$$

Since $\Psi = \frac{Dp}{Dt}$, we must gather the the pressure derivative terms to produce,

$$\left[1-\sum_k \frac{R_\mu Y_k}{c_p W_k}\right]\frac{Dp}{Dt} = -\,p\,(\nabla\cdot\boldsymbol{v}) + \left(\sum_k \frac{R_\mu Y_k}{c_p W_k}\right)\left[\Phi + \nabla\cdot\frac{\lambda}{c_p}\nabla h - \sum_{k=1}^{N_s}\left[h_k\dot{\omega}_k + \xi(Y_k)\right]\right] + \sum_k \frac{R_\mu T}{W_k}\left[(\nabla\cdot\rho D_k\nabla Y_k) + \dot{\omega}_k\right].$$

We will reduce this expression to,

$$\frac{Dp}{Dt} = \dot{p}^E + \dot{p}^I - \alpha\sum_{k=1}^{N_s}\left[(\beta h_k + \zeta_k)\,\dot{\omega}_k\right], \tag{B.46}$$

where

$$\dot{p}^E = \alpha\sum_{k=1}^{N_s}\left[\beta\xi(Y_k) + \zeta_k(\nabla\cdot\rho D_k\nabla Y_k)\right] + \alpha\beta\Phi \tag{B.47}$$

$$\dot{p}^I = \alpha\beta\left(\nabla\cdot q^I\right) - \alpha p\,(\nabla\cdot\boldsymbol{v}) \tag{B.48}$$

$$\beta = \sum_{k=1}^{N_s}\left(\frac{R_\mu Y_k}{c_p W_k}\right) \tag{B.49}$$

$$\frac{1}{\alpha} = [1-\beta] \tag{B.50}$$

$$\zeta_k = \left(\frac{R_\mu T}{W_k}\right). \tag{B.51}$$

We may elect to use any combination of these or the previous evolution and state equations. However our system will be overdetermined if we attempt to evolve all of these equations independently of their relational expressions and the equation of state.

Appendix C

Combustion Chemistry

Chemical Kinetics & Reaction Equations

The equations that govern the full equilibrium chemical reactions are presented now

$$\sum_{k=1}^{N_s} \nu_{kj}^{'} X_k \rightleftharpoons \sum_{k=1}^{N_s} \nu_{kj}^{''} X_k \quad , \quad j = 1, \ldots, N_r \tag{C.1}$$

$$\dot{\omega}_k = \sum_{j=1}^{N_r} \dot{\omega}_{kj} = M_k \sum_{j=1}^{N_r} \nu_{kj} q_j \quad ; \quad \nu_{kj} = \nu_{kj}^{''} - \nu_{kj}^{'} \tag{C.2}$$

$$q_j = K_{fj} \prod_{k=1}^{N_s} [X_k]^{\nu_{kj}^{'}} - K_{rj} \prod_{k=1}^{N_s} [X_k]^{\nu_{kj}^{''}} \tag{C.3}$$

$$K_{fj} = A_{fj} T^{\beta} \exp\left(\frac{-E_j}{RT}\right) \tag{C.4}$$

$$K_{rj} = K_{fj} / K_{cj} \tag{C.5}$$

$$K_{cj} = \left(\frac{p_{atm}}{RT}\right)^{\sum_{k=1}^{N_s} \nu_{kj}} \exp\left(\frac{\Delta S_j^0}{R} - \frac{\Delta H_j^0}{RT}\right) \tag{C.6}$$

$$\Delta S_j^0 = \sum_{k=1}^{N_s} \nu_{kj} S_k^{st} \tag{C.7}$$

$$\Delta H_j^0 = \sum_{k=1}^{N_s} \nu_{kj} H_k^{st} \tag{C.8}$$

Global One-Step Reaction

The complete specification of equilibrium reaction rates scales exponentially with the number of carbon atoms in the hydrocarbon fuel. For this reason, global one-step reactions are often

used as approximations to govern the net consumption and generation of products and reactants. The following equations describe this simplified approach:

$$\dot{\omega}_k = M_k \nu_k q_{rxt} \tag{C.9}$$

$$q_{rxt} = A_0 \exp(-T_a/T)[C_{fuel}]^a [C_{o_2}]^b \tag{C.10}$$

$$q_{rxt} = A_0 \exp(-T_a/T) \frac{\rho^{a+b}}{(M_{fuel})^a (M_{o_2})^b} [Y_{fuel}]^a [Y_{o_2}]^b \tag{C.11}$$

To illustrate this formulation clearly, let us take a look at a typical stoichiometric balance of combustion between a hydrocarbon fuel and air:

$$C_\alpha H_\beta O_\gamma + (\alpha + \frac{\beta}{4} - \frac{\gamma}{2}) [O_2 + 3.76 N_2] \rightarrow (\alpha) CO_2 + (\frac{\beta}{2}) H_2O + 3.76(\alpha + \frac{\beta}{4} - \frac{\gamma}{2}) N_2. \tag{C.12}$$

In this case, the net reaction coefficients ν_k for each reactant are the balance coefficients in the equation. For the products, the net reaction coefficients are the negated balance coefficients in the equation,

$$\nu_{fuel} = -1 \tag{C.13}$$

$$\nu_{O_2} = -(\alpha + \frac{\beta}{4} - \frac{\gamma}{2}) \tag{C.14}$$

$$\nu_{CO_2} = \alpha \tag{C.15}$$

$$\nu_{H_2O} = \frac{\beta}{2}. \tag{C.16}$$

All that is left is to specify the molecular weights of each species M_k and the values of $A_0, T_a, a,$ and b. These values have been measured experimentally and are available in tables.

Appendix D

Fluid Properties

Fluid properties

There are five fluid properties that need to be computed: shear viscosity (η), volume viscosity (κ), thermal conductivity (λ), specific heat (c_p), and diffusivity ($\mathcal{D}$) . These properties do not have evolution equations, rather they are evaluated as functions of the evolved variables.

Single Component System

For single component systems there are simple relationships between the temperature and most of the fluid properties. Standard air is the test fluid for all single component systems [120].

Shear Viscosity

$$\eta(T) = 1.716e^{-5} * \left(\frac{T}{273}\right)^{0.66} \left[\frac{Ns}{m^2}\right] \tag{D.1}$$

Volume Viscosity

$$\kappa(T) = 0.778e^{-5} * \left(\frac{T}{200}\right)^{1.38} \left[\frac{Ns}{m^2}\right] \tag{D.2}$$

Thermal Conductivity

$$\lambda(T) = 0.0241 * \left(\frac{T}{273}\right)^{0.81} \left[\frac{W}{mK}\right] \tag{D.3}$$

Specific Heat

$$c_p(T) = 1.704e^{-4} * T + 0.96 \tag{D.4}$$

The specific heat for air in many instances may be set to "1". This linear equation is fit to the measured specific heats of air at $250K$ and $1500K$. Diffusivity is not a parameter that exists for a single component system.

Multicomponent Systems

In general, the fluid properties in multicomponent systems are dependent on (T, p, y_k).

Specific Heat

NASA thermodynamic data specifies the specific heat of each individual gaseous component as a polynomial function of the temperature,

$$(c_p)_k = R\left[a_{1k} + a_{2k}T + a_{3k}T^2 + a_{3k}T^3 + a_{5k}T^4\right]. \tag{D.5}$$

There are typically two such polynomials specified over different temperature ranges. Unfortunately, the composite polynomial is only C^0 continuous. Any derivatives involving $(c_p)_k$ must be computed carefully. The approach taken herein is to extend one of the polynomials to cover the range of simulated temperatures.

For computing the mixture properties, the formulation presented in [30] is employed. The mixture coefficients in multicomponent systems have the following form:

$$G\mu = \beta^\mu \tag{D.6}$$

$$\langle \mathcal{G}, \mu \rangle = 0, \tag{D.7}$$

where μ represents a generic mixture property, G is the system matrix, $\mathcal{G}$ is the constraint vector, $\langle , \rangle$ represents the scalar product, and β^μ is the the associated right hand side for a given μ. This system of equations is linear and is solved iteratively [30].