OPEN-SOURCE COMBUSTING FLOW SIMULATION

An Undergraduate Thesis

By

Henry A. Le Berre

hberre3@gatech.edu

School of Computer Science

Georgia Institute of Technology

December 2024

Approvals:

Prof. Spencer H. Bryngelson (shb@gatech.edu)
School of Computational Science and Engineering
Georgia Institute of Technology

Signature: _____

Date: _____

Prof. Florian T. Schaefer (fschaefer7@gatech.edu)
School of Computational Science and Engineering
Georgia Institute of Technology

Signature: _____

Date: _____

Dedicated to **Dr. Marion Edward Wilson** for giving me sight.

This thesis summarizes the final year of my three years of undergraduate research. As my journey comes to an end, the heart full of gratitude, I remember the people who made it all possible.

My friends never left my side, no matter the distance. Antoine, while I was blind and locked down at dusk, researched fish excavation, as kernel-level move semantics ushered to us cryptic, yet ominous, blue-tinted words. Raphaël grounded me in the present moment and never ceased to cultivate me. McKay and Lily, serving courtside, gave me force over time, finding our lives forever changed after dispassionately standing in line, stumbling upon an empty parking lot. Emiliano, an acquaintance of mine, ought to be thanked for his music, and for volunteering to drive, only that one time, when it was only somewhat late, dark, foggy, and raining outside. Though there may be competition from other places, there is nobody I would rather be lost in space with, near a national park, rocket launch, or SR-71.

I was fortunate to be surrounded by sharp, jovial, and supportive colleagues: Ben had unprecedented simulation ideas; Anand esoteric book recommendations; Max cookies; Dimitri sensational wit and a powerful `c_avg` formulation; Spencer cubes, a ray-tracing showcase, a shearing claim to rule, and `.md` commits; and Esteban *complexified* papers he beheld.

Lastly, I must thank my family: my father, Daniel, for his incredible selflessness; my brother, Romain, for our deep discussions; my mother Maria, and brother Vincent, for their unwavering belief in my academic persuits; and my grandparents Virginia, Warren, Paulette, and Daniel for their love beyond borders and time.

"And after all, we're only ordinary men."
— Pink Floyd, Us and Them (1973)

CHAPTER 1
INTRODUCTION

Combustion processes power the modern world, from cooking appliances, aircraft, and cars, to industrial manufacturing equipment, rockets, and missiles. About 35% of the world's energy is still produced by the exothermic reaction of coal-burning.

Scientists have developed increasingly sophisticated models to describe the mechanics of combustion in the presence of fluid flow, where many of its applications lie. However, these models most often only correctly describe local behavior, and cannot easily be extended to predict how systems of non-trivial size and complexity will evolve over time.

In some cases, simplifying assumptions can be made and used alongside empirical results to reason about plausible outcomes, though the confidence with which these predictions can be made is variable. In other cases, when our predictive power is insufficient and the consequences of failure are severe – as may be the case when designing critical infrastructure – we must often rely on computer simulations and experiments, to ground our designs and decisions thereabout in objectivity.

To illustrate, engineers developing the Apollo program's F-1 rocket engine famously struggled with combustion instability, a phenomenon that remains a challenge today and a focus of study. Small local disturbances in the combustion chamber would lead to catastrophic failures. Fortunately, this problem manifested itself during testing and mitigation strategies were developed. Undiagnosed, however, difficult to predict issues like these could lead to the loss of human life. We are therefore motivated to deepen our understanding of combustion processes.

Our goal in this work is to begin the development of a high-performance, open-source, and permissibly licensed (MIT) solver capable of simulating reacting flows. We hope to help engineers and researchers build novel, efficient, reliable, and safe combustion systems without proprietary software they cannot improve or share with others. To this end, we are extending the Multi-Component Flow Code (MFC), a GPU-accelerated multi-physics compressible fluid flow solver capable of scaling to the largest supercomputers in the world today.

CHAPTER 2
LITERATURE REVIEW

In this work, we will continue to develop the Multi-Component Flow Code (MFC) [1], the flow solver mentioned in the introduction, born at the Computational and Data-Driven Fluid Dynamics research group at the California Institute of Technology. MFC is as a software package for the simulation of multi-component and multi-phase flows, targeted at solving problems in the defense, aerospace, and healthcare sectors. For many problems of interest, such as the modeling of bubble cavitation in the wake of a submarine's propeller, computer simulations help to overcome the burdens of physical experimentation and theoretical analysis, while allowing for more flexibility in the design and validation of new technologies.

Fully resolved simulations require significant computational resources, surpassing the capabilities of a single computer. High-performance computing (HPC) systems provide a network of computers that can be used simultaneously to solve a single problem. To this end, MFC's Fortran codebase leverages the Message Passing Interface (MPI) to distribute work among multiple computer nodes, assigning portions of the computational domain to individual processors in the pre-processing step.

Communication between processors is required to compute fluxes across the faces of cells residing on the boundary of processor domains, as they lack information about their neighbors. MFC makes use of "halo" regions, thin layers of cells surrounding the interior domain of each processor, whose purpose is to store the values of the neighboring processors' boundary cells, for use in the computation of fluxes. These regions are updated every stage of our Runge–Kutta time integration scheme. At the very edge of the computational domain, non-periodic user-specified boundary conditions are enforced by prescribing the values of the halo cells thereon.

In a subsequent research effort, we introduced hardware acceleration through the use of General-Purpose Graphics Processing Units (GPGPUs) [2]. At the time, on the nation's most most powerful supercomputer, Oak Ridge Leadership Computing Facility's (OLCF) Summit, a 40-times node-wise speedup was achieved when comparing GPU-offload to traditional CPU computation, greatly reducing the cost of running large simulations. Of-

floading was done through the OpenACC programming model, using meta-programming and case-specific optimizations to achieve maximum performance, among other techniques.

MFC employs the 4- and 5-equation models from Preston et al. [3] to model the physics of flow. These are systems of partial differential equations of the form:

$$\frac{\partial q}{\partial t} + \nabla \cdot F(q) + h(q)\nabla \cdot u = s(q) \tag{2.1}$$

where q is the state vector of conserved quantities, F the flux tensor, u the velocity field, and $s(q)$ the associated source term [1]. For the purposes of this review, we will not detail the $h(q)\nabla \cdot u$ term. Each component of q represents a conserved quantity at any given grid location, such as ρE (density times energy). Fluxes designated by $F(q)$ represent how each component in the state vector is expected to be transported to and from the current location (to and from other cells). By definition, a flux represents the net transport of a quantity through a face. MFC is a finite-volume code and represents the domain as a collection of cells with non-zero volume [1]. Naturally, adjacent cells share faces. Preston et al. [3] provide expressions for q, $F(q)$,$h(q)$, and $s(q)$, capable of modeling the flows MFC is interested in simulating.

To compute fluxes across cell faces, MFC uses (Monotonicity Preserving) WENO (Weighted Essentially Non-Oscillatory) schemes to reconstruct values at cell faces from their cell-centered counterparts in a way that does not introduce spurious oscillations into the simulation [4, 5]. Indeed, the simulation stores and manipulates state vectors at the cell centers, not their faces, whose values are necessary to calculate derivatives using finite differences and other numerical methods. WENO uses the cell-centered values in the neighborhood of a cell to derive its face-centered values through weighted interpolation, while attempting to minimize numerical artifacts that can result in the formation of spurious oscillations. WENO schemes are dissipative, meaning they can dampen oscillations and smear discontinuities.

From these WENO-reconstructed face-centered values, one could naively compute fluxes across cell faces using $F(q)$. However, MFC is interested in simulating flows that involve shocks, often in the form of propagating waves. They give rise to sharp discontinuities in the flow field, where across two cell faces, the value of a state variable can abruptly jump from one value to another. For example, if two fluids are inserted, each on one side of the domain with their own density, an interface forms between the two components, where density jumps from one value to another without a smooth transition. Differentiating a state variable at the discontinuity can lead to unphysical solutions, as the field can no longer be modeled as continuous. This is due to the discreteness of the grid imposed by computer simulation. Approximate Riemann solvers (solving the Riemann Problem) such as HLLC (Harten-Lax-van Leer-Contact) are used to determine the flux through a discontinuity [6]. They use information about the shock waves moving through the domain to determine fluxes accurately.

The methods presented above let us determine $\frac{\partial q}{\partial t}$ at a particular time-step, the (partial) time derivative of the conservative state vector. These are called "right-hand-side (RHS) evaluations". To advance the simulation through time, we must integrate it – with respect to time. MFC uses a popular class of iterative schemes known as explicit Runge–Kutta (RK) methods [7]. They evaluate and compose $\frac{\partial q}{\partial t}$ multiple times per time-step to provide a guess for the q state vector at the next time-step. Their popularity lies in their great accuracy at a low computational cost (measured in the number of right-hand-side evaluations). There are multiple versions of this method. For example, RK2 can be written as

$$\begin{cases} K_1 & = hf\left(t_n, q_n\right) \\ K_2 & = hf\left(t_n + \frac{h}{2}, q_n + \frac{K_1}{2}\right) \\ q_{n+1} & = q_n + K_2 \\ t_{n+1} & = t_n + h \end{cases}$$

for a given (time) step size $h \ll 1$ if we express our problem as $\frac{\partial q}{\partial t} = f(t, q)$ [7]. MFC simulations often make use of another variation, a third-order total variation diminishing (TVD) Runge–Kutta method from Gottlieb and Shu [8] that provides more guarantees about the stability of the integration scheme.

To simulate chemistry, more equations and state variables are needed. Poinsot and Veynante provide a framework for simulating chemical advection, diffusion, and chemical kinetics, among other phenomena [9]. Most notably, they provide the following

conservative transport equation for species k:

$$\frac{\partial (\rho Y_k)}{\partial t} = \dot{\omega}_k - \nabla \cdot (\rho Y_k \vec{u}) - \nabla \cdot (\rho Y_k \vec{V_k}) \quad (2.2)$$

where Y_k is the mass fraction of the k-th species, ρ is the density, u is the velocity field, $\dot{\omega}_k$ is the reaction source term for species k, and $\vec{V_k}$ is the correction velocity for species k (which accounts for chemical diffusion). As is evident from the above equation, the authors recommend the use $q_k = \rho Y_k$ as the conservative variable associated with species k and, Y_k (or X_k, the molar fraction) as the associated primitive variable.

Martínez-Ferrer et al. [10] present detailed validation procedures for combustion CFD codes, some of which we will follow in chapter 4. These mostly stress advection fluxes, the reaction source term, shock-capturing, and our temperature and pressure calculations. The two one-dimensional shock tube problems they present, *Multicomponent inert shock tube* and *Multi-species reactive shock tube* are already present in MFC's source package but without the chemistry.

To validate the chemical diffusion term, we can either begin with the full formulation shown above, or consider a simpler case such as binary diffusion, where only two species are present and susceptible to diffuse into each other. With these simplifying assumptions, Poinsot and Veynante [9] show that the transport equation can be rewritten as

$$\frac{\partial (\rho Y_k)}{\partial t} = \dot{\omega}_k - \nabla \cdot (\rho Y_k) - \nabla \cdot (\rho D \nabla Y_k)$$

where D is a constant specific to the two species mixing, dropping the need to solve a system for diffusion velocities.

This can be validated using results from Antonio L. Sánchez et al., who study one-dimensional binary mixing problems [11]. Their work lets us validate our implementation of the above differential equation with the following problem definition and time-dependent exact analytical solution thereof

$$\begin{cases} \rho(r,t) & = \rho_b + \frac{\rho_a - \rho_b}{2}\left[1 - \operatorname{erf}\left(\frac{r}{2\sqrt{Dt}}\right)\right] \\ Y_1(r,t) & = \frac{\rho_a[\rho_b - \rho(r,t)]}{\rho(r,t)[\rho_b - \rho_a]} \\ Y_2(r,t) & = 1 - Y_1(r,t) \end{cases}$$

where t is the time, ρ_a and ρ_b are, respectively, the densities of the two mixing gases, $\rho(r,t)$ and $Y_k(r,t)$ are, respectively, the density and mass fraction of the k-th species at time t and position r along the 1D domain. The mixing layer has a thickness of $2\sqrt{Dt}$ (in domain space).

Similarly, multi-species diffusion can first be validated using the above binary diffusion case, before moving to the more complex case described by Martínez-Ferrer et al. [10].

Multiple software packages exist to assist in computing correction velocities and the reaction source term such as Cantera [12], Pyrometheus [13], and KinetiX [14]. Pyrometheus and KinetiX are both designed to generate optimized thermochemistry routines from Cantera mechanisms, aiming to maximize performance by unrolling loops, optimizing the computation graph, and hard-coding mechanism constants like NASA polynomial coefficients, so as to reduce memory accesses and eliminate redundant computations. Neither provide Fortran bindings.

One application of simulating reacting flows is the design and validation of scramjets. They are engines whose combustion happens at supersonic speeds (Mach > 1) [15]. F. Ladeinde brings attention to spray modeling, encouraging more research to be done in the area, and highlights the prevalence of flamelet methods as a tool to simplify combustion simulation [15]. This method is also presented in Poinsot and Veynante's book [9] as a simplification that models each element of the frame-front (the part of the flame that is reacting) as a laminar flame. These elements are called flamelets and can provide significant performance improvements [9, 15].

CHAPTER 3
METHODS

MFC with chemistry (herein CheMFC), is based on a 1-fluid formulation of the 5-equation model from Allaire et al. [3], augmented with N_{sp} species transport equations from Poinsot and Veynante [9].

$$\frac{\partial (\rho Y_k)}{\partial t} = \dot{\omega}_k - \nabla \cdot (\rho Y_k \vec{u}) - \nabla \cdot (\rho Y_k \vec{V_k})$$

We naturally take Y_k as our primitive variable and ρY_k as our conservative variable for species $k \in \{1, \ldots, N_{sp}\}$. Each Y_k is WENO-reconstructed and the associated advection flux computed using either the HLL or HLLC Riemann solver. Therein, γ, the ratio of specific heat capacities, is computed in

one of two ways:

$$\gamma = \frac{c_p}{c_v} = \frac{\sum_{k=1}^{N_{sp}} Y_k c_{p,k}}{\sum_{k=1}^{N_{sp}} Y_k c_{v,k}} \quad \text{or} \quad \gamma = 1 + \left(\sum_{k=1}^{N_{sp}} \frac{X_k}{\gamma_k - 1} \right)^{-1}$$

the latter formulation being taken from Deiterding [16].

To compute temperature (T) from the mixture density (ρ) and our conservative energy variable (ρE), we must perform an iterative solve using the Newton-Raphson method, as to satisfy

$$e - \sum_{k=1}^{N_{sp}} e_k(T) Y_k = 0 \tag{3.1}$$

where, e, the internal mixture energy, can be defined as $e = E - \frac{1}{2}\|u\|^2$. Pressure is simplistically defined using the ideal gas law, $p = \rho R T$ where R is the ideal gas constant.

We extended Pyrometheus [13] to provide a Fortran interface for computing the reaction source term ($\dot{\omega}_k$) and the specific heat capacities at constant volume and pressure, as well as performing the Newton-Raphson solve for temperature described in Equation 3.1, while supporting OpenACC offload. MFC's build system was amended to allow for mechanism-specific code-generation and compilation to occur transparently to the user.

To conclude, our conservative and primitive variable vectors are defined as:

$$\vec{Q}_{\text{cons}} = \begin{bmatrix} \alpha\rho \\ \rho u_i \\ \vdots \\ \rho u_{N_{dim}} \\ \rho E \\ \alpha \\ \rho Y_1 \\ \vdots \\ \rho Y_{N_{sp}} \\ T \end{bmatrix} \qquad \vec{Q}_{\text{prim}} = \begin{bmatrix} \alpha\rho \\ u_i \\ \vdots \\ u_{N_{dim}} \\ p \\ \alpha \\ Y_1 \\ \vdots \\ Y_{N_{sp}} \\ T \end{bmatrix}$$

where the fluid void fraction (α) and temperature (T) variables, and their respective equations, could be done without, but kept for simplicity, given α is a remnant from the 5-equation model (given our 1-fluid formulation) and temperature, derived from energy, is cached (and not recomputed) every time it is used. The memory and computational overhead is considered negligible since the species transport equations are expected to dominate the number of equations in the system.

CHAPTER 4
VALIDATION

4.1 Perfectly Stirred Reactor

This non-dimensional problem aims to stress the species rate of formation source term, our equations of state, and computation of energy. The initial condition consists of a perfectly stirred mixture of $H_2/O_2/AR$ with molar ratios 4/2/94 at $T = 1600$ K and $P = 1$ atm. For this configuration, Skinner et al. [17] expect an ignition delay of 0.052 ms, characterized by [OH] reaching 10^{-6} $\frac{\text{mol}}{\text{L}}$. As seen in Figure 4.1, our results agree with theirs within 1.3% at 0.0513 ms. With the Cantera [12] software package, our relative error drops below 10^{-12} %.

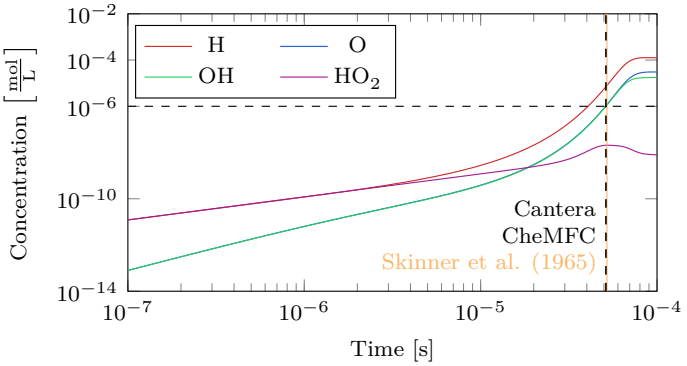


Figure 4.1: Perfectly Stirred Reactor Ignition Delay

4.2 Multi-species Inert Shock Tube

Here, we aim to validate our shock-capturing capabilities and our implementation of multi-species advection and convection, using a problem first described by Fedkiw [18], mentioned by Martínez-Ferrer et al. [10].

The initial condition consists of $L = 10$ cm-long tube discretized with 400 grid cells, containing a 2/1/7 molar ratio of $H_2/O_2/AR$. The temperature and pressure are subject to a Riemann problem at $x = \frac{L}{2}$ whose left and right states are given, respectively, by $(T_L, P_L) = (400\text{ K}, 8\,000\text{ Pa})$ and $(T_R, P_R) = (1\,200\text{ K}, 80\,000\text{ Pa})$. The left boundary

is reflective while the right boundary is extrapolative.

We run the simulation for $t = 40$ µs using a constant step size of 0.2 µs. Results are shown in Figure 4.2 against those of Fedkiw [18]. We also plot Martínez-Ferrer's results for γ, the ratio of specific heat capacities, given the low resolution of Fedkiw's data.

4.3 Multi-Species Reactive Shock Tube

We now consider a reactive shock tube problem adapted by Martínez-Ferrer et al. [10] to validate the interplay of advection, convection, and chemical reactions while in the presence of discontinuities.

Initially, a 2/1/7 molar ratio of $H_2/O_2/AR$ is placed in a $L = 12$ cm-long tube discretized with 400 grid cells. A Riemann problem is set at $x = \frac{L}{2}$ whose left and right states are given by, respectively, $(\rho_L, u_L, P_L) = \left(0.072\ \frac{\text{kg}}{\text{m}^3}, 0\ \frac{\text{m}}{\text{s}}, 7\ 173\ \text{Pa}\right)$ and $(\rho_R, u_R, P_R) = \left(0.18075\ \frac{\text{kg}}{\text{m}^3}, 0\ \frac{\text{m}}{\text{s}}, 35\ 594\ \text{Pa}\right)$. The left boundary is reflective while the right boundary is extrapolative.

When the leftward-moving wave reflects off the left boundary, it initiates a rightward-moving reaction wave, pictured in Figure 4.3. We compare our results to those of Martínez-Ferrer et al. [10] in Figure 4.4.

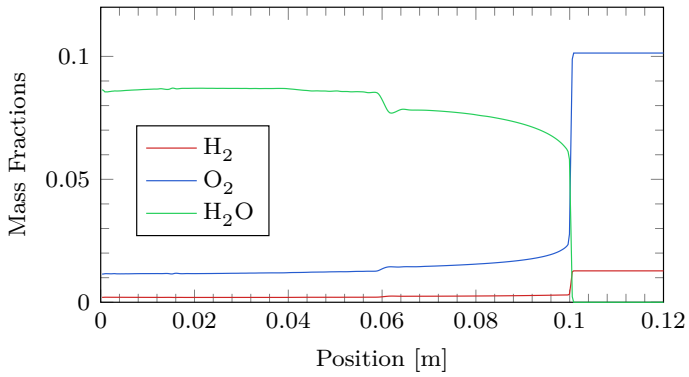


Figure 4.3: Multi-Species Reactive Shock Tube. Mass Fraction profiles of H_2, O_2, and H_2O at $t = 230$ µs.

CHAPTER 5
EXAMPLE SHOWCASE

In this chapter, we seek to demonstrate CheMFC's capabilities beyond the validation test cases presented in chapter 4 by studying the structure of two-dimensional detonations. As previously described by Kailasanath et al. [19] and Lefebvre et al. [20], this can be done by simulating a reacting shock tube whose flow upstream of the reaction front is perturbed to allow for the formation of transverse waves.

To this end, we use results from our one-dimensional reactive shock tube simulation (section 4.3) to initialize a two-dimensional simulation. More specifically, we extract the one-dimensional state of the reactive shock tube once a unique detonation front has formed and reached the Chapman-Jouguet (CJ) velocity. The two-dimensional initial condition is then constructed by replicating this state in the y-direction, and perturbing the flow upstream of the detonation front with three rectangular regions of unburned fuel at pressures 1.6 times greater than that of the background. In this problem, the left boundary is reflective, the right boundary is extrapolative, and the top and bottom boundaries are periodic.

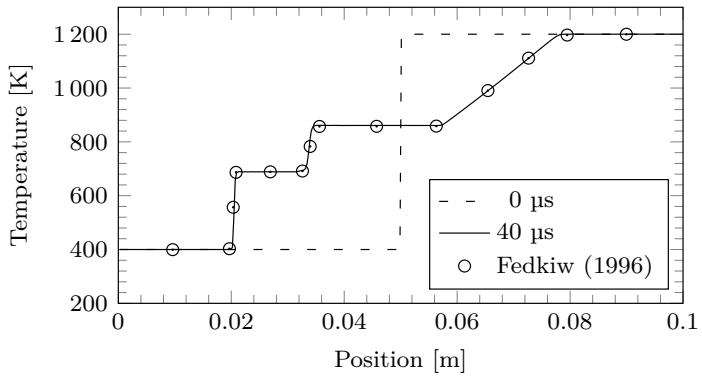
The 12 cm × 6 cm domain is discretized with 800 × 200 grid cells and simulated using the HLLC Riemann solver on 4 NVIDIA A100 GPUs and Cantera's `h2o2.yaml` mechanism, a (modified) subset of GRI-Mech 3.0 [21]. A plot of the numerical schlieren of the results is shown in Figure 5.1, revealing detonation cells similar to those of Kailasanath et al. [19].

CHAPTER 6
DISCUSSION AND FUTURE WORK

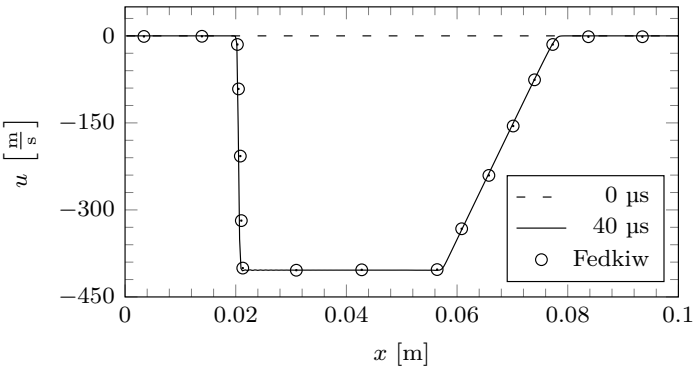
We set out to extend the Multi-component Flow Code (MFC) with combustion modeling capabilities, in an effort to provide a performant and permissively licensed solution for combustion simulation in the open-source space. We chose MFC for its performance, scalability, cross-compatibility across GPU vendors and HPC systems, interface-capturing schemes, and extensibility.

The goal was met with MFC's v4.9.7 release, the first to contain validated 0-1D combustion modeling capabilities, as seen in chapter 4. Some fixes required for the 2D simulation shown in chapter 5 are awaiting to be merged.

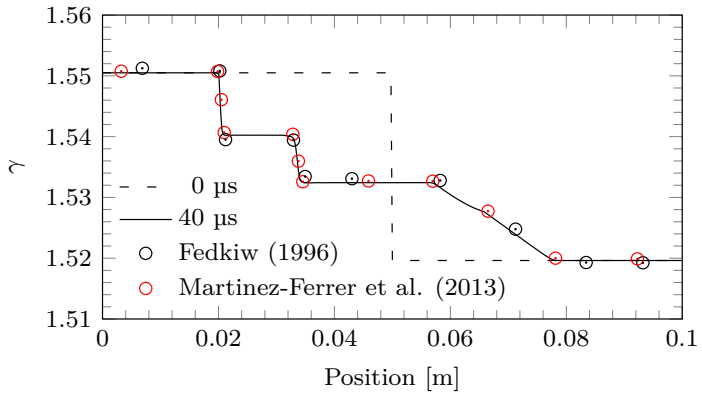
However, several limitations remain, preventing us from simulating some more complex scenarios. Three main areas of improvement are identified: the lack of chemical diffusion, the need for non-ideal equations of state, and the need for varying bound-



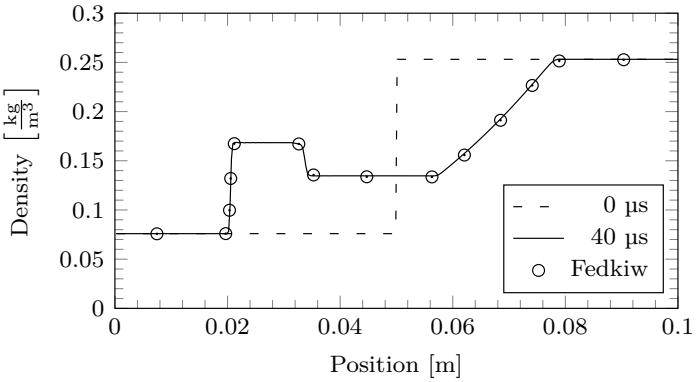
(a) Temperature



(b) Velocity

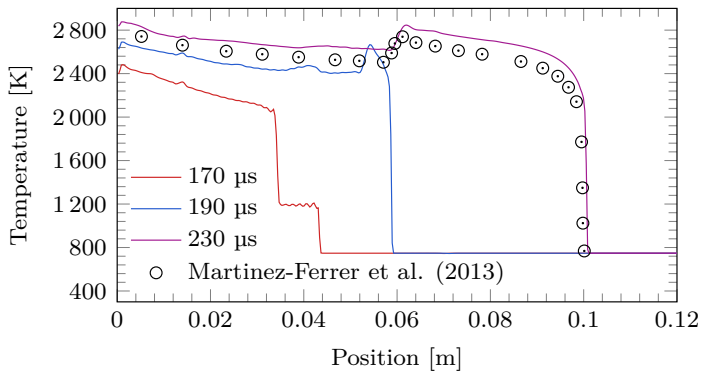


(c) Ratio of Specific Heat Capacities

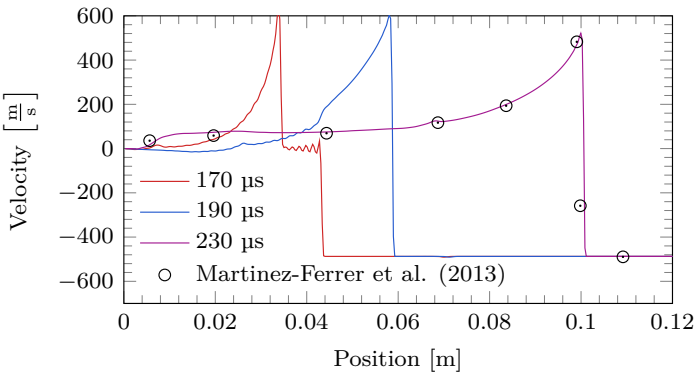


(d) Density

Figure 4.2: Multi-Species Inert Shock Tube. Temperature, Velocity, Ratio of Specific Heat Capacities, and Density profiles at $t = 40$ µs.



(a) Temperature



(b) Velocity

Figure 4.4: Multi-Species Reactive Shock Tube. Temperature and Velocity profiles at $t = 170, 190, 230$ µs.

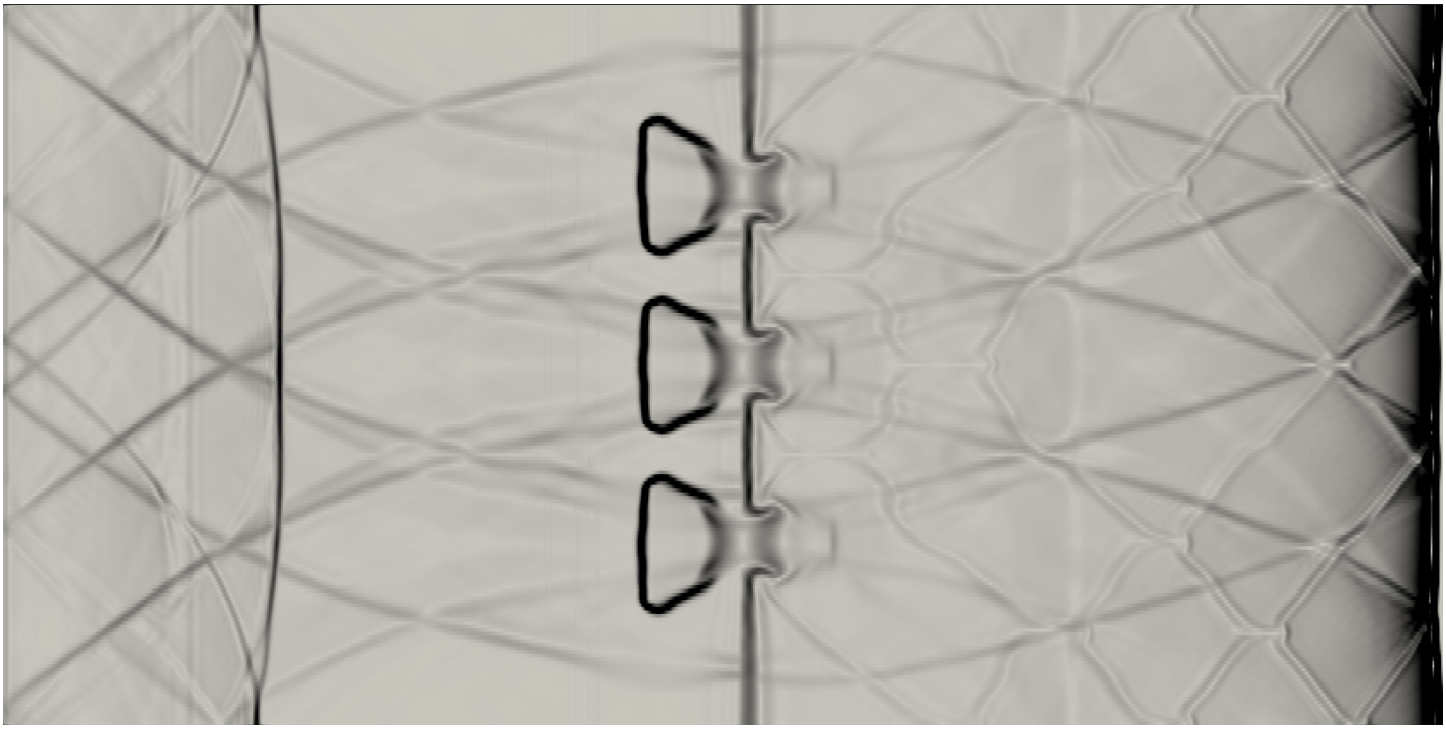


Figure 5.1: Example Showcase: Numerical schlieren of a reacting shock tube whose flow upstream of the reaction front is perturbed to allow for the formation of transverse waves. Detonation cells are visible.

ary conditions along the edges of the domain. The first two are critical for the fidelity of the simulations, while the third is necessary to properly define the associated problem setups, such as having an inflow boundary condition between two no-slip walls.

Performance and high memory footprint will remain a concern, even when offloading to GPUs, given the number of equations to be solved is an order of magnitude higher than in regular MFC simulations. It impacts the size of the simulations we can reasonably expect to run. This problem, however, is to be expected for this class of solver, and one is we can attempt to further mitigate by optimizing the Pyrometheus [13] thermochemistry library.

ACKNOWLEDGEMENTS

This work used resources from the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725 (PI Bryngelson; allocation CFD154).

I acknowledge and thank Dimitrios Adam, Dr. Esteban Cisneros-Garibay, and Prof. Spencer H. Bryngelson for their respective contributions to this work.

REFERENCES

[1] S. H. Bryngelson, K. Schmidmayer, V. Coralic, J. C. Meng, K. Maeda, and T. Colonius, "MFC: An open-source high-order multi-component, multi-phase, and multi-scale compressible flow solver," *Computer Physics Communications*, p. 107 396, 2021. DOI: 10.1016/j.cpc.2020.107396.

[2] A. Radhakrishnan *et al.*, *Method for portable, scalable, and performant gpu-accelerated simulation of multiphase compressible flow*, 2024.

[3] G. Allaire, S. Clerc, and S. Kokh, "A five-equation model for the simulation of interfaces between compressible fluids," *Journal of Computational Physics*, vol. 181, no. 2, pp. 577–616, 2002. DOI: https://doi.org/10.1006/jcph.2002.7143.

[4] V. Coralic and T. Colonius, "Finite-volume weno scheme for viscous compressible multi-component flows," *Journal of Computational*

Physics, vol. 274, pp. 95–121, 2014. DOI: https://doi.org/10.1016/j.jcp.2014.06.003.

[5] E. Johnsen and T. Colonius, "Implementation of weno schemes in compressible multicomponent flow problems," *Journal of Computational Physics*, vol. 219, no. 2, pp. 715–732, 2006. DOI: https://doi.org/10.1016/j.jcp.2006.04.018.

[6] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 2009, ISBN: 978-3-540-25202-3.

[7] J. Butcher, "A history of runge-kutta methods," *Applied Numerical Mathematics*, vol. 20, no. 3, pp. 247–260, 1996. DOI: https://doi.org/10.1016/0168-9274(95)00108-5.

[8] S. Gottlieb and C.-W. Shu, "Total variation diminishing runge-kutta schemes," *Mathematics of Computation*, vol. 67, no. 221, pp. 73–85, 1998.

[9] T. Poinsot and D. Veynante, *Theoretical and Numerical Combustion*. R.T. Edwards Inc., 2005, p. 522.

[10] P. J. Martínez-Ferrer, R. Buttay, G. Lehnasch, and A. Mura, "A detailed verification procedure for compressible reactive multicomponent navier–stokes solvers," *Computers & Fluids*, vol. 89, pp. 88–110, 2014. DOI: https://doi.org/10.1016/j.compfluid.2013.10.014.

[11] A. L. Sánchez, M. Vera, and A. Liñán, "Exact solutions for transient mixing of two gases of different densities," *Physics of Fluids*, vol. 18, no. 7, p. 078 102, Jul. 2006. DOI: 10.1063/1.2221349.

[12] D. G. Goodwin, H. K. Moffat, I. Schoegl, R. L. Speth, and B. W. Weber, *Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes*, https://www.cantera.org, Version 3.0.0, 2023. DOI: 10.5281/zenodo.8137090.

[13] E. Cisneros, A. Kloeckner, and H. Le Berre, *Pyrometheus: Code generation for combustion mechanisms*, https://github.com/pyrometheus/pyrometheus, 2024.

[14] B. A. Danciu and C. E. Frouzakis, *Kinetix: A performance portable code generator for chemical kinetics and transport properties*, 2024.

[15] F. Ladeinde, "A critical review of scramjet combustion simulation (invited)," in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. DOI: 10.2514/6.2009-127.

[16] R. Deiterding, "Parallel adaptive simulation of multi-dimensional detonation structures," Ph.D. dissertation, Brandenburgische Technische Universitat Cottbus, Sep. 2003.

[17] G. B. Skinner and G. H. Ringrose, "Ignition delays of a hydrogen—oxygen—argon mixture at relatively low temperatures," *The Journal of Chemical Physics*, vol. 42, no. 6, pp. 2190–2192, Mar. 1965. DOI: 10.1063/1.1696266.

[18] R. P. Fedkiw, "A survey of chemically reacting, compressible flows," Ph.D. dissertation, University of California, Los Angeles, 1996.

[19] K. Kailasanath, E. Oran, J. Boris, and T. Young, "Determination of detonation cell size and the role of transverse waves in two-dimensional detonations," *Combustion and Flame*, vol. 61, no. 3, pp. 199–209, 1985. DOI: https://doi.org/10.1016/0010-2180(85)90101-4.

[20] M. Lefebvre, E. Oran, K. Kailasanath, and P. Van Tiggelen, "The influence of the heat capacity and diluent on detonation structure," *Combustion and Flame*, vol. 95, no. 1, pp. 206–218, 1993. DOI: https://doi.org/10.1016/0010-2180(93)90062-8.

[21] G. P. Smith *et al.*, *Gri-mech 3.0*, http://www.me.berkeley.edu/gri_mech/, Accessed: 2024-12-03.