A COMPUTATIONAL APPROACH FOR PRELIMINARY COMBUSTOR DESIGN AND GASEOUS EMISSIONS EVALUATIONS USING A METHOD FOR SPARSE KINETICS

A Dissertation
Presented to
The Academic Faculty

by

Adam D. Siegel

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December, 2016

A COMPUTATIONAL APPROACH FOR PRELIMINARY COMBUSTOR DESIGN AND GASEOUS EMISSIONS EVALUATIONS USING A METHOD FOR SPARSE KINETICS

Date Approved: 4 November 2016

To my parents,

the ones who have helped me the most along the way.

ACKNOWLEDGEMENTS

It is an incredible opportunity to critically think about, analyze, and attack a problem that is truly one's "own". This document is the culmination of ample work including long hours coding, debugging, writing, and documenting, with countless conversations involving technical explanations, deliberation, and testing. Most importantly, this document marks the end of the first problem I have sincerely tackled on my own, and the beginning of a long and fruitful career with many more hard problems to ahead.

I would like to thank my advisor, Professor Dimitri "Doc" Mavris, who gave me the incredible opportunity to study and earn an advanced degree in the field of Aerospace Engineering. It was an honor to attend Georgia Tech, as I have thoroughly enjoyed my time in Atlanta. I truly can not thank you enough for everything you have done for me, and am honored to have worked under you. I would also like to thank Russ Denney, who worked with me closely, listening and providing insight on a regular basis. You provided invaluable feedback, and for that I am appreciative. My other committee member from Georgia Tech, Professor Jeff Jagoda, always made time to meet with me and provided me with feedback and encouragement as I worked very hard to attain my goals.

I also have two external committee members, Dr. Cristian Nastase and Bill Cummings of Rolls-Royce. Cris, I can not thank you enough for your technical input on CFD, and Bill, for your technical input on combustor design. Neither of you were obligated to sit on my committee, but both accepted with immeasurable willingness. I can not extend enough thanks to you both.

I would also like to give a special thanks to my undergraduate advisors, Professors Alan Calder, Mike Zingale, and Doug Swesty, in the department of Physics and

Astronomy at Stony Brook University. You are the ones who started me on this long and treacherous road they call ”computational physics”. You all equipped me with the skills I needed, and for that I am very thankful. My only hope is that you continue your tenures as professors and influence others the way you influenced me.

Most importantly, it is my honor and pleasure to thank all of my family members and friends, without whom I would not be where I am today. All of my friends and colleagues from Georgia Tech have helped me in one way or another. Additionally, I would like to thank Anthony Camino, a friend and fellow Ph.D. student in computer science for spending time helping me debug my code. I would like to thank my girlfriend, Alexis, who has shown so much patience, love, and support for me while working on my dissertation. Alexis, you had to deal with my complaining regularly, and your response was nothing short of happiness and encouragement. Most importantly, I would like to thank my parents, without whom I would not be here (literally). You have always been my biggest supporters, and words can not describe my level of gratitude to the both of you. I owe everything to you both, and it would take more than a lifetime to give you the thanks that you deserve!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Despite multifarious advances in the fields of numerical computing and experimentation, the development of new gas turbine combustors continues to be an extremely costly endeavor. Contemporary aircraft engine component design involves utilizing knowledge from previous architectures (in the form of trends from data, etc.), which the designer uses to develop an improved product. Burners are no different, as gas turbine combustors are conventionally designed using correlations for initial sizing and precursory emissions evaluation. With pollutants becoming an ever increasing concern in air-breathing propulsion systems, the International Civil Aviation Organization strictly regulates the amount of NOx which can be emitted from a given engine. This makes oxides of nitrogen one of the central drivers of modern combustor design, since NOx production considerations cause major restrictions on the properties of the combustor and the conditions under which it can operate.

Future engine cycles are constantly pushed to operate at increasingly higher overall pressure ratios in order to achieve new SFC goals. Consequentially, these higher pressure ratios also tend to increase NOx emissions, and so new combustor concepts and technology must therefore be employed to achieve the lower NOx emissions called for by regulations. Utilizing correlations for these future designs however is not possible (since these combustors do not yet exist), and creating these correlations involves very expensive experimentation via computational fluid dynamics (CFD) and rig tests. To counter this cost ramification, many techniques oriented at combustor sizing and NOx prediction have been developed in the past such as flow network modeling, combustor reactor network modeling, and hybrid CFD methods. These approaches still fall short of what is needed since most are either correlation based (or necessarily involve a calibration step), and all are unable to predict NOx emissions quickly enough and accurately enough for a concept which does not yet exist.

This work aims to establish a new sizing technique which will give the combustor designer a starting point in the development of an advanced concept. This approach should not rely so heavily on previous data and testing and should be on the same order of accuracy as current preliminary design techniques (if the data for the new combustor were available). Additionally, the method should be as speedy as possible in order to run fast enough to perform rapid trade studies and assessments of new concepts via known design approaches. This proposed approach will bring accurate information about flow behavior and kinetics earlier in the preliminary design process by combining geometry and combustor characteristics into the combustor sizing and emissions predictions processes. Ultimately, this methodology will allow the designer to understand more accurately and less expensively if an advanced concept is viable, while reducing the amount of expensive testing. Although this approach makes necessary sacrifices in certain areas of accuracy (as compared to full CFD solutions), it aims to lower the computational cost of current methods, which will in turn contribute to lowering the overall cost of developing new combustor concepts for future applications. By introducing a novel approach to indirectly couple source terms with the conservation equations, the overall work attempts to determine the viability of (and develop a methodology to) bring geometry and accurate emissions predictions to an earlier stage in the design process, all the way to the initial sizing phase. The goal is to accurately and more quickly predict emissions (including the effects from geometry) in the preliminary design phase without relying on expensive testing and CFD models to predict emissions. Ultimately, this overall approach aims to reduce the amount of iterations during the design process.

The MoST (Method of Sparse Thermochemistry) algorithm offers a solution to this problem by overlaying a sparse kinetics grid on top of a converged non-reacting CFD solution. The two codes are tied together and controlled by an overarching algorithm that indirectly couples the flow field and kinetics in such a way that the flow field is still influenced by the chemical process happening within it. This eliminates the need to solve coupled nonlinear PDEs, and instead solves them as a much smaller ODE (or algebraic) systems. Research is performed to understand the elements which need to be captured to turn the flow field into a chemical kinetics network (CKN), exactly how to orient the reactors to best mock the physical processes within the flow field, and how to converge on a flow field solution while taking into account its response to chemical reactions. Grids with varying cell densities and chemical networks with varying numbers of reactors are run using this method. The results obtained show good agreement with emissions data from reacting flow CFD runs with moderate fidelity.

CHAPTER I

BACKGROUND AND MOTIVATION

Air breathing propulsion systems are among the most expensive pieces of integrated machinery known. Designing an aircraft engine continues to be an extremely costly endeavor, even when utilizing knowledge of previous designs. Conventionally, the design of said engines is split among component groups, e.g., compressors and combustors (and even further specialized to sub-component groups, e.g., structures, aerodynamics, and materials). Specifically, requirements derived from the engine cycle and goals (e.g., higher efficiency, lower emissions, etc.) are given to each component group where sub-systems are developed to meet these specific targets before they are further integrated into a complete engine design. Contemporary component design involves using knowledge of previous architectures (in the form of trends from data, etc.), which allows the designer to develop an improved product over it's predecessor. The development of new technologies and designs which enable better performance is always under way, bringing forth more advanced components and therefore allowing the engine to achieve new goals.

Pollutant emissions from aircraft engines are becoming an ever increasing concern. The topic of aircraft engine emissions is quite broad, including gaseous and solid particulates. One well known class of gaseous pollutant emissions are the oxides of nitrogen (NOx), which are of large concern in air-breathing propulsion systems, and consist of molecules such as nitric oxide (NO), and nitrogen dioxide (NO_2). The International Civil Aviation Organization (ICAO) strictly regulates the amount of NOx which can be emitted from a given engine (Figure 1).

NOx emissions are measured using an overall evaluation criteria (OEC), which is

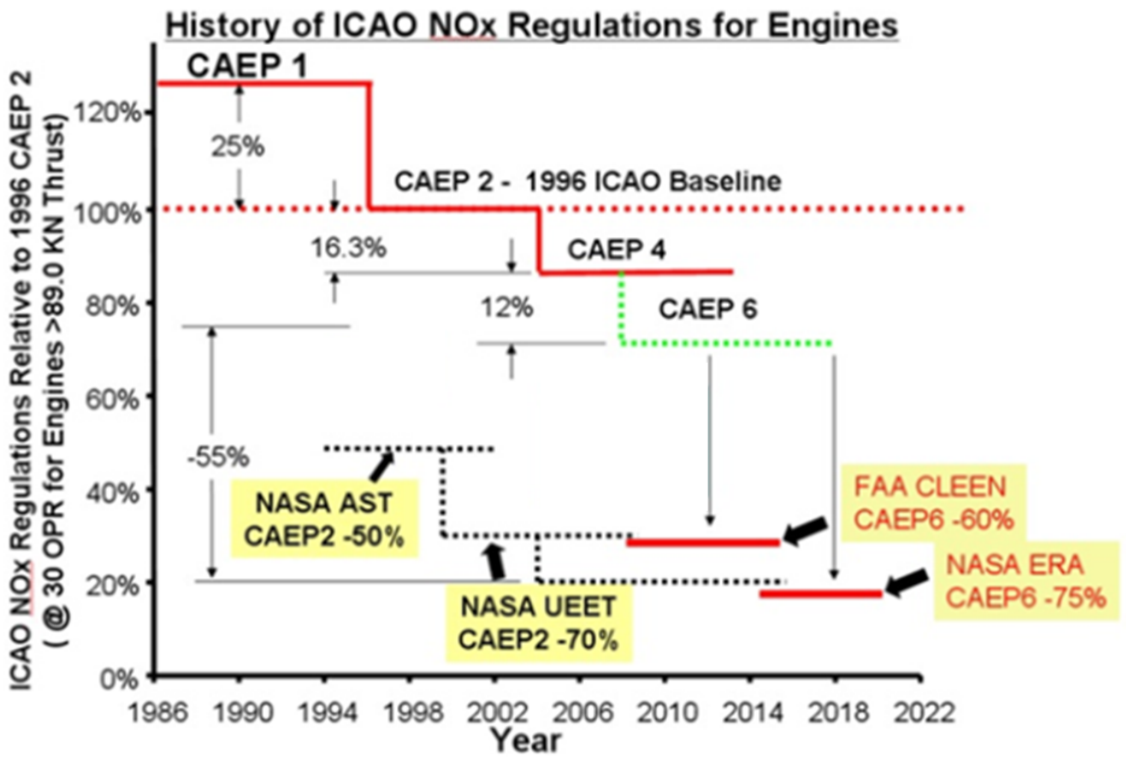


Figure 1: History of ICAO NOx regulations and NASA program goals (taken from [1]).

based on the operation of an aircraft in and around an airport (below three thousand feet). It consists of a weighted average of NOx production over a civil transport aircraft's mission as it descends on approach to the time when it attains the same altitude after takeoff [2]. The OEC is referred to as LTO NOx or D_p/F_{00}, and is given by

$$LTO\ NOx = \frac{1}{F_{00}} \sum_{i=TO}^{i=ID} EINOx_i \times w_f \times t_i, \tag{1}$$

where $i = TO$ is the takeoff point and $i = ID$ is the idle point (the LTO cycle consists of takeoff, climb out, approach, and idle), $F_0 0$ is the engine uninstalled sea level static (SLS) thrust, $EINOx$ is the emissions index of NOx (measured in grams of NOx per kilogram of fuel burned), w_f is the fuel flow in the flight regime, and t_i is the amount

Table 1: Time spent at each power setting from the ICAO LTO cycle[3].

Operating Mode	**Thrust Setting** (% of F_{00})	**Time in Mode** (minutes)
Takeoff	100	0.7
Climb Out	85	2.2
Approach	30	4.0
Idle	7	26.0

of time spent in the aforementioned regimes. This is shown in Table 1.

The goals in Figure 1 are specified at an overall pressure ratio (OPR) of 30, and represent NOx reduction targets determined during CAEP meetings (Committee on Aviation Environmental Protection). Since higher OPR engines tend to produce both higher thrust and higher NOx, emissions regulations are a function of the OPR of the engine. Importantly, these emissions constraints impose hard restrictions on the engine manufacturer since an aircraft engine will not be allowed in service if it does not meet the emissions requirements. Therefore, oxides of nitrogen are one of the central drivers of modern combustor design since NOx production considerations greatly restrict the properties of the combustor and partly dictate the conditions under which it can operate. Although NOx production is a very important consideration, numerous other targets must be satisfied as well, making things more difficult since combustor and overall engine design requirements typically conflict (as shown in Table 2).

As an example, it is essential for a cycle designer to attain a high turbine inlet temperature since this effectively dictates the overall engine efficiency for a given cycle. However, high turbine inlet temperatures (T_4) produce higher NOx due to higher combustion (flame) temperatures (T_f). NOx can be lowered at the expense of increased carbon monoxide (CO) and unburned hydrocarbon (UHC) concentrations, in addition to considerable stability and acoustics issues. Therefore the designer must be cautious and clever when developing a new combustor technology due to

Table 2: Conflicting requirements encountered during combustor design (adapted from [4]).

Design Requirement	**Conflicts**
Maximum combustion efficiency	NOx
Minimum pressure loss	PF, NOx, smoke, turbine backflow margin
Combustor exit temperature quality	ΔP, combustor length and channel height, number of fuel nozzles
Cool combustor walls, long life	PF, low power CO and HC
Large operability envelope	Smoke, ignition, stability, PF
Minimum NOx and smoke	CO, HC, ignition, stability, complexity
Simple construction, light weight, and low cost	Long combustor life, PF variations, emissions
Minimum size and length	Operability envelope, PF
Minimum number of fuel nozzles, simple fuel injection system	PF, ignition, stability, emissions

the caveats which arise when trying to satisfy the ever more stringent requirement for engines to produce lower NOx levels. With all of this in mind, new combustor technology development is necessary since current engine technology does not achieve the desired NOx reduction in combination with other targets (namely turbine inlet temperature, pressure ratio, etc.) for future goals.

1.1 Combustor Design

Not all thermochemical propulsion systems produce NOx. For example, thermochemical rockets do not, since the hydrogen and oxygen sit on-board and mix in a chamber which is not exposed to any nitrogen. However, NOx production is unavoidable in air-breathing propulsion systems since the incoming air is a heterogeneous mixture of oxygen and nitrogen (among other species). As a matter of fact, diatomic nitrogen makes up the largest portion of air which is taken in by the engine during service (approximately seventy-eight percent). This would not be a concern if the air was composed purely of oxygen, but it is virtually impossible to allow air into the engine without taking in any amount of N_2. This nitrogen creates new pathways for the reactions to take place, and reacts with the fuel and oxidizer mixture to create these NOx molecules which are ever present, and dangerous to the atmosphere and people (Section 1.3).

On-board oxidizers are too heavy and expensive for gas turbine engines, so one must rely on other types of technology to reduce NOx formation. From the standpoint of combustor architecture, strategies to reduce NOx formation often involve fuel and air staging, which controls where, when, and how fuel mixes with the air and reacts. Many types of combustor concepts have already been developed and are in use (Chapter 2), and new combustor designs *rely* on data from these previous concepts, since these combustors have already been built, tested, and are in operation. Therefore, when designing a new combustor, preliminary sizing is done using these data in various forms. Initially, combustor data from a given type (or class) are used along with a sizing model to develop a new design [5]. Preliminary NOx emissions estimates are determined using EINOx correlations from previous concepts, as well [6].

For instance, consider a "conventional" combustor design (a combustor which has already been designed and belongs to a specific class, e.g. single annular combustor (SAC), dual annular combustor (DAC), etc.) These data and a sizing model are used in combination to determine the initial size of the new burner. Different sizing models exist, the most popular being reaction controlled systems, mixing controlled systems, and evaporation controlled systems [2]. The most widely used is the reaction controlled system, with the the stirred reactor model being the most widely used submodel, shown in Equation 2.

$$\eta(\theta) = f[\frac{P_3^n V_c e^{T_3/300}}{\dot{m}_A}] \tag{2}$$

The variables on the right hand side of the equation (T_3, P_3, etc.) are experimentally swept through to create efficiency data, which are correlated into functions of efficiency, $\eta(\theta)$ (the left hand side of the equations). For example, given a specific combustor volume, reference area, and reference diameter, the pressure, temperature, and air flows can be swept through to develop the correlations. One then "sizes" the combustor (its volume, reference area, and reference diameter) by picking an efficiency value and utilizing the correlations to satisfying the above equation at given conditions. This is shown graphically in Figure 2. Figure 3 displays a typical combustor design space.

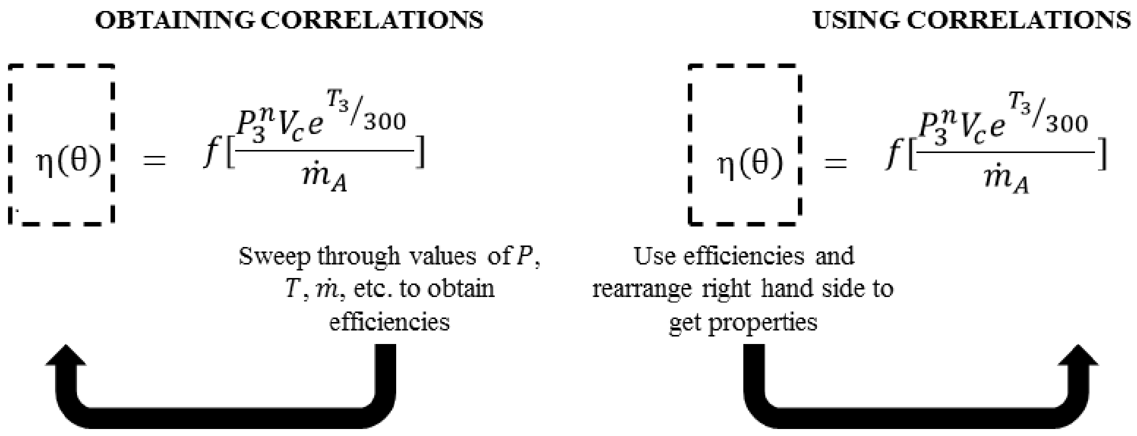

OBTAINING CORRELATIONS
USING CORRELATIONS
$\eta(\theta) = f[\frac{P_3^n V_c e^{T_3/300}}{\dot{m}_A}]$
$\eta(\theta) = f[\frac{P_3^n V_c e^{T_3/300}}{\dot{m}_A}]$
Sweep through values of P, T, $\dot{m}$, etc. to obtain efficiencies
Use efficiencies and rearrange right hand side to get properties

Figure 2: Processes for first developing correlations (left) and then using these correlations to develop a new combustor design (right).

1.2 Advanced Concepts

The simultaneous drives for lower SFC, NOx, and smoke demand the development of advanced concepts since future goals aim to achieve lower fuel consumption with higher thrust, leading to engines with higher OPRs. This directly influences NOx since higher OPR engines lead to higher combustion pressures as shown in Figure 4. These higher pressures also affect other thermodynamic properties such as temperature, which also increases NOx production via the Zeldovich mechanism, currently

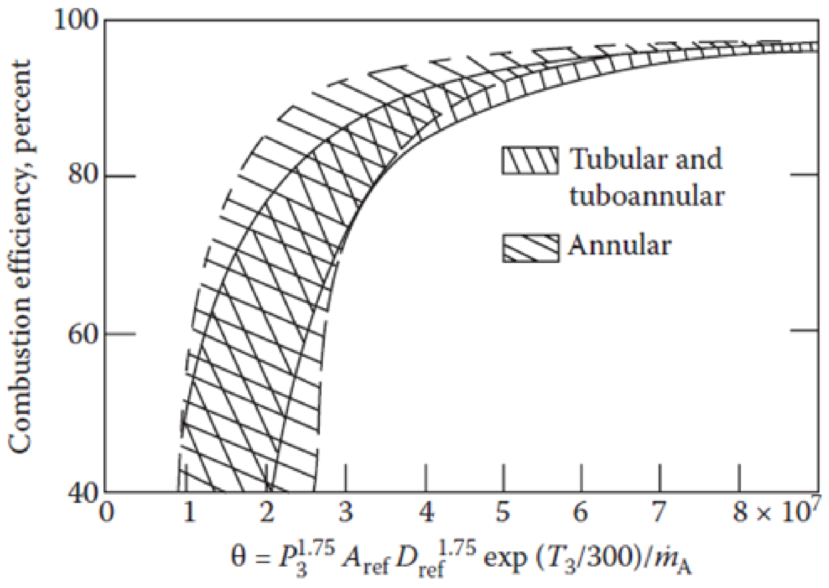


Figure 3: Design chart for conventional combustors (taken from [2]).

the most prominent mechanism in NOx production for high temperature environments [7]. This promotes a push towards new combustor technology, which must be engineered so that lower NOx can be achieved under these non-ideal, high pressure conditions.

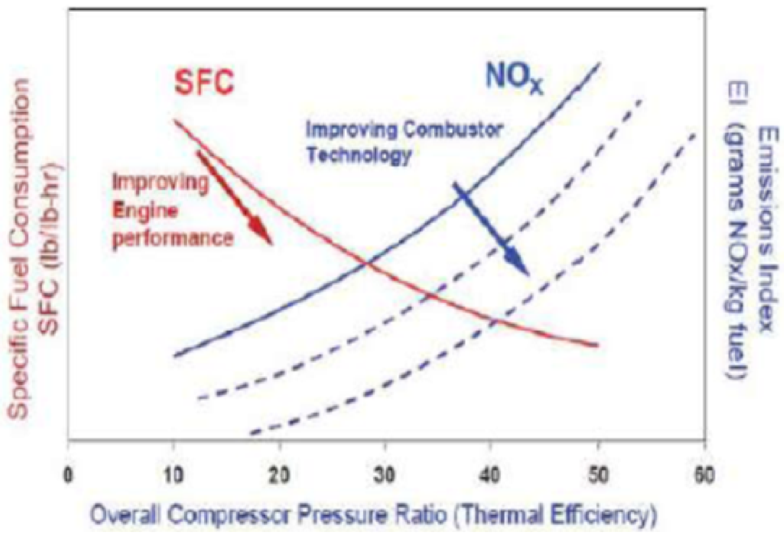
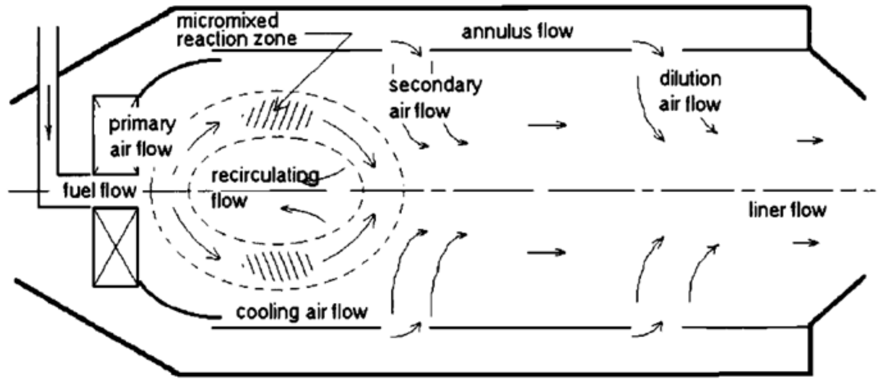


Figure 4: Impact of overall pressure ratio on NOx formation (taken from [8]).

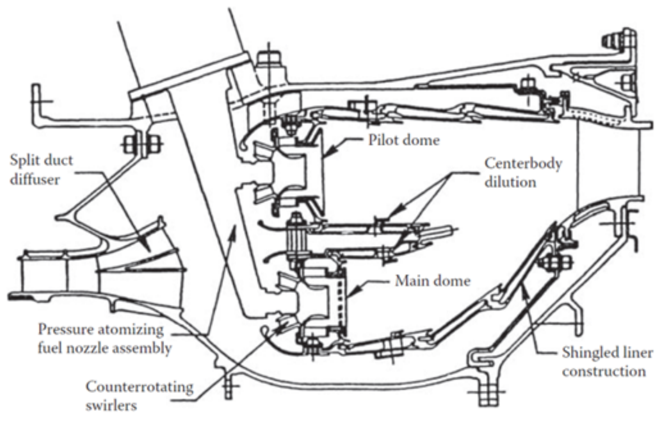
The fact that correlations are used as a starting point for preliminary combustor sizing is a direct result of the fact that *no* purely analytical sizing methods presently exist within the field of combustor design. These correlations effectively take a previous combustor concept and "size" a new one using information about previous technology and configurations. For instance, the correlation Equation 2 scales a specific type of combustor for a new application, but is *not* capable of *sizing* a combustor which utilizes different technology. This is due to the fact that they are not equations which rely on first principles. This is easily understood, since these correlations were developed using test data from specific combustor architectures. Therefore, sizing a new combustor using known methods will rely on designing, building, and testing numerous combustors for each concept. This is an extremely expensive endeavor, since it involves developing and constructing numerous concepts just to obtain correlations before preliminary sizing even occurs!

With this in mind, there are concerns when it comes to the development of brand-new burners. Specifically, how will the preliminary sizing be done if there are no correlations to utilize for the specific architecture? The new design will not have been tested under different conditions since it does not yet exist, and testing is an *extremely* expensive proposition. Furthermore, the designer would like to know as much as possible about the combustor and its design space as early on in the design process as possible (ie., the combustor designer would like to know if the design of a new concept low NOx combustor will be feasible based on all of the aforementioned criteria for design (Figure 2), and preferably with as little testing as possible). Of specific interest in this work will be whether or not the combustor is capable of hitting its NOx target. With the additional constraint that one does not want to rely so heavily on testing, a new method must be developed which will allow one to save time and money by rapidly performing trade studies and assess whether an advanced combustor concept will be viable based on its physical properties and emissions production.

Aside from the sizing aspect, the preliminary design of combustors requires information for emissions (specifically NOx in our case), as well. As with sizing rules, these NOx production estimates come from correlation equations regressed from previous data. Oxides of nitrogen are extremely difficult to predict due to the many variables and interaction terms which contribute to NOx formation. For instance, consider Figure 5.



(a) Main burner flow pattern for a conventional combustor (taken from [9]).



(b) General Electric DAC (Dual Annular Combustor, taken from [2]).

Figure 5: Different combustor architectures with the same bulk properties will produce different NOx emissions.

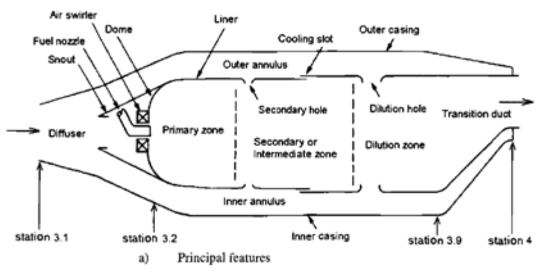
Temperature and pressure were stated above as contributors to NOx production. Although these influence NOx greatly, they are not the only drivers. To name a few, NOx production is *heavily* determined by

- Temperature
- Pressure
- Residence time (τ_{res})
- Fuel droplet diameter, typically measurd as the sauter mean diameter (SMD)
- Equivalence ratio in the combustor (ϕ)
- Heating value of the fuel, typically referenced as the lower heating value (LHV)
- Geometry of the combustion chamber
- Mixing of fuel within the combustion chamber (how homogeneous the fuel and air is mixed)
- etc.

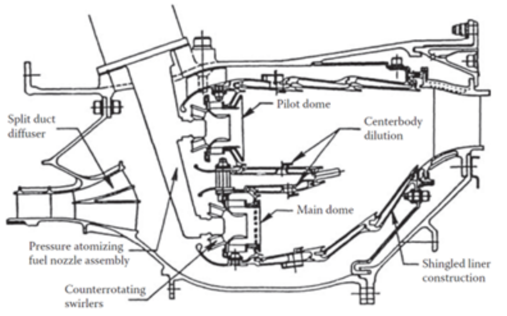
and their interaction terms (e.g. high pressure and high temperature will increase the reaction rate more than just high pressure or high temperature alone). The last few bullets show the importance of geometry and aerodynamics on NOx production, which complements the argument in Figure 5. Two combustors of different architectures can produce *vastly* different NOx emissions even though their bulk thermodynamic properties are the same (e.g., their inlet temperature (T_3), inlet pressure (P_3), and equivalence ratio (ϕ) can all be the same, yet the two different combustors produce different NOx emissions). The DAC in Figure 5(b) will produce less NOx than that of the conventional SAC combustor depicted in Figure 5(a) due to a combination of properties related to the geometry and amount of mixing within the combustor.

Many different combustor architectures exist, all aimed at improving different aspects of the combustion process. A few concepts aimed at lowering NOx are shown in Figure 6.

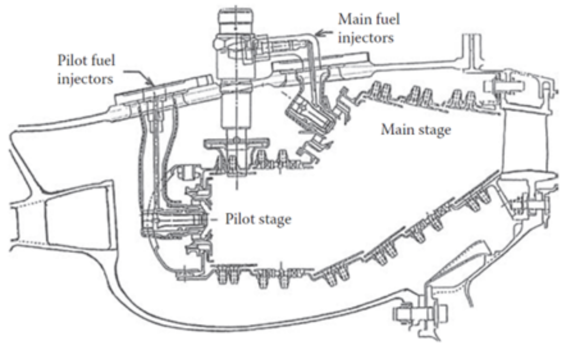
As a matter of providing an example for statements made above, one may be unable to compute different levels of NOx emissions if not using the correct correlations.



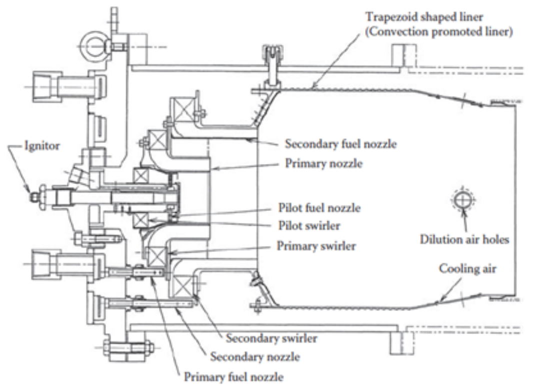
(a) Single Annular Combustor (SAC, taken from [9]).



(b) General Electric DAC (Dual Annular Combustor, taken from [2]).



(c) Pratt and Whitney Axially Staged Combustor (taken from [2]).



(d) Dry-low NOx combustor concept (DLN, taken from [2]).

Figure 6: A small sample of combustor concepts which were developed to lower NOx emissions.

For example, consider the SAC combustor in Figure 6(a). To size a SAC combustor for a new application, correlations for a SAC combustor are used to determine the NOx emissions. These hold for a SAC and *only* a SAC. Therefore, if one wants to size a DAC (Figure 6(b)), DAC correlations must be used. If using SAC correlations for a DAC, the NOx emissions predictions will not be correct even though the bulk properties (pressure, temperature, etc) for the SAC and DAC may very well be the similar or the same!

To further illustrate the point, typical NOx correlation equations are shown in Equations 3 through 6 [2].

$$NOx = f(T_3, P_3, \phi) \tag{3}$$

$$NOx = 9 \times 10^{-8} P^{1.25} V_c e^{0.01 T_{st}} / \dot{m}_A T_{PZ} \quad [g/kgfuel] \tag{4}$$

$$NOx = 29 e^{-21,670/T_c} P^{0.66} \times [1 - e^{-250\tau}] \quad [g/kgfuel] \tag{5}$$

$$NOx = 15 \times 10^{14} (t - 0.5 t_e)^{0.5} e^{-71,100/T_{st}} P^{-0.05} (\Delta P/P)^{-0.5} \quad [g/kgfuel] \tag{6}$$

The correlation in Equation 3 involves only temperature, pressure, and equivalence ratio, which is most popular for a combustor which has already been built and tested and whose inner workings are attempted to be captured via a correlation. Since this combustor is typically already sized, this form of a correlation is not very popular to use to determine NOx during the sizing phase of a *new* combustor. Normally, one will use a NOx correlation in the forms of Equations 4 through 6, since these involve not only temperature, pressure, and equivalence ratio, but also some other aspects more specific to the architecture of a given combustor. For instance, Equation 4 utilizes the combustor volume, Equations 5 and 6 utilize the residence time.

When developing a new combustor architecture, it is important to be able to quickly develop a new design and determine if said design is viable (by eliminating as much uncertainty as possible) before moving to the next stages. Understanding a design and its behavior early on is important so as not to waste time and money and invest in large scale simulations and testing before one has any idea if the concept is capable of achieving the goals. When emissions is a main driver of the design, this becomes a difficult multidisciplinary problem.

1.3 Theoretical Considerations

1.3.1 Reaction Pathways

One of the many reasons why aircraft NOx emission prediction is such a difficult problem stems from the fact that the chemical kinetics processes behind NOx production from a jet fuel/air reaction are complicated and involve so many species. As a matter of fact, determining the concentration of any non-equilibrium species is a relatively difficult problem, indeed. Stepping away from NOx production for a moment, consider only the formation of water. For the equilibrium reaction, we have

$$2H_2 + O_2 \longleftrightarrow 2H_2O \tag{7}$$

Here, we can algebraically determine the concentrations of hydrogen, oxygen, and water would be (if we only consider those species). In reality, however, this reaction proceeds according to the time dependent laws of kinetics, and involves many more intermediate reactions. For example, consider Figure 7, which contains many intermediate species and radicals (e.g. H and OH), and interactions (M can be an intermediate species, radical, or interaction term[1]).

Reactions can not proceed quickly without chain branching steps (the creation of radicals which act to move the reaction along). At equilibrium, the concentration of these intermediate species will be small, however stopping the reaction early (perhaps

[1]For example obtaining energy from a collision with a wall.

$$
\begin{gathered}
H_2 + M \leftrightarrow H + H + M \\
H_2 + O_2 \leftrightarrow HO_2 + H \\
H + O_2 \leftrightarrow O + OH \\
O + H_2 \leftrightarrow H + OH \\
H_2 + OH \leftrightarrow H_2O + H \\
H + O_2 + M \leftrightarrow HO_2 + M \\
H + OH + M \leftrightarrow H_2O + M
\end{gathered}
$$

Figure 7: Example kinetics mechanism of a hydrogen/oxygen reaction (taken from [10]).

via quenching) will cause the system to contain a larger amount of these intermediate species. Lastly, it is very important to note that this is only *one* possible reaction pathway. In reality, H_2O can be formed not only by these sets of reactions, but by many other chemical reactions, as well (depending on what else is around). This makes the task of modelling H_2O production a very difficult one.

Coming back to combustion systems, chain branching reactions are very important because it allows the reaction to proceed and heat to be released (the entire purpose of the combustion system). Since the equilibrium concentration of NOx is too high for emissions requirements, the combustor is specifically designed such that its characteristic residence time is not long enough to allow equilibrium concentrations of NOx molecules to form. As a matter of fact, thermal NOx levels are typically *much* lower than their equilibrium concentrations [11].

As seen with the water example, even the simplest chemical reactions involve numerous intermediate species. Jet fuel (specifically Jet-A, which is used in commercial aviation in the United States), it is found to contain numerous different chemicals and compounds [12]. These conventional petroleum-based jet fuels contains an average of about 60% paraffins, 20% napthenes, 20% aromatics, about 500 parts per million (ppm) sulfur, in addition to various other contaminants [2], as shown in Figure 8.

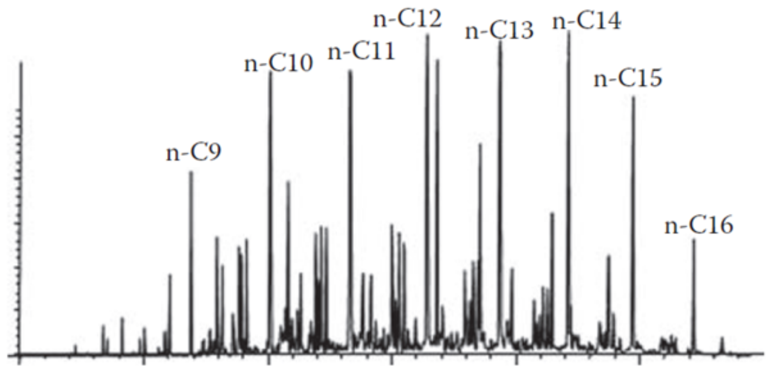


Figure 8: Gas chromatograph of jet fuel (taken from [2]).

Just considering paraffins (saturated hydrocarbons with single bonds), one can see from the figure that there is a widespread variation in starting composition of the fuel. These paraffins will break down into progressively smaller molecules during reactions, including various types of radicals. These will all interact with the chemicals in the air (again, a non-homogeneous mixture of oxygen, nitrogen, and other molecules). Each of these smaller reactions will have their own reaction rates, further producing different molecules during reaction, as well. This doesn't even take into account double bonded hydrocarbons, triple bonded hydrocarbons, molecules in different molecular formations (such as benzene rings, etc), and other species produced during reactions with these other chemicals. It is not hard to realize that there are virtually thousands or tens of thousands of different reaction pathways through which these chemical reactions can proceed, depending on the thermochemical conditions under which they meet and interact.

NOx production is commonly modeled using four mechanisms, the thermal (or Zeldovich) mechanism, Fenimore (or prompt mechanism), the N_2O-intermediate mechanism, and the NNH mechanism [2]. The extended thermal NOx mechanism is composed of three reactions, as shown in Equation 8. It is the dominant mechanism by which NOx is produced in high temperature environments and wide ranges of equivalence ratio, making it the focus of NOx modeling for standard rich burn combustors.

$$\begin{aligned} O + N_2 &\longleftrightarrow NO + N \\ N + O_2 &\longleftrightarrow NO + O \\ N + OH &\longleftrightarrow NO + H \end{aligned} \tag{8}$$

The prompt NOx mechanism is particularly important in hydrocarbon combustion since it produces NOx very rapidly. This mechanism commences before the thermal mechanism even has a chance to create NOx, affecting all types of combustors which burn hydrocarbon fuels (e.g. rich burn, lean burn, RQL, etc.) It is caused by hydrocarbon radicals reacting with molecular nitrogen to form amines and cyano compounds. These molecules are then converted to intermediate compounds and go on to produce NO [2]. The rate limiting steps of the prompt NOx mechanism are given in Equation 9, which ignores the processes that forms CH radicals.

$$\begin{aligned} CH + N_2 &\longleftrightarrow HCN + N \\ C + N_2 &\longleftrightarrow CN + N \end{aligned} \tag{9}$$

There are an additional set of four reactions which become important when $\phi \lesssim 1.2$, but are not listed here [2].

The N_2O-intermediate mechanism is important in lean burn environments (typically $\phi \leq 0.8$), and therefore low temperature conditions [2]. This is due to the fact that many other NOx production routes and pathways open up in this low temperature kinetics regime. This mechanism is displayed in Equation 10.

$$\begin{aligned} O + N_2 + M &\longleftrightarrow N_2O + M \\ H + N_2O &\longleftrightarrow NO + NH \\ O + N_2O &\longleftrightarrow 2NO \end{aligned} \tag{10}$$

Lastly, the NNH mechanism consists of the two key steps shown in Equation 11. It is particularly important for fuels with large carbon-to-hydrogen ratios (e.g. alkynes).

$$\begin{aligned} N_2 + H &\longleftrightarrow NNH \\ NNH + O &\longleftrightarrow NO + NH \end{aligned} \tag{11}$$

One of the most important points in this discussion is the fact that are the *interactions* between the chemical mechanisms as exemplified in Figure 9. Here, the Zeldovich mechanism has coupled with the prompt mechanism to cause a more complicated NOx production and destruction interaction.

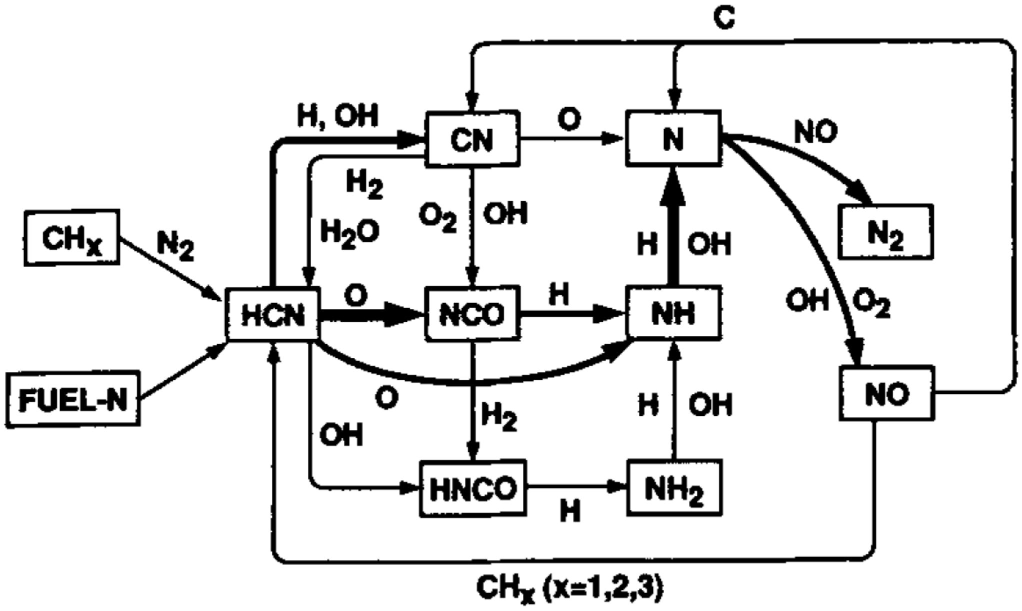


Figure 9: Production of NO associated with the Fenimore prompt mechanism (taken from [13]).

Indeed this is quite typical of NOx mechanisms and chemical kinetics in general: each species can interact with each other species and complicate the chemical scheme tremendously. This makes NOx emissions modeling a difficult and expensive task, even just when looking at the sheer number of molecular and atomic inter dependencies. Additionally, some species are changing much more rapidly than others, leading to mathematically stiff equations [14]. It is quite a bit easier when dealing with small molecule fuels and a single oxidizer, but this is not the case for Jet-A, as was shown in Figure 8.

A multitude of molecules open up the reaction mechanisms to interact with more species other than just the simple radicals given in Equations 8 through 11. In the past, NOx emissions from combustors have been dominated by the thermal NOx mechanism due to the high temperatures within the reaction chamber (for example, the rich burn chamber in an RQL combustor). With lean burn combustors becoming more popular however, this may not always be the case, and these other mechanisms may have to be taken into account during combustion modeling. Here, mechanisms such as prompt NOx and the N_2O-intermediate mechanism will play larger roles. Looking at Equation 10, one of the main reactions involves "M", which can be any molecule, radical, or intermediate species willing to push a reaction forward or even quench a reaction early. Although small step mechanisms may have given good NOx predictions in the past, this may not be the case for advanced combustors which operate in different thermochemical regimes.

1.3.2 Residence Time

Another key combustor characteristic is residence time, which is a large contributor to NOx formation due to reaction rate kinetics. Standard reaction rate laws have the form

$$RR = \frac{d}{dt}[W] = k \times \frac{[Y]^c[Z]^d}{[W]^a[X]^b} \tag{12}$$

for the generic reaction

$$aW + bX \longleftrightarrow cY + dZ, \tag{13}$$

where W is the species of interest (NOx in this case), X, Y, and Z are the species which interact to form the species of interest, k is the reaction rate, and a, b, c, and d are the reaction rate orders. Therefore, integrating Equation 12 shows that the concentration of a given species (NOx in this case) is a time dependent quantity. Typically, a short residence time is desired in order to keep NOx to a minimum, however this directly impacts the efficiency, since short residence times impede the oxidation of CO to CO_2 (the intermediate reaction with radicals which releases the majority of the chemical energy in the fuel [7]). However, allowing the reaction to proceed too long is dangerous since more NOx will be produced, and these higher values are never desired. Thus, a lot of work is done to allow reactions to proceed far enough for the important heat releasing oxidation reactions to ensue, but finish before the NOx levels become too high.

Residence times are sometimes taken into account in NOx correlations (Equations 5 and 6), however these must be measured (from experimentation or CFD). Different burner concepts will have different aerodynamics to consider and will therefore need to have different residence time correlations. This is due to the fact that the combustor is desiged to undergo flameholding which purposefully causes mixing and backflow as shown in Figure 10.

This is not only a function of aerodynamics, but also the geometry of the combustor, since the flame zone size is tied to the air flow pattern through the burner. Indirectly then, the chemistry becomes tied to the actual combustor design. One mus then rely on CFD to get accurate estimates for residence time.

1.3.3 Droplet Size and Mixing Effects

Fuel injector design is an immensely important aspect as well, since it directly impacts fuel droplet size. Fuel droplet size is an important parameter since fuel and oxidizer only react in the gas phase, meaning that any liquid fuel which enters the

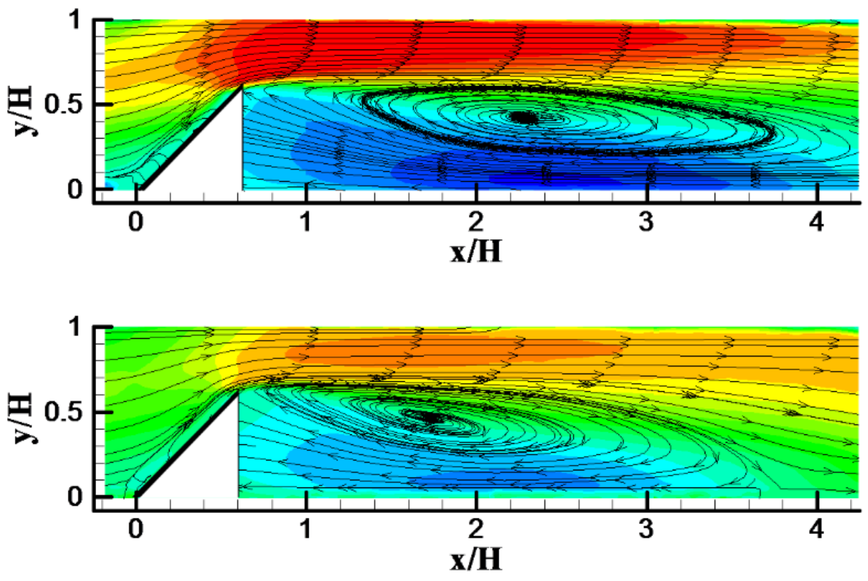


Figure 10: Recirculation regions. Combustors are designed to recirculate flow so that hot products mix with incoming gases and ignite, which is the process of flameholding (figure taken from [15]).

chamber must first evaporate before it reacts. Although certain types of technology to bypass the evaporation stage within the chamber indeed exist, including prevaporized combustors and premixed-prevaporized combustors [16, 2], common combustor design and operation includes storing liquid fuel and injecting and swirling before burning. The process of injecting and swirling leads to atomization (and supports mixing) so that the liquid droplets are small enough to evaporate quickly [17], since evaporation time is proportional to the square of the droplet size [18, 2]. The big problem here is that if the fuel droplet is large, it will burn as it evaporates which causes stoichiometric burning resulting in high temperatures and therefore high thermal NOx concentrations due to the Zeldovich mechanism (see Section 1.3.1). If the fuel is atomized well, it can mix with the incoming air and become a premixed flame and burn at other flammable equivalence ratios.

Liquid droplet evaporation is an entirely rich field in itself, and is the focus of many works. Much research is conducted on fundamental principles of liquid droplet evaporation and sheet breakup, as well as applications to gas turbine combustor design. Future research on fuel atomization and NOx reduction occurs regularly, including the development of fuel injection models for CFD and other tools which may help future NOx reduction concepts. This very brief overview is included for completeness, however fuel injection does not play a large role in the current work.

CHAPTER II

PREVIOUS WORK

Section 1.1 introduced the fact that correlations are used in the preliminary design phase for initial combustor sizing and also for emissions prediction, which is the first step in the development of a new combustor. The entire process of combustor design is shown in Figure 11, using example tools from Rolls-Royce.

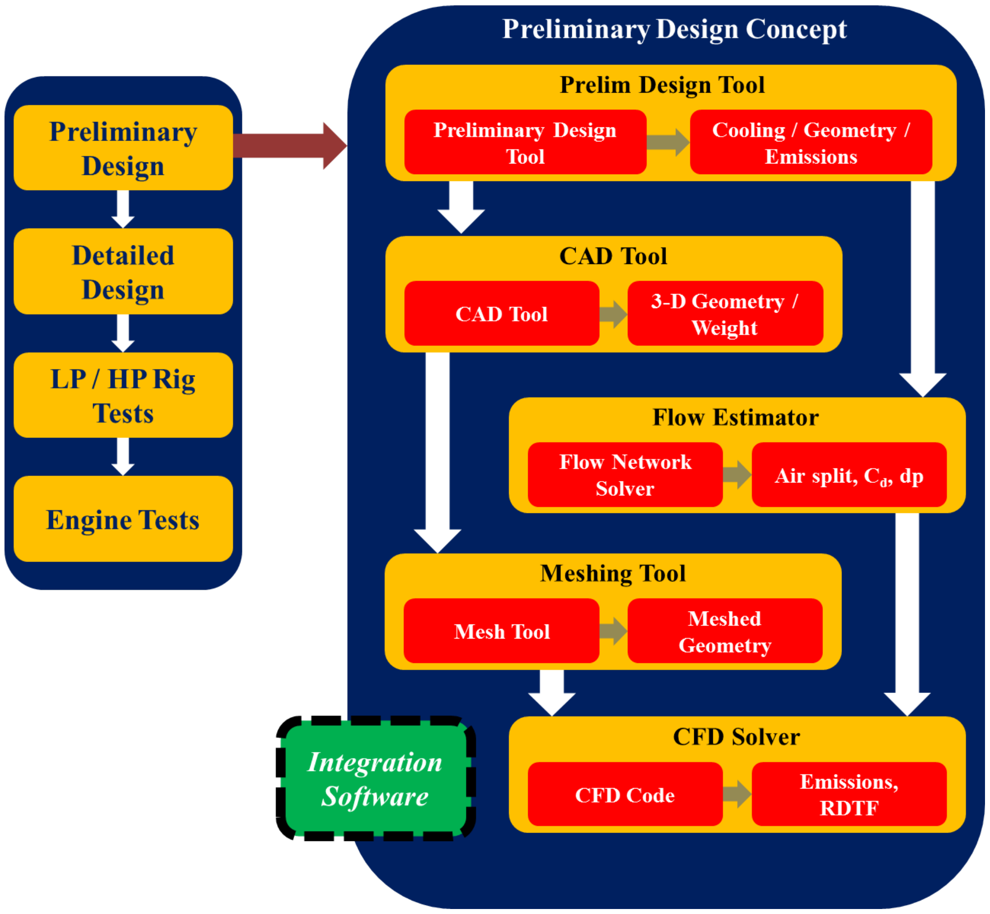
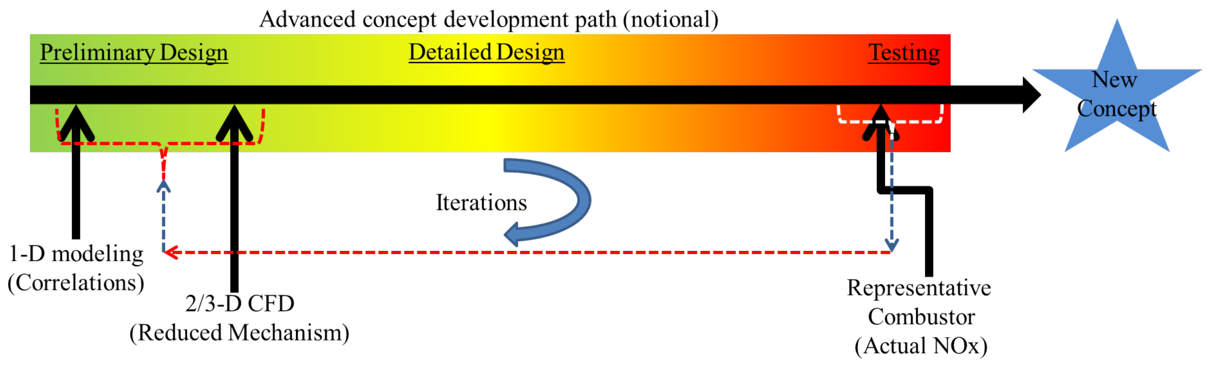


Figure 11: Standard design process [left] and the preliminary combustor design sub-process [right] (adapted from [19]).

The unified process depicted on the right hand side of Figure 11 is the preliminary design of a combustor concept. The top right of the figure shows that the first estimates of geometry and NOx are among the inaugural steps (carried out in the form of correlations and low order models). As preliminary design proceeds, a large amount of computational power is required before the detailed design phase is even started, since CFD is the next step in the PD phase. The preliminary design process is a long and expensive one, which has only relatively recently been automated [19, 20, 21].

Sizing and emissions prediction in the preliminary design phases is not a new problem, and correlations are currently the standard practice [22, 2]. As noted in previous sections, these correlations only hold for the same class of a combustor [22], so these correlations do yet not exist for an advanced concept. In order to obtain these correlations, one will have to bring the concept all the way through to the testing phase, then go back to preliminary sizing step and iterate. This is a costly venture, especially when one considers the fact that the advanced concept may not even meet the NOx targets when tested. It is important to note that just within the PD phase, there is a high discrepancy between the preliminary estimates of NOx and those computed during CFD [19]. Furthermore, the NOx emissions obtained during an engine test will also be different than both the preliminary estimates via correlations and those obtained via CFD, since CFD is typically run with reduced order mechanisms to make the calculation viable based on computer resources and vast temporal considerations [23] (see Section 2.4). With current methods, the only way to develop a new combustor is to test the engine to get correlations and then size it accordingly, or to develop a new concept which is constrained to behave similarly to one previously developed. A claim is then made that an inexpensive method for the design and development of an advanced concept combustor from scratch does not exist. Figure 12 depicts the problem in pictorial format.

There is a definite need to be able to more *accurately* predict emissions earlier on in the design process for concepts about which one has little or no information



Advanced concept development path (notional)
Preliminary Design
Detailed Design
Testing
New Concept
Iterations
1-D modeling (Correlations)
2/3-D CFD (Reduced Mechanism)
Representative Combustor (Actual NOx)

Figure 12: Standard iteration procedure for developing a new combustor. Iterations are performed numerous times through the process within each of the three main design segments. The only iteration shown in the figure is one from testing back to preliminary design, which is displayed for illustrative purposes. Iterations are performed because more knowledge is gained further down in the design process, and is used to modify the way preliminary design is done.

(besides certain targets such as thrust and emission targets). As an additional constraint, these methods and tools must be able to evaluate the combustor design very quickly (specifically, what does the flow field look like inside the chamber, and will the design achieve the NOx targets?), so that one does not have to wait a long time to determine if an advanced concept is viable. This allows consideration of many competing concepts. Many researchers have tried to attack this problem which resulted in the formulation of multiple tools and methodologies. These tools have a range of fidelity as shown in Figure 13. Note however, that the intentions of the tools also vary. For instance, some have proposed the use of flow networks alone [24] (without the ability to predict emissions, but solely to obtain aerothermodynamic properties). However, the ability to procure emissions goals is of primary concern in this work (due to their growing austerity), so special interest is taken on the approaches which are directly oriented to achieving this objective, and/or contain this capability.

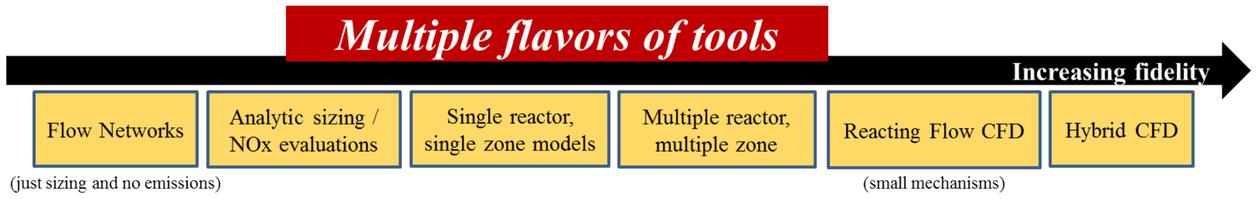

Multiple flavors of tools
Increasing fidelity
Flow Networks
Analytic sizing / NOx evaluations
Single reactor, single zone models
Multiple reactor, multiple zone
Reacting Flow CFD
Hybrid CFD
(just sizing and no emissions)
(small mechanisms)

Figure 13: Multiple tools have been developed to size and analyze combustors. They exist at varying levels of fidelity.

2.1 Flow Network Approach

Strict flow network approaches alone do not contain the capability to predict NOx by themselves, however, the ability to predict the flow field within the combustor is a very important step. For example, flameholding is largely a function of the flow aerodynamics, in addition to swirler design. Also, the aerodynamic flow field is highly coupled to the heat release during chemical reactions, which makes the determination of the flow field interesting and necessary. Specifically, flow network solvers allow one to determine burner properties such as pressure, temperature, density, etc. These network methods consist of a number of independent subflows which are linked together to form a physical process, and have the ability to model complicated and unusual geometries effectively. This is done with little numerical difficulty, and retains the advantage of rapid execution [24, 25, 26]. Rapid execution is of great benefit, and much work has been done to create design tools using these models by incorporating optimizers. There is still work to be done on this end, as well, since critical details such as combustor relight, flame stability, and pollutant emissions are unable to be captured in the optimization procedure [25].

The idea of a network flow model (shown in Figure 14) is to conceptualize the overall geometric structure of the combustor without requiring the use of geometry restrictive semi-empirical models [24]. Here, elements and nodes are used to simulate the physical features of the domains and join them together, respectively, so that

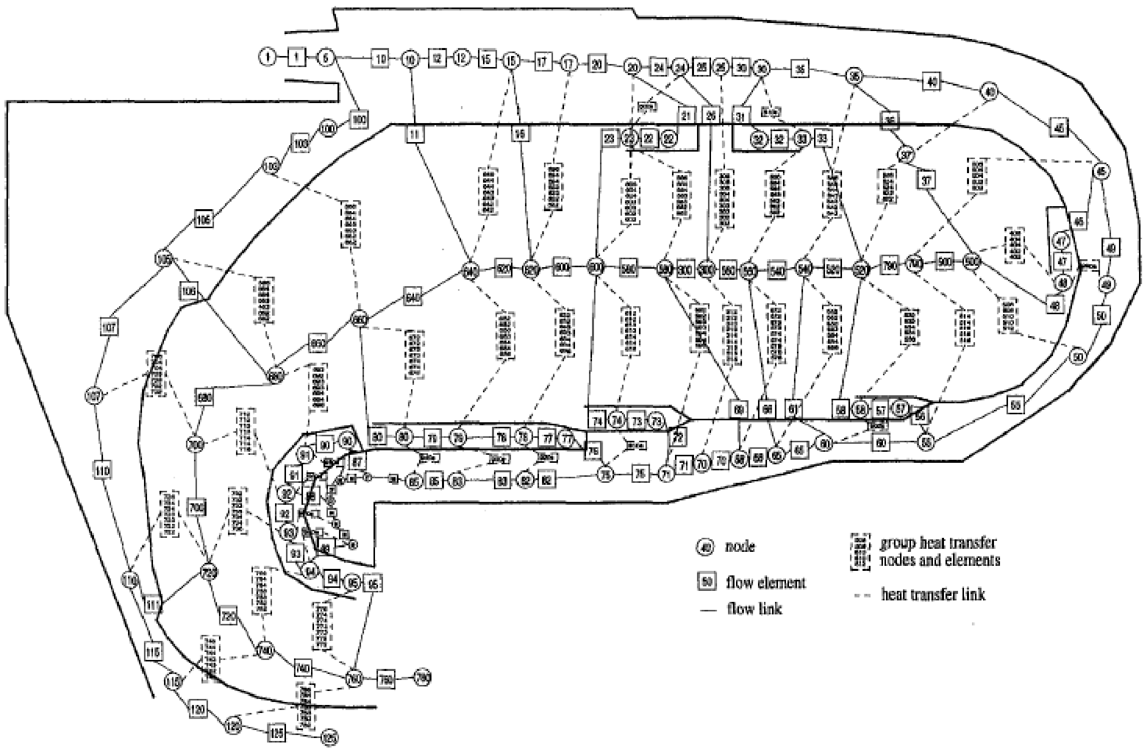


Figure 14: Network diagram for a reverse flow combustor (taken from [24]).

physical laws of continuity are satisfied. The continuity and pressure drop/flow rate relationships are the only equations which are required. However, the functional forms of these relationships are derived from semi-empirical formulations of combustion features [24], making them less generic than one may desire. In effect, these functional forms inherently cause restrictions, and an indirect coupling to previous architectures. Interestingly, versatile radiation models exist for use with network approaches, which were designed to model all conceivable combustor types [5], making it perfect to use for advanced concepts. This takes away the necessity to use correlations for radiative heat transfer [27], which is important since NOx production is highly coupled to flame temperature, (and is in turn is heavily influenced by heat transfer). Heat transfer to the liner and correct liner temperatures are also important when determining wall interaction effects such as CO oxidation at the wall. Although the radiation heat transfer coefficients are based on previous data, this is less important for a new concept and more important when examining new materials for the concept in question.

Though some properties may be captured well, others are not. Empirical correlations are used for simulating physical features such as flameholding [24], which means that some properties rely on previous data and are not calculated using first principles. Therefore, a lot of information about the design is required to create these correlations, and typically comes from CFD or hardware tests [24]. Some have attempted to take NOx into account, but emissions in this case are computed using constrained equilibrium models and correlations [24]. In these approaches, functions of adiabatic flame temperatures are used to compute species concentrations instead of physical kinetics mechanisms which take into account things like residence times. Others have simulated the combustion process and recirculation in different and cunning ways, specifically simulation via one-dimensional stream tubes [28], but this still does not correctly capture the physical kinetics process.

From an emissions perspective, the big issue with this method is that constrained equilibrium calculations are performed to compute the combustion gas temperature [5]. Minor species will be present in the incorrect concentrations since the model is not coupled with a kinetics mechanism. Therefore, the temperature and emissions concentrations will be different than that computed by a kinetics scheme and may lead to a poor emissions predictions. Since the emissions computations are inaccurate, flow networks are not a good tool to understand the viability of an advanced combustor concept.

2.2 Single Reactor, Multiple Zone Models

The major pitfall of the flow network models was the inability to accurately predict emissions. The need to develop a method to predict the amount and composition of exhaust gases from aircraft engines has been identified and attempted by many

[29], and each of these approaches have had varying levels of success. Consequently, reactor models have been developed to better capture the kinetics associated with NOx production. In some cases, these models fit into an overarching simulation environment, such as GSP (Gas turbine Simulation Program [30]). The GSP combustor model takes a multi-reactor approach by stacking a set of thermochemical reactors on top of one another as shown in Figure 15 [29].

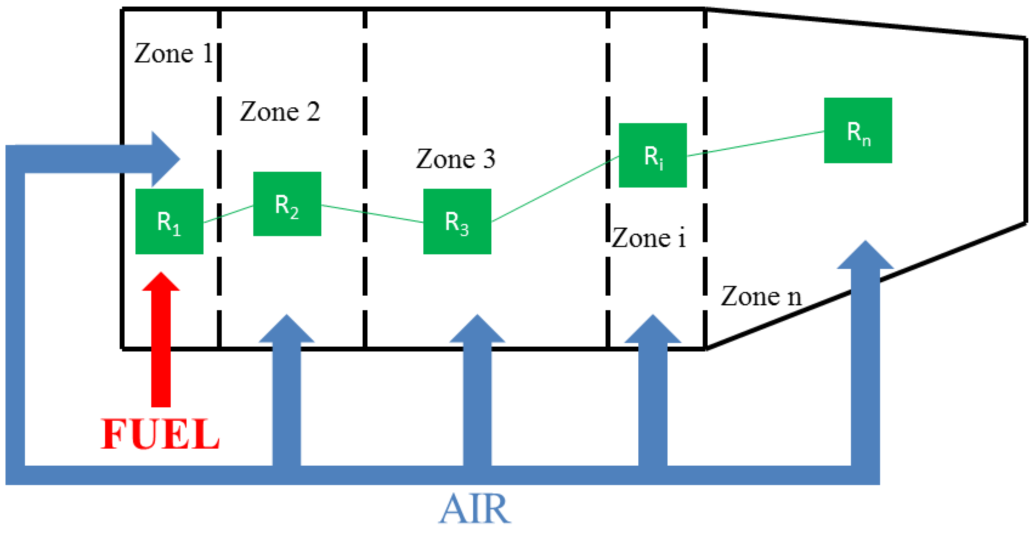


Figure 15: GSP multi-reactor combustor model example.

It is very important to note that this type of model has many shortcomings when used as a predictive tool. First off, it is stated that this multi-reactor model includes the calculation of one-dimensional combustion kinetics, but requires detailed geometrical combustor data to do so [30]. Additionally, only eighteen species are taken into account during reactions (GSP version 8) before chemical equilibrium is used for the calculation. When this equilibrium calculation is done, it only takes into account a five species system [30]. GSP seems to do a great job at predicting entire engine performance [29, 30], but seems to fall short when it comes to predicting emissions.

Leaving the GSP combustor sub-model aside due to the fact that it is only a single implementation, the theory behind this type of reactor model exhibits the fact that it is a step forward from flow networks with semi-empirical correlations. There are three main idealized reactor types, a plug flow reactor (PFR), a partially stirred reactor (PaSR), and perfectly stirred reactor (PSR). The PFR is grounded on a geometrical argument, while the PaSR and PSR is a temporal based reactor. What differentiates the three are the assumptions behind each of the models. These assumptions are highlighted in Table 3. The purpose of these models are to solve for chemical species in a speedy manner, by uncoupling chemical species production and destruction with the flow physics. In the case of the PSR and PaSR, they are completely uncoupled, while they are "mostly" uncoupled in a PFR. This is most easily understood from the standpoint of a PSR, which is treated as a zero-dimensional "volume" with some chemical time spent for reactions to take place.

Coupling the species creation with the entire flow field is quite an expensive undertaking (see Section 2.4). The reactor approach allows one to describe the chemical formation of species with a coupled set of ordinary differential equations (ODEs) or even algebraic equations if running as a steady-state PSR [7]. This is *much* less of a computational burden than solving the coupled set of partial differential equations (PDEs) associated with reacting-flow CFD (RFCFD), since spatial variables are taken out of the equation (in a PSR and PaSR). With this being said, the reactor model approach still has its pitfalls. For example, the single reactor multiple zone configuration shown in Figure 15 only simulates changes in one flow direction. Though it may be useful in some cases to treat the gas-turbine combustor as a simple series of PSRs, this approach will not always yield ideal results for many reasons. First and foremost, it is quite obvious that all of the fuel and air does not concentrate itself in a one-dimensional formation and mix perfectly. Though this is a useful approximation [29], it does not actually represent the real physics of the problem since there are numerous other effects within the combustor.

In this type of model, mixing, flameholding, wall interactions, multidimensional

Table 3: Summary of idealized reactor types used in combustor models and their important assumptions [7].

Reactor	Important Assumptions
Perfectly Stirred Reactor (PSR)	• All species entering reactor immediately and completely mix relative to the mean residence time of the flow through the volume [31] ($Da \to 0$). • Mixture is homogeneous in all areas of the control volume. • Mixture reacts within the volume for some residence time.
Plug Flow Reactor (PFR)	• Only flow is in the axial direction and there is no mixing in this direction. • One-dimensional flow. • Reactor flow properties and composition are a function of the one-dimensional position, x.
Partially Stirred Reactor (PaSR)	• Reaction *not* limited by a mixing time scale, but by both mixing time and kinetics times cales ($0 \leq Da \leq \infty$). • There is a maintained compositional difference between the inlet stream and the contents of the reactor [32], which leads to a steady-state of ”unmixedness”.

effects, and many other thermodynamic and geometric aspects are not taken into account with any of the breadth or detail which is needed for simulating a combustor's operation. These are important, especially for new low-NOx concepts of interest. Additionally, fuel flow will typically be low, as well, making it hard to sustain a flame. For conventional combustors of the past decades, the NOx production was much higher than new targets, so if one was off by a few (or even a few tens) of grams of NOx per kilogram of fuel burned, this might only be a few (or a few tens) percent different. If you are off by the same amount and the NOx production is low however, this could mean only simulating NOx production to some "close" order of magnitude. In order to improve this model, other physical effects must be taken into account.

2.3 Multiple Reactor, Multiple Zone Models

Many physical effects which contribute to NOx formation (e.g. droplet size and mixing) are not taken into account in SRMZ models. Therefore, work has been done since the 1970s to develop reactor models which can facilitate flow patterns for recirculation, low fidelity mixing effects (based on turbulence theory), fuel droplets, etc. [33]. Although many of these were submodels and were not computed from first principles (they were either input based on experimental tests or very simplified models), this approach showed reasonable results for existing combustor concepts when compared to experiments. With the idea of creating a higher fidelity model, each individual reactor from the SRMZ model is presently replaced with parallel arrays of like-reactors in order to more accurately simulate fuel/air unmixedness on chemistry while simultaneously decreasing execution times (as compared to RFCFD) [34]. This allows one to partition total inflow according to an equivalence ratio distribution [23]. The approach is shown in Figure 16, where eleven reactors are used for the first two zones and a plug flow reactor for the burnout of the fuel/air mixture.

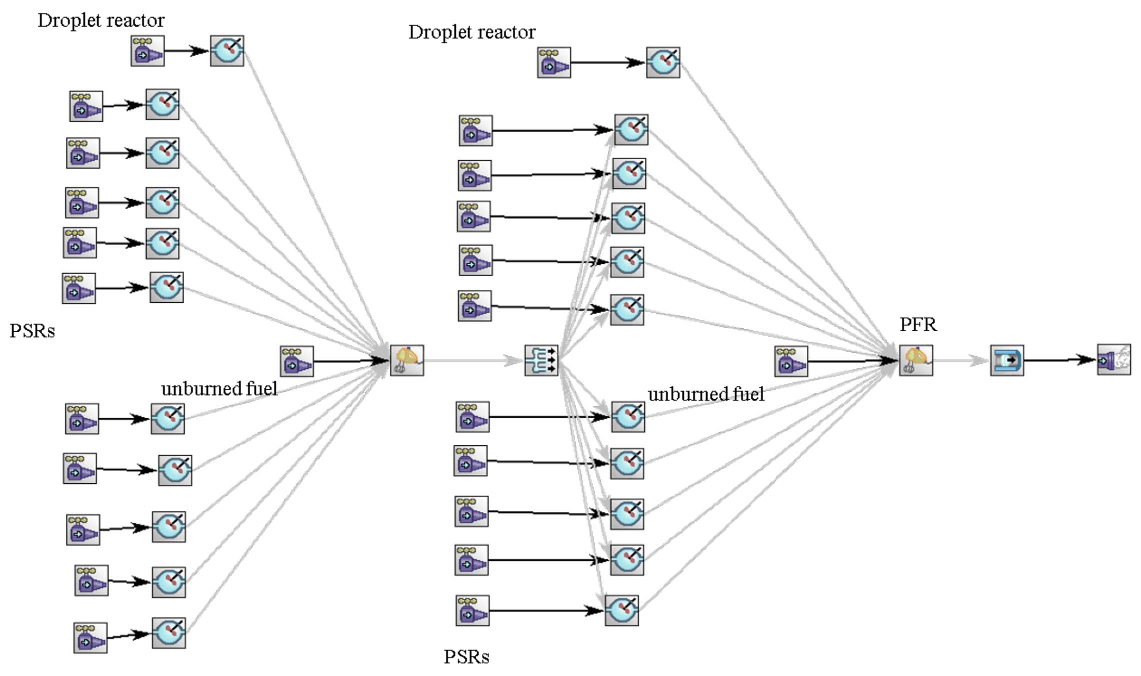


Figure 16: Multiple reactor multiple zone model example.

Recent work has gone from using correlations to calibrating these reaction networks to hit targets based on test data. Here, the ten PSRs are used to simulate the distribution of equivalence ratio within the combustion zone, while one last reactor per zone is used to simulate droplet burning. Since droplets burn at $\phi = 1$, this reactor always burns at stoichiometric conditions while the vaporous fuel is distributed according to input parameters, which are computed before the model executes. The fuel/air ratio (FAR) is divided according to some mathematical function which utilizes the mean equivalence ratio of the reaction zone along with some measure of the spread of equivalence ratio across the reactors. The ϕ distribution and mixedness are both used to compute the amount of fuel and oxidizer in each reactor, and it has been stated that approximately ten to thirteen parallel reactors are adequate when trying to capture NOx [23]. Droplet size can also be simulated with the addition of a fuel evaporation model and the use of the reactor which is reserved for a stoichiometric equivalence ratio. This type of model can be run very quickly and utilizes quite detailed kinetics mechanisms. Although the reaction mechanisms can be the same between the SRMZ and MRMZ models, the extra reactors allow one to capture the physics of the problem quite accurately and very quickly.

In the reactor network shown in Figure 16, the GRI 3.0 mechanism was used to solve for chemical species [35]. This consists of 53 species and a 325 step mechanism. This is a very large equation set which is not feasible if using RFCFD, but here the equations are coupled algebraic equations (due to the steady-state nature of the ODEs), and can be solved very quickly as compared to the coupled PDEs of the reacting flow equations. For the 23 reactor network shown in Figure 16, about forty seconds of runtime is required on a modern personal computer (PC). This approach executes very quickly, and would obviously slow if more reactions were added to the equation set, and speed up if reactions were removed. Importantly, even if these reactions were not run as a steady-state model, the ODEs would still be faster to solve than the coupled PDEs. These reaction networks are detailed models, and therefore really the only feasible approach with detailed enough kinetics capable of predicting minor species whose concentrations range from parts per billion to a few parts per million [36].

2.3.1 Deficiencies

Although this approach sounds like an excellent method for predicting emissions, it is not without its drawbacks. It shares many weaknesses with the single reactor multiple zone model mentioned previously, so one must be cautious to utilize it properly and be fully aware of it's deficiencies. In order to exploit these types of tools, a large amount of input data for a specific design (hence, it must already exist) is required. For instance, one must input all of the properties of the combustor at the locations of each reactor before the model can execute, and the results of the CKN will be *heavily* reliant on these. For instance, the user must input the correct temperature, pressure, fuel flow, air flow, etc. into each reactor before the model can execute, which means one must have a method of determining these properties for a given combustor without testing, as the user is required to compute all of these properties offline and use them as inputs to the CKN. This is a major drawback, since it takes away from the CKN being generic enough to size a combustor and predict emissions through a known design techniques such as design of experiments (DOE). Cardinally, this type of model also requires a calibration step before use, which limits the use of this type of model to existing combustors.

Since this model is one-dimensional (with two-dimensional type corrections in the form of how well mixed are the fuel and oxidizer, etc.), the MRMZ model is unable to truly model two- or three-dimensional effects from first principles. Therefore, numerous "calibration factors" are used within this type of model in order to properly simulate the physics for a given combustor. This is an *absolutely* necessary step because no aerodynamic information is present within the CKN. As an example, the mixedness is typically taken from a curve of experimental data (an empirical model) with a calibration factor as a multiplier, since the unmixedness data was not taken from the specific combustor in question. However, generic trends are available to which one can calibrate, allowing the simulation of the actual combustor [37]. One must therefore know at least one value of a characteristic of the combustor in order to get these calibration factors correct. One popular choice is the EINOx value, however, this is unknown for a future combustor which has not been designed, built, or tested yet! Furthermore, it is very typical to have more calibration factors than data, so an infinite amount of solutions can be found by tweaking calibration factors.

For example, consider calibrating a model of this type to the LTO NOx emissions data for an existing combustor. LTO NOx comprises four data points (TO, CO, AP, and ID). The model will typically consist of numerous calibration factors (k-factors) which may include, but are not limited to

- Evaporation k-factor (to get droplet size). Evaporation may even occur in two zones if fuel is injected in two zones.
- Residence time / volume k-factor in each zone (typically at least 2 zones in combustor)
- k-factor on mixedness. Again, this could occur in two zones if fresh fuel is mixed with old products in an additional zone downstream.
- k-factor on equivalence ratio in each zone (typically at least 2 zones in a given combustor
- etc.

Above, six calibration factors are listed with four EINOx targets, which basically represent the fact that these values should be close, but are not exactly correct since much of the actual physics within the burner being neglected. Additionally, the calibration factors may vary with power setting, making a total of twenty-four unknown factors (in this example). With so many unknowns and only four known values for EINOx (typically, unless on has more information about how the combustor behaves, which is impossible since this concept will not exist), this system is under-constrained, and an infinite number of solutions are possible. One must use engineering judgment to determine which calibration factors are the "correct" ones, and which are incorrect. However, given some ranges on each of the parameters, there is still an infinite amount of solutions!

Scoping back to the interest of this work, this approach is not capable of predicting emissions for a new design. The MRMZ model can be an accurate approach once the model is calibrated, so it is actually more useful for technology forecasting and uncertainty quantification rather than design, meaning that once a combustor model is calibrated, it may be useful for predicting the behavior of the combustor when operating conditions are changed. Although there are multiple reactors per zone, they are only utilized for the sake of simulating distributions and do not necessarily represent spatial locations within the combustion chamber. The conclusion is therefore that the MRMZ approach is useful to study how a given combustor behaves, but is not useful for design.

2.4 Reacting Flow CFD

As computational speed increases and it becomes less expensive to run large scale simulations, higher fidelity tools have come into play earlier on in the design process. A plethora of numerical techniques have been developed for solving non-reacting flow equations [38] and reacting flow equations [39]. These equations are employed within numerous commercial, industrial, and open source CFD codes, and are well-established and used by many (Figure 17). Work has even been done to link internal and external aerodynamics for a complete combustor CFD simulation [40].

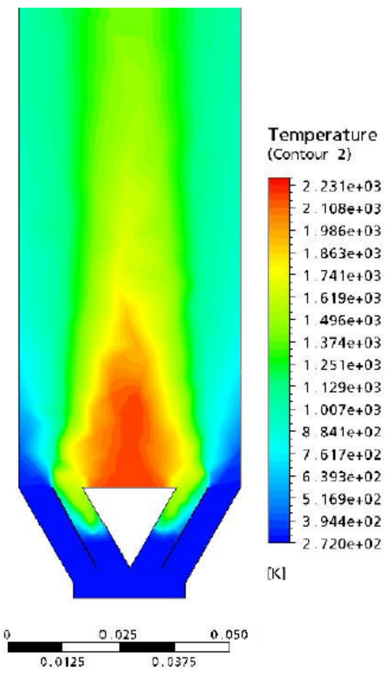


Figure 17: Temperature contour of a bluff body stabilized laminar premixed flame from a CFD simulation (taken from [41]).

In many ways, RFCFD is the highest fidelity and typically most accurate simulation method, with more accurate information only available by testing. This technique utilizes first principles and progressive methods for numerically solving PDEs, in addition to fuel spray models, droplet models, turbulence models, etc. This, in addition to direct computation of species rates, are the main reasons why RFCFD gives the most accurate representation of NOx formation. Specifically, all of the thermodynamic and flow information is coupled with species formation and destruction rates and solved *simultaneously*. According to the Arrhenius law, the reaction rate constant of a chemical reaction has an exponential dependence on temperature [2, 42, 43]

$$k = Ae^{-E_a/RT}, \tag{14}$$

where E_a is the activation energy, R is the gas constant, T is the absolute temperature, and A is the pre-exponential factor. Its temperature dependence is shown separately as

$$A = A' \, T^b. \tag{15}$$

where A' and b are empirical constants. This equation and its form come from collision theory.

In reality, a considerable amount of physics goes into the derivation of the reaction rate equations, and the full form of these equations are not shown here [2, 42]. Each species has its own reaction rate which is directly proportional to not only its concentration and other species' concentrations, but also the temperature. In turn temperature depends on the species concentrations. Since this is a coupled system, it must be solved iteratively. To make matters worse, temperature is linked to other thermodynamic properties, for example a change in pressure will change the temperature (or density, etc), and vice-versa. Therefore, any change in the flow field may change the thermodynamic state and must be solved simultaneously for species at each location within the flow field. Although it involves a large amount of computational work, reacting flow CFD gives a method for one to *directly* tie the chemical reactions to the aerodynamics, so that all of the aerodynamic, thermodynamic, and chemical properties are solved together. It is worth mentioning that ample approximation methods exist within reacting RFCFD, as well. However, many of these methods are oriented at resolving reacting flow at small scales of motion, such as the flamelet approximation [44, 45], and are not necessarily beneficial from the standpoint of emissions approximations.

RFCFD simulations is the approach that one would like to use for all predictions, however it is extremely costly. Species creation and destruction, in addition to the normal flow equations (conservation of mass, energy, momentum, and an equation of state), are all satisfied concurrently [46]. The addition of these reactions into the flow equations introduces additional difficulties when developing a stable, accurate, and efficient solver [31]. Studies have reported that CPU time increases with the number of chemical species to as much as the sixth power [23]. This CPU time increase will be *extremely* constraining given the fact that so many species are involved in Jet-A/air combustion (Section 1.3.1). To avoid this problem, reduced reaction mechanisms are utilized. For rich burn technologies, the thermal NOx mechanism is consistently most prominent and other reaction schemes may be ignored. This is not necessarily the case for new combustors which employ lean burn technology, meaning that more species and more sophisticated mechanisms will need to be taken into account during simulation. Doing so will increase the runtime for a single RFCFD run to a very infeasible amount of time!

Notwithstanding, reacting flow CFD is a very useful and powerful tool. The need for reacting flow CFD will not be eliminated anytime in the near future, since rig tests and hardware are still many times more expensive than monetary costs associated with numerical computing. Since CFD sits as an intermediate step between sizing/preliminary analysis and rig testing, removing it would be detrimental for verification and validation of the initial sizing step. Additionally, numerical algorithms for partial differential equations is a popular field, and work is always being done in an effort to decrease the computational burden associated with running these high fidelity models. One of the most popular methods is just using a reduced order mechanism, since this approximates the heat release and flow field quite well- it just does not give an accurate representation of NOx emissions. Using these approaches many have developed tools which will run RFCFD in an automated fashion [47, 48]. This approach was young a decade ago, and a three-dimensional RFCFD simulations of a fuel injector coupled with the flame tube analysis was quite costly, taking approximately fifteen hours on a single processor for *just* a *single* CFD run (no automated meshing or preliminary tools were run, either) [47]. It also still involved a "man-in-the-loop" approach, since somebody had to drive the low fidelity design which was used as input into this high precision tool.

State of the art approaches now allow one to efficiently automate various stages of the design process via RFCFD. This includes geometry generation, modification, identification, aero-thermal network generation, meshing, CFD preparation, and running automatically (no user input required). It can be driven through an optimizer, effectively allowing one to run DOEs so that RFCFD can be used for autonomous design [48]. As with non-automated processes however, it does not include a reaction mechanism capable of capturing all of the physics which may be necessary in a lean burn combustor (or some other future concept which may be essential to decrease NOx emissions to new CAEP limits). Therefore, this approach is a giant leap towards design automation, but does not cover the preliminary sizing process or NOx emissions fast enough and well enough to be used exclusively in the design process.

2.5 Segregated CFD

As RFCFD is computationally expensive and is unable to support large sets of chemical species due to runtime constraints [36, 43], many utilize segregated (or hybrid) CFD approaches since they allow one to obtain a combination of desired information [49, 34, 50, 36, 51]. The process involves initially solving for aerodynamic and thermodynamic information for a given input geometry by running low order RFCFD. It then utilizes this information to create a chemical reaction network similar to the single reactor multiple zone and multiple reactor multiple zone methods (Sections 2.2 and 2.3). Doing so allows one to ascertain more accurate emissions predictions than could be obtained by the small chemical mechanisms used during RFCFD, utilized solely to keep computational costs to a minimum. Typical RFCFD mechanisms are only between one and eight steps, but reaction mechanisms with many more steps can be utilized in a CKN, up to and including steps with hundreds or even thousands of reactions [35, 51, 36, 52].

The hybrid CFD approach has been run both manually and automatically. It is run manually [49, 34] by user inspection of the converged reacting flow field followed by user creation an equivalent reactor network (ERN) of that given combustor, and automatically [50, 51] by using an algorithm to do this for you. Manually, this involves finding the points where recirculation occurs, looking at the temperature distribution, fuel/air ratio, etc. [49], as in Figure 18. Here, two recirculated product streams are found (dome and main recirculation regions), and the CKN is constructed to simulate flameholding by forcing products to flow between the same reactors. The automatic approach involves a previously developed algorithm doing this for the user (Figure 19).

The manual approach has been very successful as an analysis tool to study emissions in combustors which already exist [49]. However, his approach is not a great fit when considerating an initial concept due to the long required runtimes and the need

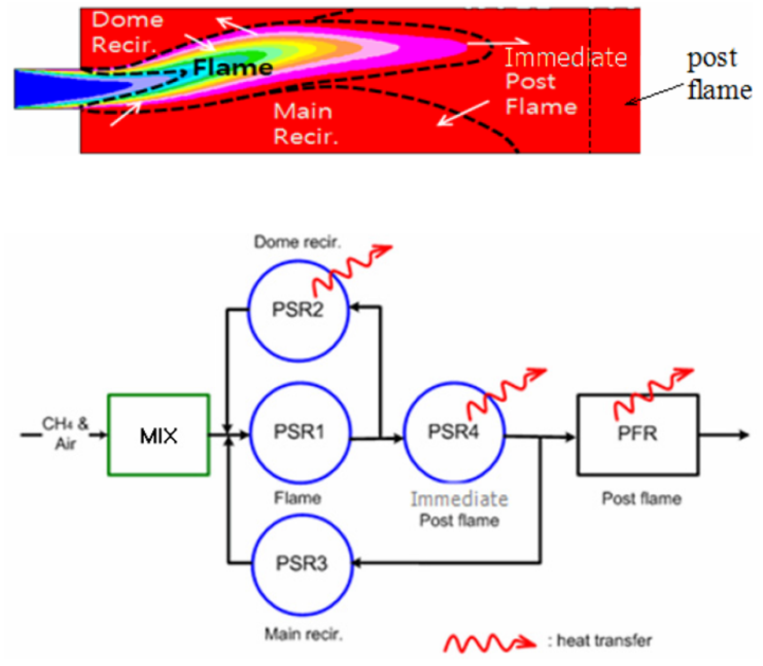


Figure 18: Chemical reaction network manually created from converged reacting flow CFD solution (taken from [49]).

for the user to analyze a flow field and create the ERN from it. With design being the goal in this work, emissions information must be deduced without a "man in the loop" approach. From the automated perspective, this is accomplished via the development of numerous algorithms which execute after a converged solution has been found from RFCFD. Nonetheless, it still executes too slowly to be used as a preliminary sizing techniques, as reacting flow CFD must be done before the algorithm is even run.

Albeit, there are many interesting concepts introduced in the hybrid CFD approach. Firstly, the use of filtering algorithms have been quite successful to determine where reactors should be placed, in addition to determining local Damkohler numbers within each of the cells for analyzing flame stability [51]. Notably, temperature information has been used directly from the reactor instead of the flow field, since that flow field temperature may be "less correct" than that from the reactor. With only an eight step mechanism, the reacting flow CFD solution does not take into account the

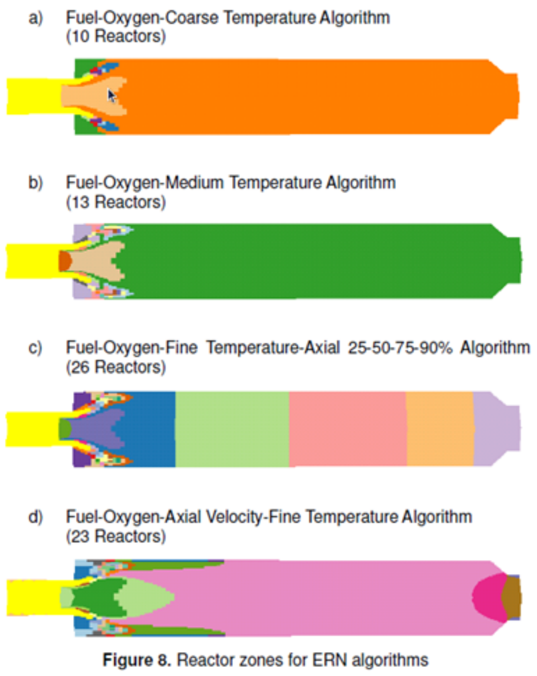

a) Fuel-Oxygen-Coarse Temperature Algorithm (10 Reactors)
b) Fuel-Oxygen-Medium Temperature Algorithm (13 Reactors)
c) Fuel-Oxygen-Fine Temperature-Axial 25-50-75-90% Algorithm (26 Reactors)
d) Fuel-Oxygen-Axial Velocity-Fine Temperature Algorithm (23 Reactors)
Figure 8. Reactor zones for ERN algorithms

Figure 19: CKNs automatically created from algorithms. Each color represents a different reaction zone (taken from [51]).

heat which will be released or absorbed by the various species present in the reactor network, but not present in the CFD solution. It has been hypothesized that minor species only marginally affect the main combustion process and consequently do not influence the overall temperature and flow field [36, 43], however no studies have been performed to counter this argument, nor has a sensitivity study been done on minor species perturbations on the flow field. Contrarily, viscosity is not present in the chemical network, while it is utilized and ever present in flow solutions. Therefore, the best approach would be to compute temperature based on the flow field (which includes viscosity if it is deemed important), while *many* reactors are discretized accordingly so that all effects from species can be taken into account. Disparately, one can leave the amount of reactors the same, but iterate between the CFD solution and reactors until the temperature converges.

Though it is possible to achieve recirculation within SRMZ or MRMZ models, it is much easier to correctly implement in a hybrid CFD approach given the extra information which comes from the converged CFD solution. For instance, within a SRMZ or MRMZ model, one must guess how recirculation occurs within the combustor, since there are no geometry effects within the model. With the hybrid CFD model, one is given much more information about the combustor, making simultaneous aerothermodynamic determinations and accurate emissions predictions much more viable and more accurate than any previous approach. It is the most modern approach since it utilizes geometry data for the flow field to obtain important knowledge like thermodynamic properties, flow characteristics, and more precise kinetics.

CHAPTER III

RESEARCH APPROACH AND TECHNIQUE DEVELOPMENT

3.1 Shortcomings in Current Methods

There are many reasons why the hybrid CFD approach is not necessarily an eminent method for the preliminary sizing of gas turbine combustors. Although temperature differences may not greatly effect sizing and computational speed will very likely be accepted over accuracy (to an extent), it is well accepted that iteration between the flow field and kinetics is unnecessary in certain cases. Arguments even exist against these iterations in later design stages[1] [43]. The second (and most prominent) reason why this approach is not unquestionably optimal is due to the required computational effort and runtime which is necessary for such a method. For sizing, one would like to have as many answers in as short a time as possible, not waiting for converged RFCFD solutions. For example, one seeks the aerothermal characteristics of a given design and whether a NOx target can be attained within a matter of minutes or hours, and not days or months. An approach which can be automated to run DOEs in a design process is desired, so that rapid trade-offs and assessments can be made on given designs as opposed to designs which already exist and are to be evaluated.

3.2 Goals and Approach

3.2.1 Goals

This work aims to size advanced (future) concepts where correlation data do not exist. As Figure 12 in Section 2 depicts, the current sizing process for a new concept has

[1]No experimental proof has been given.

many iterations between initial sizing and testing. The goal of this work is to develop a technique which reduces the number of these iterations. This work targets the development of a method which allows one to rapidly and more accurately (i.e., more rapidly than reacting flow CFD, and with accuracy comparable to other methods which may require calibration) size combustors and evalute their NOx emissions. Many other types of emissions exist (including solid particulates), however the scope of this work encompasses NOx emissions specifically. The technique should predict NOx emissions, burner characteristics, and properties on the same order of accuracy as a correlation / CFD could (assuming the correlation existed for the given design), in order to accomplish a reduction in the number of iterations during design and testing, as shown in Figure 20. A summary of the attributes which should be contained within this new method are listed below.

- Must take into account the necessary mechanisms for accurate predictions of NOx formation.
- Must have about the same accuracy as sizing correlations (assuming the data were available for the combustor in question).
- Must be at least as accurate as NOx correlations (assuming the data were available for the combustor in question).
- Must execute more expeditiously than current RFCFD simulations in order to be used to "quickly" run cases within DOEs.

A plethora of techniques (outlined previously) are available for preliminary sizing, detailed design, analysis, and emissions predictions. Hybrid CFD approaches introduce much more accuracy in chemistry, yet they add computational time. Many of the aforementioned approaches utilize a CKN to get more accurate emissions estimates, and these reaction networks will play a key role in the development of this new sizing methodology.

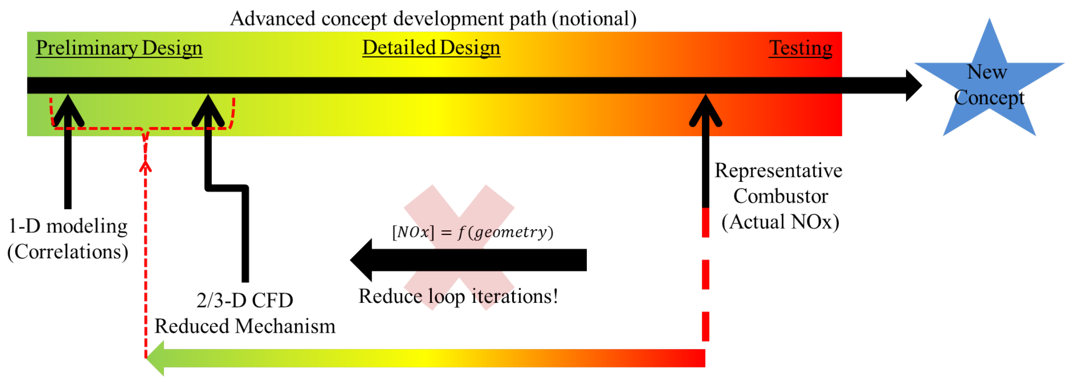

Advanced concept development path (notional)
Preliminary Design
Detailed Design
Testing
New Concept
1-D modeling (Correlations)
2/3-D CFD Reduced Mechanism
[NOx] = f(geometry)
Reduce loop iterations!
Representative Combustor (Actual NOx)

Figure 20: The goal of this work is to reduce the amount of iterations necessary when developing a new, advanced concept combustor. More accurate information (representative of data which would be obtained in the testing phase) will be used during sizing and CFD tests. This will give the designer a "head start" on how the combustor behaves, eliminating the need for so many tests.

Two key features must be present in this new preliminary sizing technique: the flow physics and chemistry must both be captured with sufficient accuracy. Therefore, this new technique involves approximating the flow field within the combustor and using a CKN to obtain accurate emissions values. Contrary to the segregated CFD approaches, runtime will be cut by running non-reacting CFD, since the coupled reactions add considerable runtime (before the filtering algorithms are even executed). Additionally, the method should not present inconsistencies between the aerothermodynamics and chemistry, as occurs in the segregated approaches via CFD and CKN temperature discrepancies. However, approximations should be made whenever possible in order to reduce the computational time necessary per simulation. The overall premise is to take speed over accuracy, but only to the extent that the precision of the method must not be much less than the accuracy of the correlations which would be used if they were available. For example, Figure 21 shows notional NOx emissions prediction accuracy at numerous phases of the design process.

All the way to the right in Figure 21, the EINOx number X is given as the true number, i.e. the EINOx value found when doing NOx certification testing. Before full

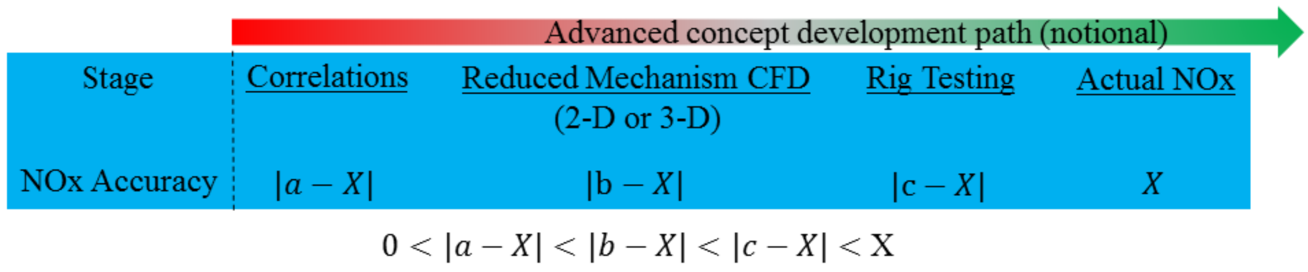

Advanced concept development path (notional)
Stage
Correlations
Reduced Mechanism CFD (2-D or 3-D)
Rig Testing
Actual NOx
NOx Accuracy
|a − X|
|b − X|
|c − X|
X
0 < |a − X| < |b − X| < |c − X| < X

Figure 21: Accuracy during development phases of a combustor. As more testing is completed, the EINOx prediction gets closer to the actual value (X) at the end of the design phase.

engine or combustor tests are performed, rig tests are typically executed during the design process in order to understand more information about how the combustor behaves. However, these tests (especially flame tube and arc-sector tests) will not be as accurate as the complete and final tests. Therefore, the number X here is multiplied by c, which is some percent under or over the correct NOx value. It is expected that the NOx predictions be worse the earlier it is in the design process, since knowledge about the actual design has not yet been gained. Therefore, one expects the numbers a, b, and c to be in increasing order.

Although exact data pertaining to how closely preliminary sizing estimates resemble actual NOx values is not published due to it's proprietary nature, an approximation is made based on research and information provided [53]. The factor a in Figure 21 is taken to be $\pm 10\%$ of the value X. However, fidelity may be sacrificed if it means more DOE runs can be done per unit time, since this is only the sizing phase. However, it is essential to note that this ten percent accuracy pertains to combustor architectures for which correlations have been extensively obtained and tested.

3.2.2 Approach and Methodology

In the absence of any starting information, the approach is developed from first principles. Taking everything into account, the full reacting flow Navier-Stokes equations are

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho \vec{u}) = 0, \tag{16}$$

which is the conservation of mass equation,

$$\frac{\partial}{\partial t}(\rho \vec{u}) + \nabla \cdot (\rho \vec{u}\vec{u}^T) = -\nabla p + \nabla \cdot \left(\mu\Big(\nabla \vec{u} + (\nabla \vec{u})^T\Big) - \frac{2}{3}\mu(\nabla \cdot \vec{u})\underline{I}\right) + \vec{F}, \tag{17}$$

which is the conservation of momentum equation [54],

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \vec{u} h) = -\nabla \cdot \vec{q} + \dot{S}_h \tag{18}$$

which is the conservation of energy equation, and

$$\frac{\partial}{\partial t}(\rho Y_k) + \nabla \cdot (\rho \vec{u} Y_k) = -\nabla \cdot \vec{J}_i + \dot{S}_i \qquad \forall k = 1, 2, \ldots N, \tag{19}$$

which are the species conservation equations [55]. Here, the normal Navier-Stokes equations were given without explanation or derivation. $\dot{S}_h$ is the net source term for viscous dissipation, other work, and heat interactions, and $\dot{S}_i$ is the production rate of the i^{th} species due to homogeneous chemical reactions [55].

By far, the most time intensive aspect is solving Equations 19, of which there are $N-1$ (number of species minus one), and coupling them to the other NS equations. These reactions increase the CPU time by orders of magnitude [23], and cannot be ignored due to the influence that heat release has on density, the velocity field, etc. In order to speed up the process of solving these equations, a method to "indirectly" couple the species creation with the aerodynamics via a CKN is chosen, as shown in Figure 22.

$$\frac{\partial}{\partial t}\rho + \nabla\cdot(\rho\,\vec{u}) = 0$$

$$\frac{\partial}{\partial t}(\rho\,\vec{u}) + \nabla\cdot(\rho\,\vec{u}\,\vec{u}^T) = -\nabla p + \nabla\cdot\left(\mu\,(\nabla\vec{u} + (\nabla\vec{u}^T)) - \frac{2}{3}\mu\,(\nabla\cdot\vec{u})\underline{I}\right) + \vec{F}$$

$$\frac{\partial}{\partial t}(\rho\,h) + \nabla\cdot(\rho\,\vec{u}\,h) = -\nabla\cdot\vec{q} + \boxed{\dot{S}_h} \longleftarrow$$

$$\cancel{\frac{\partial}{\partial t}(\rho\,Y_k) + \nabla\cdot(\rho\,\vec{u}\,Y_k) = -\nabla\cdot J_i + \dot{S}_i} \dashrightarrow \textit{CKN}$$

Figure 22: Mathematical description of *indirectly* coupling species creation and heat release with aerodynamics. The heat release comes from the CKN, and is inserted as a source term into the energy equation. This introduces the correct heat release into the flow field while still coupling to the aerodynamics.

Non-reacting CFD will distinguish the aerodynamic properties within the combustor, but will not give any information about the kinetics. The CKN will in turn resolve all of the information about the kinetics, but none about the aerodynamics. Capturing the combined approach involves an interaction between the two models. An iterative procedure results, and is run until the flow field is fully resolved and converged, as shown in Figure 23.

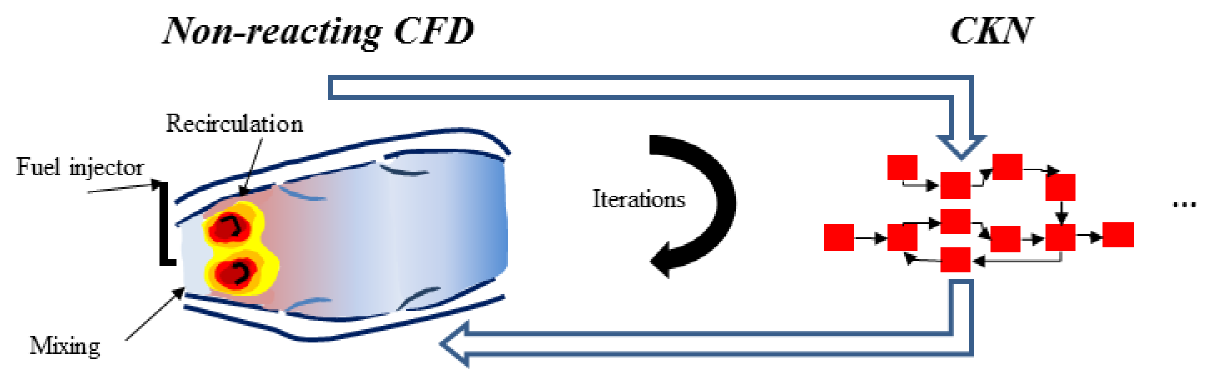


Figure 23: Iterations between the flow field and the CKN take place until convergence is achieved. At each iteration, a converged non-reacting flow field with source terms is found, and a CKN is run based on this information. At the end, the CKN is be run to give the actual NOx values.

The temperature information from heat release goes into the energy equation as a source term, only the value of the source term is found in the CKN, since this will give the most accurate heat release based on its ability to take many species into account. The steps shown in Figure 23 are explicitly listed below.

- A converged solution from the non-reacting flow field is obtained.
- Said solution is examined and filtered in order to resolve where reactions take place.
- This information is used to construct a CKN, which gives actual temperature and heat output.
- This heat release information is inserted across the nonreacting flow field.
- This new flow field is run until convergence is achieved.
- After the flow field converges with the correct heat release, the proper NOx is obtained.

This motivates the first research question of the work.

Research Question I

Can an algorithm which applies filters to a non-reacting CFD solution and utilizes chemical rate information from a CKN be developed in order to bring kinetics information into the flow field? Will this method be fast enough, accurate enough, and robust enough to use as a preliminary sizing tool for advanced concepts, or when correlations do not exist?

As a side note, using a CKN allows one to run large chemical mechanisms and capture minor species, which is intriguing for more reasons than just emissions. Multiple software packages (NPSS, [56] as an example) use chemical reaction functions which don't take all of the true species into account. This may only change the temperature by a small percentage, however a few tens of degrees may be the difference between achieving and not achieving objectives such as an SFC target. Furthermore, some PD software use equilibrium burn functions in addition to not resolving all species, making matters even worse.

Considering full non-reacting Navier-Stokes flow is run, this may still be time consuming contingent upon mesh size. Additionally, iterations are required to obtain a converged solution since reactions are present and introduced as species source terms (though the amount of iterations depends on the influence that the temperature has on the flow-field). To get this to run as fast as possible, the fidelity of the computational mesh and CKN are explored. This leads to the second and third research questions.

Research Question II

Which size mesh should be used in the non-reacting flow segment in order to obtain "accurate enough" NOx solutions? How does this compare with the actual RFCFD solution? Does the non-reacting CFD mesh have a large impact on the emissions evaluations?

Research Subquestion II

Which type of filtering algorithm does one employ in order to detect where reactions occur, and which non-reacting CFD properties are used to do so?

Research Question III

How many chemical reactors are necessary to use in this method? How are their locations determined from the non-reacting flow field (related to research sub-question II)? How must the reactors be arranged in order to obtain an "accurate" answer?

The accuracy of a CFD solution increases with the number of cells employed in the computational domain[2]. Since the new approach indirectly couples the kinetics with the aerodynamics, a coarse "kinetics mesh" (KM) that takes more species into account in combination with a coarse CFD mesh may still be more accurate than a coarse RFCFD mesh with a reduced mechanism.

Though all efforts will be made to obtain expeditious, yet accurate solutions, the investigation of specific numerical methods for solving coupled nonlinear PDEs and explicit approximations to flow physics within the combustor are both of minimal concern in this work. These are both popular techniques for abridging runtime, but are separate areas of research. For example, some target the removal of viscosity and diffusion and only solve the Euler equations, since viscous heating will play a minimal role as compared to heat released during the reactions [39] since the flow through a conventional gas turbine burner is so slow. The Euler equations are

$$\frac{\partial}{\partial t}\rho + \nabla \cdot (\rho\vec{u}) = 0, \tag{20}$$

which, again, is the conservation of mass equation,

$$\frac{\partial}{\partial t}(\rho\vec{u}) + \nabla \cdot (\rho\vec{u}\vec{u}^T) = -\nabla p + \vec{F}, \tag{21}$$

[2]This is however not true when the cell sizes are on the order of the machine's precision, in which case the answers are useless.

which are the momentum conservation equations, and

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\vec{u}(E + p)) = -\nabla \cdot \vec{q} + \dot{S} \tag{22}$$

which is the energy conservation equation (with the source term added for the reactions that take place in the CKN).

Additionally, many make constant pressure assumptions [39], since heat is released at a constant pressure during the Brayton cycle. Probably the most popular assumption is the incompressible flow assumption, which is made to tremendously abate the time it takes to run the flow solver [57]. Here, density changes come from the heat release as opposed to the flow field. This (in addition to many advances in GPU computing) have facilitated rapid fluid flow solution acquisition. Specifically, NVidia's PhysX technology allows remarkably dense meshes to be used and stills solve the Navier-Stoke's equations at 30 frames per second [58]. Keeping this in mind, the focus of the work is to develop an innovative method to tie time rate chemistry to flow solutions, regardless of which type flow solver is used.

3.3 Experimentation Plan

Research which explores this indirect coupling does not currently exist, as previous (and similar) approaches (i.e. segregated CFD) still utilize temperature information from "directly coupled" reacting flow solutions [51]. Without performing research, the feasibility of an indirect approach is put into question since it is unclear that this method can even produce a viable and accurate solution. Experiments must therefore be conducted to prove or contradict the validity of the assumptions within and the methodology of the approach. The following hypotheses are stated, with associated research experiments proposed to validate them.

Research Hypotheses I

- An algorithm which applies flow field filters to determine where reactions occur can be developed and used to accurately predict chemical reaction locations. It is possible to iterate between the aerodynamics and CKN to find the correct flow field and predict NOx emissions on the same order of accuracy as preliminary sizing correlations.

- A small chemical mechanism is good enough to obtain an acceptable flow field estimation, but a much larger reaction mechanism will be necessary to determine emissions.

Research Experiment I

An algorithm will be implemented (developed as part of this research), which makes it possible to determine where the reactions within the combustor occur. Certain algorithms found in research may be used as a starting point. This will be done via trial and error to find which properties must be utilized and which filters must be applied to give the best answer as compared to known results or to software.

Research Hypotheses II

- A lower fidelity grid with a detailed CKN will give a more accurate NOx prediction than RFCFD with a low order mechanism.
- The reactions will occur at locations where the fuel and air mix well enough for a reaction to ensue, irrespective of the density or pressure.

Research Experiment II

Grids of varying fidelity will be used, and the results compared to those known or those from validated software. Testing will take place on segments of the algorithm as they are established. Concepts which work will be kept and those which fail to produce good results will not be used.

Research Hypothesis III

The number, arrangement, and type of reactors will be vary based on the problem at hand. Different geometries will require different combinations of reactors, but there is a single algorithm which gives the best results (in terms of obtaining the correct flow field and emissions).

Research Experiment III

Multiple KM geometries will be tested in the form of numerical experiments. The results will be compared with known results or software solutions.

Research Experiment I requires a literature review not only in combustion and numerical computing literature, but also in computer science literature for useful algorithms and lessons learned in subjects such as pattern recognition, filtering algorithms, and software automation (since this method should be one single process consisting of flow detection and model execution). It will also require running reactive flow CFD for comparison against the developed method, in addition to running the process with different levels of reaction mechanism fidelity (as to answer Research Subquestion I). In addition to developing and coding the algorithm, this step also includes writing gridding software and a CFD solver. For all of these reasons, this step requires the most time.

Research Experiment II involves iteration with Experiment I while the CFD mesh fidelity is varied and multiple runs with the developed algorithm are performed. Comparisons of each run will be made in addition to an assessment of the solution fidelity.

Research Experiment III involves iteration with Experiments I and II since the temperature is sent back to the flow field to determine its effect. As with experiment II, it will also be done after the algorithm is complete.

Note that much of the work is coupled, meaning that different techniques tie together and rely on one another. For example, running different sized meshes will be done simultaneously with other parts of the algorithm development, since they each have an effect on one another.

Most aspects of validation for this technique come from tests against similar geometries in other software. Ideally, combustor designs (geometries and properties) from actual hardware which have been tested (and whose correlations exist) are best to use, but these tend to be highly proprietary. Therefore the focus of the work is a computational proof of concept. In theory, the methodology can be shown to work well even by utilizing simple geometries and comparing against known solutions, such as flame tube tests, CFD results, or other software packages. The overall validation plan utilizes information from such software tools, which are much higher fidelity and more costly to run than the proposed method. The proposed sizing technique should show similar results to RFCFD, but with much higher chemical fidelity (better NOx estimates for architectures that require large obfuscated chemical mechanisms, such as lean burn designs). Overall, this work should show that such an indirect coupling technique makes it possible to run extremely large chemical mechanisms and is feasible to use as a starting point (based on computational and temporal constraints) by combustor designers for concepts about which little information is known, besides the design's targets and goals.

CHAPTER IV

ESTABLISHED METHODOLOGY AND COMPUTATIONAL TECHNIQUE

The development of this novel approach involves the use of three different tools. Expressly, a non-reacting CFD code is required to solve for the aerothermodynamics, a chemical kinetics code for the time rate chemical properties, and an algorithm or set of algorithms to couple them together. These coupling algorithms serve many purposes. Preeminently, these algorithms filter the CFD flow field to determine where reactions occur. They place the reactors, act to tie the CFD output to the chemical kinetics input, in addition to situating the chemical information back into the flow field.

The formulation of this overall approach therefore takes place in three major steps. First, a non-reacting CFD code must be chosen or developed. A multitude of CFD codes with many classes of numerical methods exist, however the source code is neither publicly available nor distributed. Typically, any computation and modification of source terms is done internally to the code via RFCFD methods. However, packaged solvers either do not make source terms available to the user, or do not work well if the user tries to modify them at will. Additionally, even if the source terms are modifiable, not having access to the source code means a severe escalation in runtime. Since this work requires experimentation to determine source term values and how they should be input, the computational cost of input/output (I/O) is too expensive to run on each iteration. Doing so involves writing out the CFD state vector, analyzing it using the algorithm developed in this work, create a CKN, run it, compute the source terms, write out those source terms, and then read them back into the CFD solution. Having source code available alleviates the need for reading and writing, which becomes more expensive as the number of CFD cells and KM cells increases (i.e. running higher fidelity models). No general enough CFD code is found to be suitable, so the first part of the work involves the development and validation of a non-reacting CFD solver.

Secondly, a chemical kinetics solver capable of integrating ODEs in accordance with stirred reactor approximations is paramount given the method directly relies on the use of a CKN. Additionally, all properties of the reactors and species within the CKN must be available. This information must be accessible from memory and have the capability to tie directly into codes based on the temporal constraints presented via I/O file methods. The last criterion which must be satisfied for the kinetics solver is the ability to create many different CKNs with many different architectures on the fly. Many chemical kinetics packages already exist, such as CHEMKIN [59] and Cantera [60]. CHEMKIN involves complicated batch scripting or the use of a graphics user interfaces (GUIs) to set up the structure of the network. This makes CHEMKIN unusable, since the network must be structured "on the fly" within the developed algorithm. This capability is however present within Cantera, which is used exclusively as the chemical kinetics solver.

Lastly (and most importantly), algorithms are researched and developed to find a method for exchanging information between the non-reacting CFD flow field and the CKN, which is in effect the method for indirect coupling. This involves numerous algorithms which work together to accomplish the task, and is the crux of the research.

4.1 Non-Reacting CFD Code

4.1.1 Computational Mesh

The first part of running any CFD code is the creation of a computational mesh. Many programs which focus on numerical grid generation currently exist [61, 62], however simple gridding software was written and optimized with the specific goals of this method in mind. Namely, the gridding functions are written to directly link with the rest of the code so that the object-oriented ”Grid” class is openly available to all other classes and routines. This allows a mesh to be generated upon program onset and modified if need be (i.e. an adaptive mesh [63, 64] during execution, if expanding upon the work), or allows an externally developed mesh to be input. This work focuses on simulations in two-dimensions so that multi-dimensional flow features are captured (on a per unit length basis) while runtime remains low (compared to three-dimensional simulations). Additionally, structured grids are used for ease of implementation and time savings. Using a set number of points in each direction, the solver is more easily implemented in a structured configuration, has the ability to compute fluxes in a vectorized fashion.

The best possible grid which can be generated in any case is a completely Cartesian grid which is evenly spaced, since each of the cells would be the absolute same shape and sit at 90° angles with respect to each other. This (and many other) CFD codes are computed using finite volume methods, which means that conservation laws are solved within each cell. The shape should therefore not make too big a difference as long as the cell shape is captured correctly using the grid Jacobian. However, finite difference methods are used throughout the code in order to compute certain variables, such as those used to impose boundary conditions. It is rare to find a computational problem which can utilize an evenly spaced Cartesian grid, and this seldom happens for gas turbine combustion systems. With orthogonality and cell skew being important, special care is taken when devising these grids.

The grid generation routines within this code utilize special functionality before placing points. An even spacing is used in the vertical direction, while characteristics pertaining to the upper and lower bounds of the domain are computed before the mesh is generated. Specifically, the number of grid points in the vertical direction along with the domain height are used to divide the domain into horizontal segments. Using the cell height and number of cells in the vertical direction, the code computes the number of grid points in the horizontal direction which must be used in order to obtain perfectly square cells. This information is then used to place grid points along the horizontal direction in a fashion which gives evenly spaced cells (and cells which are as square as possible). The algorithm for doing this is shown below.

```
/* User sets corner points */

/* Below sets upper and lower boundaries */
for (i=1; i<numGridPts_x-1; i++) {
  for (j=1; j<numGridPts_y-1; j++) {
    upper     =  f(x)
    lower     = -f(x) + min(-f(x))
    dx_{i,j} = (upper - lower) * numGridPts_y
  }
}
y_j_upper =  f(x)
y_j_lower = -f(x) + min(-f(x))
x_i       = x_{i-1} + dx_{i,j} //All x-points

/* Below sets all interior points */
for (i=1; i<numGridPts_x-1; i++) {
  for (j=1; j<numGridPts_y-1; j++) {
    x_{i,j} = x{i-1,j} + dx{i,j};
    y_{i,j} = y{i,j-1} + ( y{i,numGridPts_y-1} - y{i,0} )
                         / numGridPts_y;
  }
}
```

A duct is used as an example, since aircraft combustors resemble ducts with convergent and divergent sections. As one moves through the duct, the distance between the walls changes. If the grid points are evenly spaced in each direction, there will be few evenly sized cells. If one takes the x-direction spacing as a function of the distance between the walls in the y-direction, a much more even spacing is obtained, which is targeted here. After the boundary points are set, the distance between each interior point is found in each direction by utilizing the distance between the boundary and the number of points in each direction.

In general, the geometry of the problem sometimes makes it difficult or inconvenient to use algebraic grids since the presence of skew may increase numerical error and degrade convergence. Grid clustering may be useful in areas where skew is present and an accurate solution is needed (for example, if one is worried about quenching near walls or heat transfer in areas close to reactions, etc.) However, clustering must be imposed and used carefully since stretching functions themselves do not typically impose orthogonality constraints. Instead, one may rely on mathematical methods for grid generation, namely an elliptic grid solver [65]. Here, the simple algebraic grid above is used as the starting point and a conformal mapping is used to map the x, y coordinates to ξ, η coordinates [66]. In order for the mapping to be analytic, the Cauchy-Riemann relations must hold. When differentiating these relations, it leads to Laplaces equation.

$$\nabla^2\xi = 0 \tag{23}$$

$$\nabla^2\eta = 0. \tag{24}$$

The Grid class contains functionality for solving these PDEs to determine the grid point locations. The x and y boundary points are specified as elliptic boundary conditions, and Laplace's equation is solved. Iterations take place on the interior of the grid (while keeping the boundary points static), until the residual is minimized, making the cells as square as possible. Equations 23 and 24 are discretized using finite differences and stored. An equation for each ξ gridline is formed and the solution to these equations gives the x and y locations of the grid point on the next iteration. Interestingly, the form in which these equations are stored can make a big difference.

Here, these equations form a tridiagonal system, so the elements are stored in one sub-diagonal, one main diagonal, and one super-diagonal array. The algorithm takes this into account so that the solution time is as fast as possible, as shown in Figure 24.

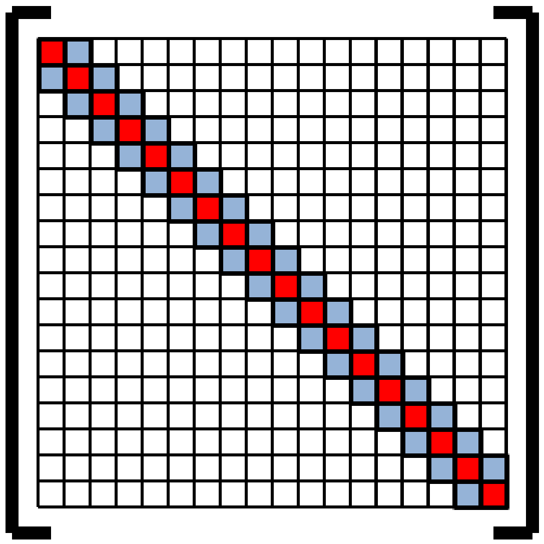


Figure 24: A tridiagonal system can be efficiently stored on a computer by only storing the colored matrix entries (as opposed to storing the zeroes, as well). The darker the color, the higher the magnitude of the entry in the matrix. A tridiagonal matrix can be solved with order n operations (O(n)) using the Thomas algorithm.

A line Gauss-Seidel method is used to solve for all of the new points because a linear system is solved along each ξ gridline such that the η direction stays constant. The governing equations for grid relaxation using the Laplace method are shown in the Appendix, Equations 55 through 59.

After each iteration, the new grid is compared to the old grid and the root mean square difference between the two is determined. If the error is low enough to meet the convergence criteria, the iterations stop and the final grid is found. The convergence criteria is given by

$$\epsilon_{RMS} = \sqrt{\frac{1}{N}\sum_{i=2}^{i_{max}-1}\sum_{j=2}^{j_{max}-1}[(X_{i,j}^{n+1} - X_{i,j}^{n})^2 + (Y_{i,j}^{n+1} - Y_{i,j}^{n})^2]} \tag{25}$$

where ϵ_{RMS} can be specified to whatever value is desired.

Poisson's equation can also be modified to employ control functions by using Equations 26 and 27. The difference between this and Laplaces equation is that it allows one to specify control terms, which allows extra regulation over the interior point distribution (though it does not create a conformal mapping). This is not necessarily a bad thing; although it necessitates the user to be more cautious, it may help eliminate skew.

$$\nabla^2\xi = P(\xi, \eta) \tag{26}$$

$$\nabla^2\eta = Q(\xi, \eta) \tag{27}$$

Utilizing the Middlecoff-Thomas method for the control terms and finite difference formulations for derivatives, the actual equations are obtained (Equations 60 and 61 in the Appendix) and solved

Note that the Cauchy-Riemann relations are no longer enforced here due to the addition of the forcing function. In reality, the algorithm for creating square cells (in the pseudocode above) helps greatly, and it is not always necessary to solve PDEs. Though it will not hurt, it is a computational burden and the grids which are generated without using these PDE techniques are typically good enough for use.

4.1.2 Non-Reacting CFD Flow Solver

During the decades that computational hydrodynamics / fluid dynamics has been around, a plethora of numerical algorithms have been developed to solve the basic

Riemann problems [67, 66, 68]. The most general equations which describe reacting fluid flow are the Navier-Stokes equations[1], written in Chapter 3 Section 3.2.2, Equations 16 through 19. The solutions of the equations must be found numerically since these equations can not be solve analytically unless severely limiting approximations are made [69, 70]. By writing the state vector as

$$\vec{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix}, \tag{28}$$

these equations can be recast into standard conservation law form, which appears as

$$\frac{\partial}{\partial t}\vec{U} + \frac{\partial}{\partial x}\vec{F} + \frac{\partial}{\partial y}\vec{G} = 0. \tag{29}$$

Here, $\vec{F}$ and $\vec{G}$ are the flux vectors in the x and y directions, respectively. The fluxes denote the amount of fluid moving in and out of each of the cells on the domain (Figure 25).

Since the grid is not Cartesian in general, a transformation is made so that fluxes through a skewed cell (as in Figure 25 (b)) may be treated as fluxes through a Cartesian cell (Figure 25 (a)). To do so, the flow equations are transformed into generalized coordinates [66, 71]. In this coordinate system, the form of the equations are

$$\frac{\partial}{\partial t}(\frac{\vec{U}}{J}) + \frac{\partial}{\partial \xi}\vec{F}' + \frac{\partial}{\partial \eta}\vec{G}' = 0, \tag{30}$$

[1]In the most strict sense, assumptions are made in the derivation of the NS equations, as well. For example, these equations hold under the continuum assumption, a Newtonian fluid is assumed for the form in Chapter 3, etc.

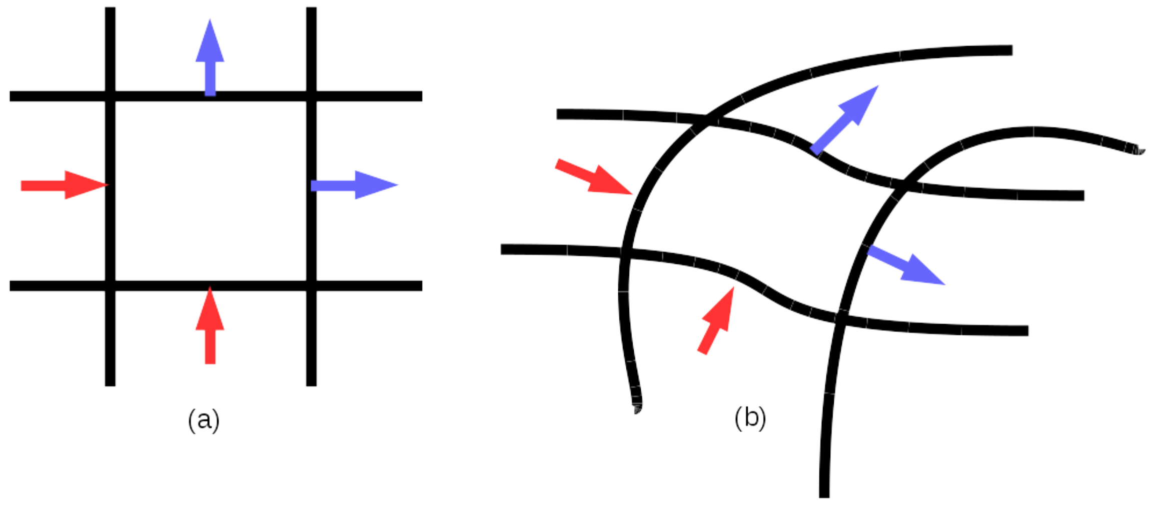


Figure 25: The numerical mesh consists of numerous computational cells. Since the fundamental equations describe conserved quantities, the fluxes describe fluid moving into and out of computational cells, (a) for a Cartesian grid, and (b) for a generalized grid. The red arrows indicate fluid moving into the cell, and blue arrows indicate quantities moving out of a cell.

which has the same form as Equation 29.

All of the solution variables are stored in the state vector, and an explicit time marching scheme is chosen to compute the solution at a later time, given the initial conditions. Though implicit schemes tend to converge faster, an explicit approach is chosen due to ease of implementation. Equation 30 is solved explicitely by evaluating the fluxes at the half point locations and marching forward in time until steady-state convergence is achieved (Equation 31). The update at each time step takes place with the equation

$$\vec{U}^{n+1}_{i,j} = \vec{U}^{n}_{i,j} - \Delta t^n_{i,j} J_{i,j}[(\vec{F}'_{i+\frac{1}{2},j} - \vec{F}'_{i-\frac{1}{2},j}) + (\vec{G}'_{i,j+\frac{1}{2}} - \vec{G}'_{i,j-\frac{1}{2}})]^n, \tag{31}$$

and the time step is recomputed at each iteration. Here, the time step is found by picking a CFL condition and solving for the maximum step which can be taken based on the grid spacing. The time step is found using Equation 32 [72],

$$(\Delta t)^n = \min_{i,j}\left\{ \frac{CFL}{|\tilde{u}| + |\tilde{v}| + c_s\sqrt{\xi^2_{x_{i,j}} + \xi^2_{y_{i,j}} + \eta^2_{x_{i,j}} + \eta^2_{y_{i,j}} + 2|\xi_{x_{i,j}}\eta_{x_{i,j}} + \xi_{x_{i,j}}\eta_{x_{i,j}}|}}\Bigg|^n_{i,j} \right\}, \tag{32}$$

Where, $\tilde{u}$ and $\tilde{v}$ are the contravarient velocities, c_s is the local speed of sound, and the ξ and η terms are the grid metrics [66].

An important part of obtaining a numerical solution is to decide when that solution is close enough to the "correct" answer that you can stop solving and accept the current answer as the "final" answer. This can be done in many different ways. Here, the solution is taken to have converged when the state vector stops changing by a specified amount from one iteration to the next, as shown in Equation 33.

$$\epsilon = \frac{1}{N}\sqrt{\sum_{k=1}^{4}\sum_{i=1}^{i_{max}}\sum_{j=1}^{j_{max}}(\vec{U}^{n+1}_{k,i,j} - \vec{U}^{n}_{k,i,j})^2}. \tag{33}$$

This is the absolute RMS, as opposed to the relative RMS, which normalizes by the first RMS value. The absolute RMS is a much more strict convergence criterion, and is used here.

The fluxes in Equation 31 can be computed any way the user desires, as numerous flux functions exist [38, 39, 66, 73, 74, 75, 76]. Two flux schemes are implemented and tested in this code. First, the Rusanov flux [77, 78, 79] is used, since it is one of the simplest flux implementations and typically works very well for subsonic flows. The form of this flux function is

$$F(\vec{U}_L, \vec{U}_U) = \frac{F(\vec{U}_L) + F(\vec{U}_U)}{2} - c\frac{(\vec{U}_L - \vec{U}_U)}{2}, \tag{34}$$

where the $F(\vec{U}_L) + F(\vec{U}_U)$ functions are the standard fluxes,

$$\vec{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u + p \\ (E_t + p)u \end{bmatrix}, \tag{35}$$

$\vec{U}_L$ and $\vec{U}_U$ are variables from the state vector at the lower and upper part of the cell, and c is the maximum local speed of sound. The Rusanov flux function is very attractive since the fluxes in both dimensions map to one coordinate direction. Though an easily implemented scheme, the nature of the algorithm leads to high gradients in thermodynamic properties (see Section 4.3.2), so another flux method is explored.

Some recent flux functions take advantage of flux vector splitting methods [80, 81], which allow one to construct a stable upwind differencing scheme while capturing both positive and negative eigenvalues [82]. Flux vector splitting methods are capable of capturing rapid changes in properties and are popularly used to capture shocks [83, 84]. AUSM (Advection Upwind Splitting Method) type schemes [85], which are a subset of general flux vector splitting methods, are particularly well equipped to handle these types of problems. The AUSMPW+ (Advection Upwind Splitting Method Pressure Weighted +) scheme [86] is one example of an AUSM algorithm. Originally developed to capture oblique shocks [87], AUSMPW+ can be used to capture rapid changes in flow field properties. In addition it has been adapted for multiple species [86] and multifluid flows [88]. The AUSMPW+ flux is given by

$$\vec{F}_{\frac{1}{2}} = \bar{M}_L^+ c_{\frac{1}{2}} \vec{F}_L + \bar{M}_R^- c_{\frac{1}{2}} \vec{F}_R + P_L^+ P_L + P_R^- P_R. \tag{36}$$

Again, $\vec{F}_{L/R}$ are the standard fluxes (Equation 35) and the Mach numbers and pressure terms are computed based on the AUSMPW+ algorithm. Equation 35 (above) shows the flux in the x-direction. A similar function is used in the y-direction, and both are transformed to generalized coordinates for use in the AUSMPW+ scheme. In this work, multiple fluids are not necessary, since a simple mixture fraction approach is used to capture the advection of the fuel/air mixture through the combustor [7, 89] (See Section 4.3.2). The flux functions are flexible enough to swap in and out, and a change in the numerical flux approach only means instituting a new flux function.

When the simulation starts, the flow through the entire domain is initialized to the inflow conditions. Although the code is general enough to handle supersonic flow (in which case the outflow conditions are specified as smooth outflow), it is not of interest in this work. Since the flow is subsonic, characteristics propagate upstream to modify the inflow boundary conditions throughout the entire simulation so that new inflow properties are computed at each time step. Therefore, the static pressure is imposed at the outlet so that a unique solution is found.

Solid walls are imposed as boundary conditions, also, since combustor cans are enclosed by physical walls on the top and bottom (on a 2-D cross section). Dilution jets may enter through the top and bottom boundaries, as well, but only in user specified locations. These conditions are constants which are specified before runtime. In actuality, there are important effects of the air flow around the outside of the combustor which must be modeled separately, so as to inject dilution air with the correct mass flow, velocities, and turbulence characteristics. The airflow external to the main can is neglected in this work, and the airflow properties coming into the can from dilution jets are simply set buy the user. Additionally, the flow may be inviscid or viscous (depending on if the Euler or NS equations are solved), so the flow may or may not slip along the walls. In the cases when the flow can slip along the walls, the flow is still unable to penetrate the wall so nothing can ”leak” through the boundary (i.e. all of the flow properties must be conserved at the wall). Here, the solid wall boundary condition is imposed by determining the value of the velocity at the nodes just interior to the wall and then setting the boundary velocity to be equal and opposite (a finite difference approach). In effect, this "cancels" any velocity traveling through the wall. Here, this is implemented by determining what the contravarient velocities would be at the wall to cancel out, since generalized coordinates are used (Figure 26).

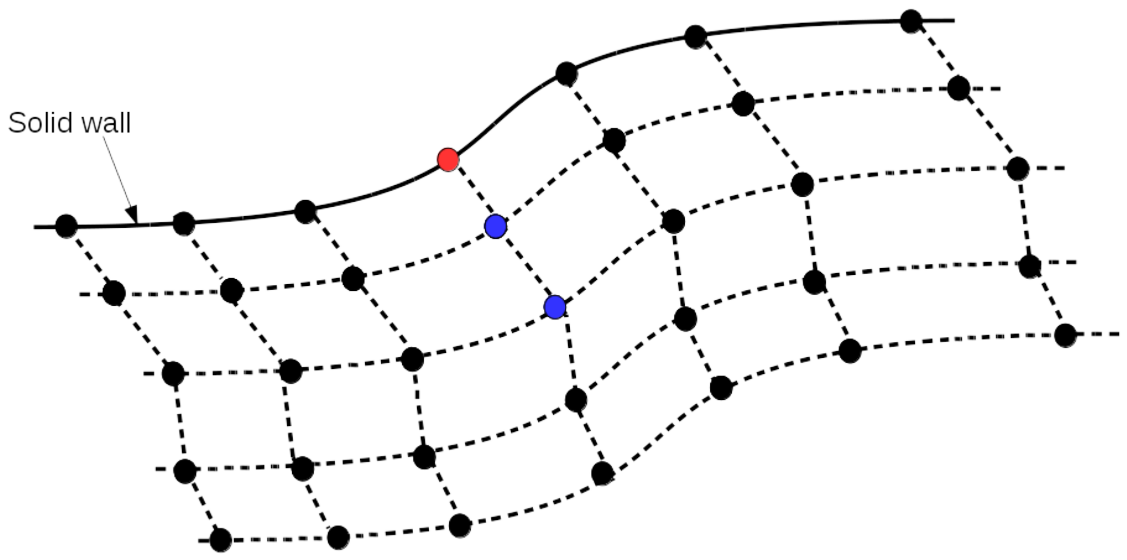


Figure 26: The temperature, density, and velocities at the wall are found by extrapolation of the two most interior points to the wall. The velocities are needed to "cancel" flow at the wall. The red point is the location along the wall where the boundary condition must be found, while the blue points are the interior points to the boundary.

Inviscid fluxes not do take into account much of the right hand side of Equation 17. In order to take viscosity into account, terms must be added to the flux vectors. In general, these extra flux terms are

$$\vec{F} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix}, \tag{37}$$

for the x-direction and similar for the y-direction [38]. These viscosity terms are added to the inviscid fluxes so that the total fluxes in Equation 31 are actually the inviscid fluxes minus the viscous fluxes, $\vec{F} = \vec{F}_{inv} - \vec{F}_{vis}$ (if viscosity is present). The heat transfer terms involve computations containing viscosity, specific heat at constant pressure, thermal conductivity, density, temperature, and free stream (inlet) conditions. These values are found in tables or computed in the flow field by utilizing formulae such as Sutherland's law for the viscosity computation [90].

As hinted above, the implementation method for the boundary conditions may change due to the presence of viscosity (i.e. a slip versus no slip wall, even though it is still a solid object). When viscosity is present, the solid wall boundary conditions are changed to a no slip condition along with either adiabatic boundary conditions (constant heat) and isothermal boundary conditions (constant temperature). Arguments can be made for both adiabatic and isothermal boundary conditions in the case of a combustor. In the case where heat transfer through the liner is not considered, adiabatic boundary conditions are more appropriate. However, if heat is transferred through the liner and interacts with the liner flow, isothermal boundary conditions may be used since the inner and outer liners will stay at about constant temperature during steady-state operation. The velocity at the wall is zero (causing the momentum, the two middle entries to the state vector, to be zero). In order to calculate the pressure, an equation is obtained using the normal momentum equation and a first order approximation during discretization.

The last boundary condition to employ is temperature. For the adiabatic boundary condition, no heat transfer occurs through the walls. Here, a similar expression as that for normal momentum to a wall is constructed. For the isothermal case, the wall temperature is specified and held throughout the entire simulation. Trivially, this condition is given by (for the lower boundary)

$$T^*_{i,1} = T^*_{wall}. \tag{38}$$

Following the computation of the pressure and temperature at the solid, the density is found using[2].

$$\rho^*_{i,1} = \frac{\gamma M_\infty^2 p^*_{i,1}}{T^*_{i,1}}, \tag{39}$$

which allows the computation of the entire state vector to be carried out.

The source terms are determined from the heat release in the chemical kinetics routines (see Sections 4.2.1 and 4.3). In general, the CFD techniques implemented here are very widely known, and the CFD code may be modified to utilize any other techniques one desires. As will be stressed in Section 4.3, the novel technique developed in this work does not rely on a specific CFD solver, which is why a simple explicit method is chosen here.

4.1.2.1 CFD Code Validation

To ensure the non-reacting CFD code operates as intended, a series of validation cases are run and outputs examined. The purpose here is to run different cases under various conditions and compare the results to known (analytic) solutions. If the results match, the user has assurance that the CFD code operates as expected and that further data obtained when running the code during numerical experimentation can be trusted.

The first validation case is a simple converging-diverging nozzle which is ten inches long and five inches wide [91]. The shape of the domain is described using a simple trigonometric function

[2]Note that the values here have been normalized. All values in the CFD code are normalized values

$$f(x) = \begin{cases} \frac{7}{4} - \frac{3}{4}\,cos(\,(\frac{1}{5}\,x - 1)\,\pi), & \text{if } x < 5 \\ \frac{5}{4} - \frac{1}{4}\,cos(\,(\frac{1}{5}\,x - 1)\,\pi), & \text{otherwise,} \end{cases} \tag{40}$$

a diagram of which is shown in Figure 27.

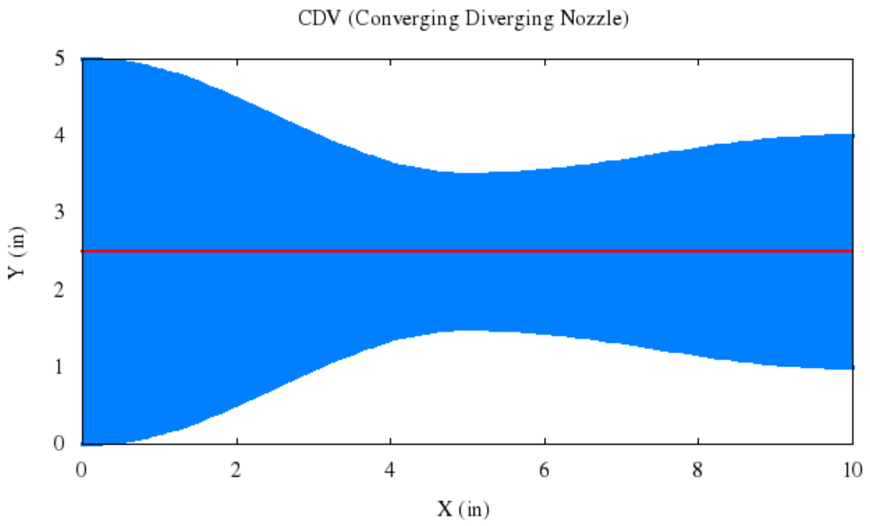


Figure 27: Converging-Diverging nozzle Validation (CDV) case [adapted from [91]]. The red line down the middle is the centerline.

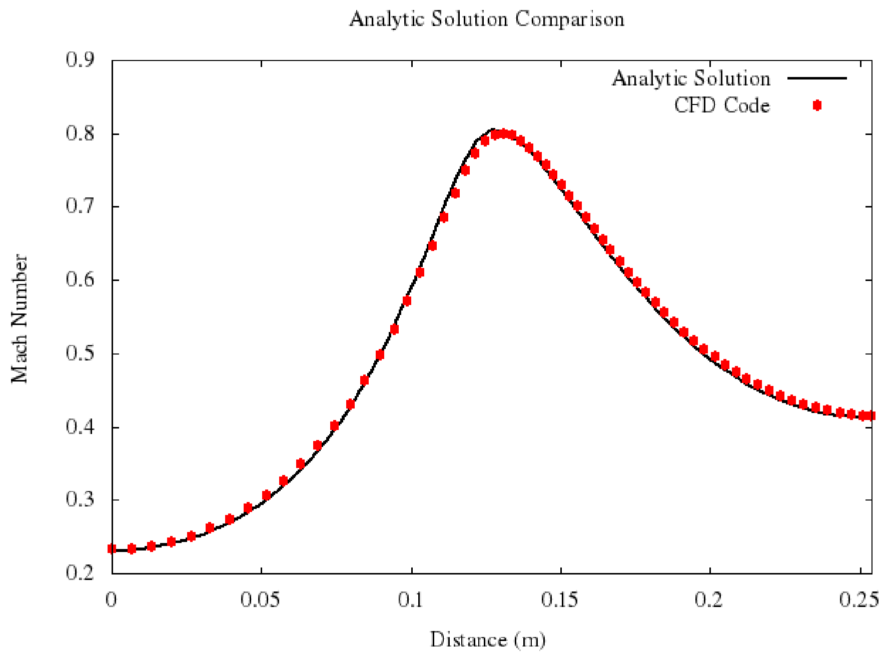
The code is run with specific inlet properties of

$$\begin{aligned} T_{tot,in} &= \quad 100 \text{ R} \;/\; 55.5556 \text{ K} \\ p_{tot,in} &= \quad 1.0 \text{ psi} \;/\; 6894.76 \text{ Pa} \;/\; 0.068046 \text{ atm,} \end{aligned}$$

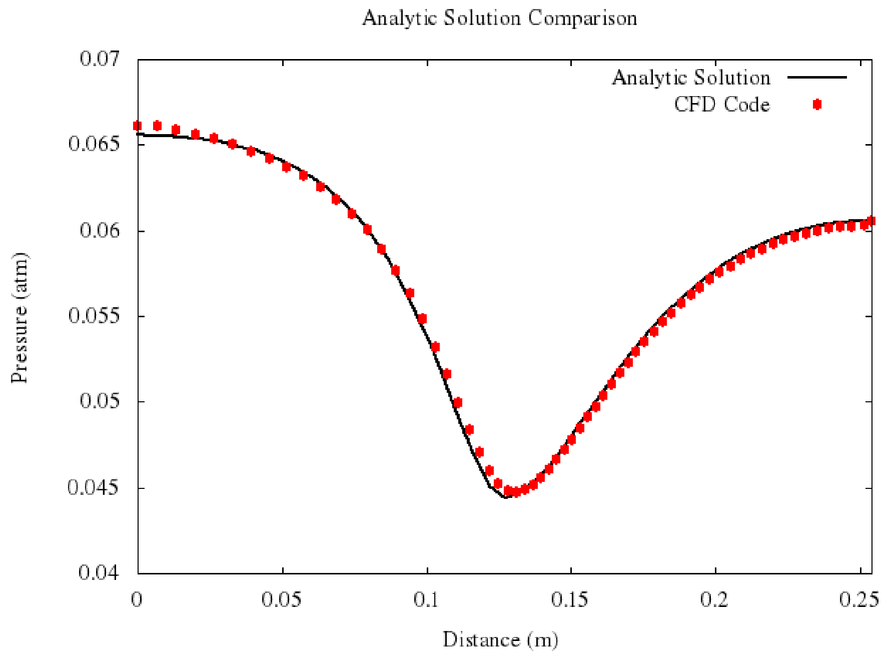
and three varying static outlet pressures of

$p_{in}/p_{out} = 0.89$ (subsonic)
$p_{in}/p_{out} = 0.75$ (supersonic, with normal shock in the diffusing section)
$p_{in}/p_{out} = 0.16$ (supersonic outflow).

Each of the three pressure conditions are run on a 65x20 node grid. The Mach numbers and pressures are computed at different places along the x-axis and compared

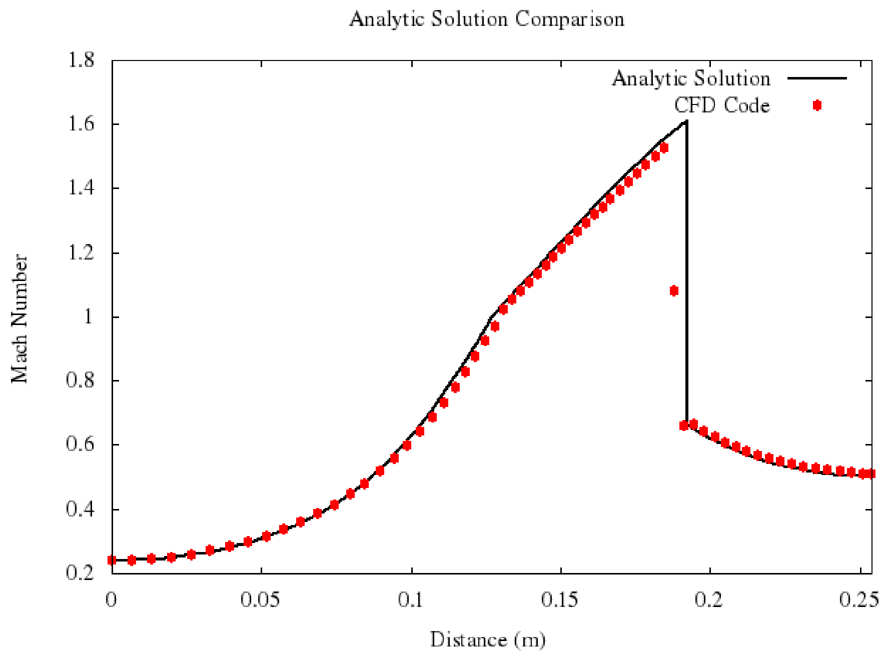


(a) Analytic solution vs. CFD solution of Mach number for $p_{out}/p_{in} = 0.89$

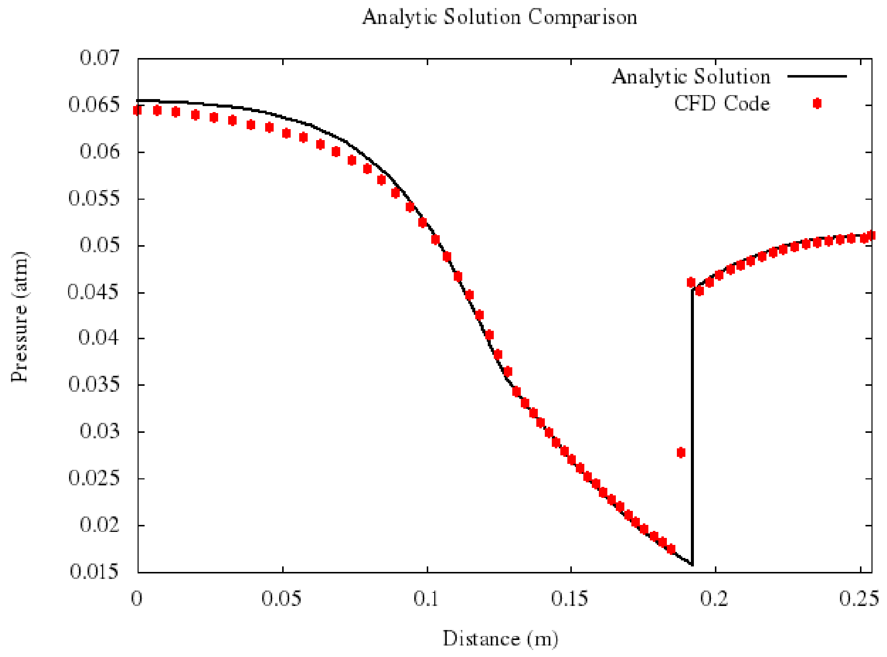


(b) Analytic solution vs. CFD solution of pressure for $p_{out}/p_{in} = 0.89$

Figure 28: Mach number and pressure analytic solution validation for $p_{out}/p_{in} = 0.89$. The grid is 65x20 nodes $(1,216$ cells) and converged to an absolute RMS$\approx 1.0 \times 10^{-12}$.

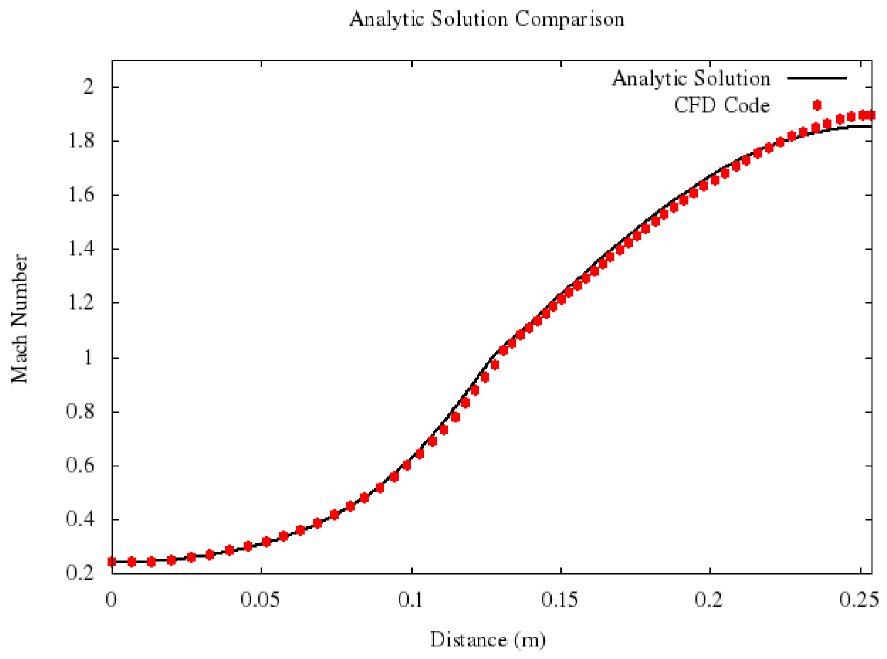


(a) Analytic solution vs. CFD solution of Mach number for $p_{out}/p_{in} = 0.75$

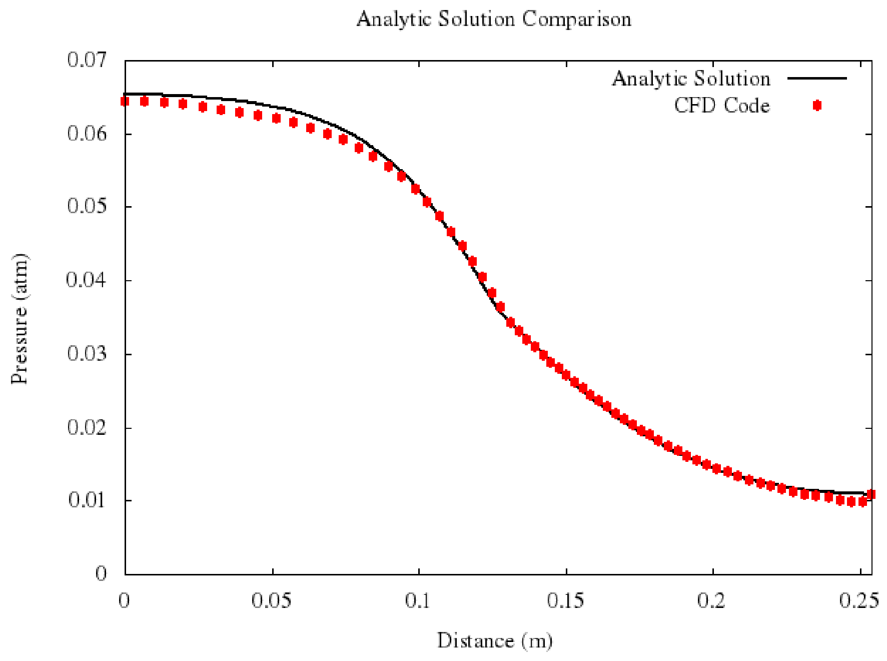


(b) Analytic solution vs. CFD solution of pressure for $p_{out}/p_{in} = 0.75$

Figure 29: Mach number and pressure analytic solution validation for $p_{out}/p_{in} = 0.75$. The grid is 65x20 nodes $(1,216$ cells) and converged to an absolute RMS$\approx 1.0\times10^{-12}$.



(a) Analytic solution vs. CFD solution of Mach number for $p_{out}/p_{in} = 0.16$



(b) Analytic solution vs. CFD solution of pressure for $p_{out}/p_{in} = 0.16$

Figure 30: Mach number and pressure analytic solution validation for $p_{out}/p_{in} = 0.16$. The grid is 65x20 nodes (1,216 cells) and converged to an absolute RMS$\approx 1.0\times10^{-12}$.

to the analytic solutions, which are known [91]. The results are shown in Figures 28 through 30.

The numerical solutions do not *exactly* match the analytic results for a couple of reasons. First, a higher fidelity grid would give a solution that is much closer to the analytic result. Additionally, the fluxes are computed using a first order approximation as opposed to utilizing higher order information in the form of extrapolations, such as the MUSCL scheme [92, 93]. However, note that the CFD code does a better job at capturing the actual shock than the example validation study in [91]. The flux vector splitting method which is implemented to capture large changes in properties does its job here to accurately capture the velocity discontinuity, as compared to the WIND CFD code [91, 94], shown in Figure 31.

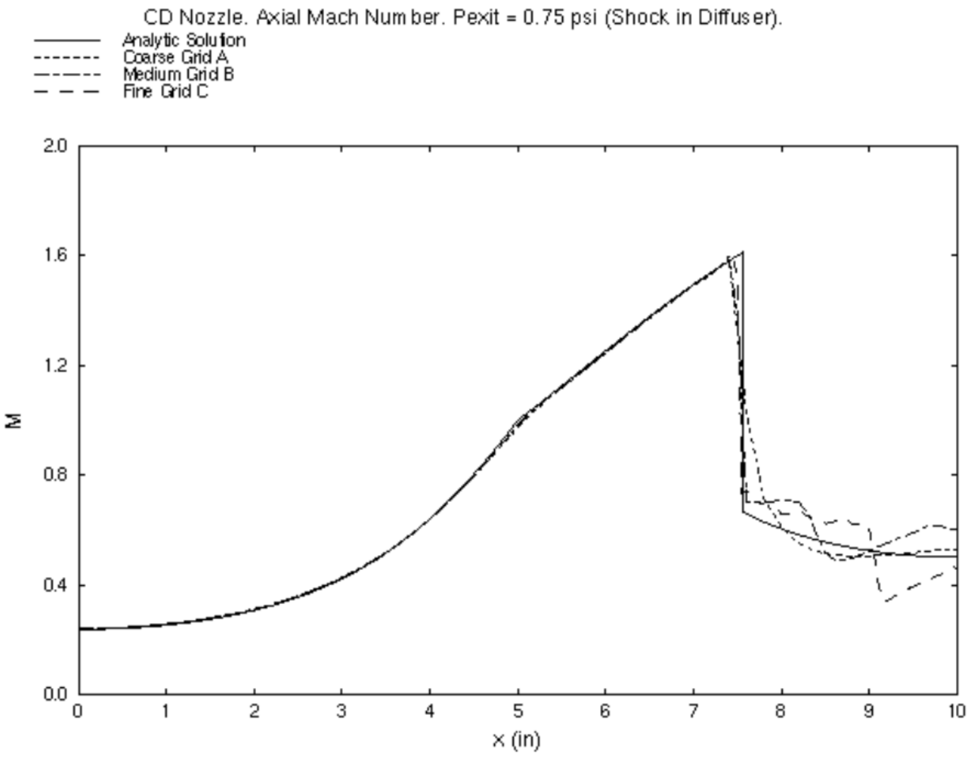


Figure 31: Mach number result from [91].

A second analysis is completed to check for the conservation of mass. Since this is a finite volume code which solves conservation laws, mass must be conserved throughout the domain. In order to complete this verification, seven grids are created with varying numbers of cells. These grids have a shape which is very similar to that of the CDV nozzle in Figure 27, however it is changed to make it a bit less symmetric (i.e. the throat is not in the exact center of the domain). The run conditions are the same as the CDV case above. Each of the grids are displayed in the Appendix in Figure 83. They are all run until a steady-state solution is achieved, at which point the input/output mass flow rates are recorded, as shown in Figure 32. Here, the difference between inlet and outlet flow rates decreases substantially as the number of grid cells are increased. With only 1,216 cells, mass is conserved to ∼0.35%. Increasing to the most dense grid of over 800,000 cells, mass is conserved to between a few thousands to a few hundredths of a percent (depending on the pressure ratio). This is just as one anticipates, since higher fidelity grids conserve and predict properties to better approximations.

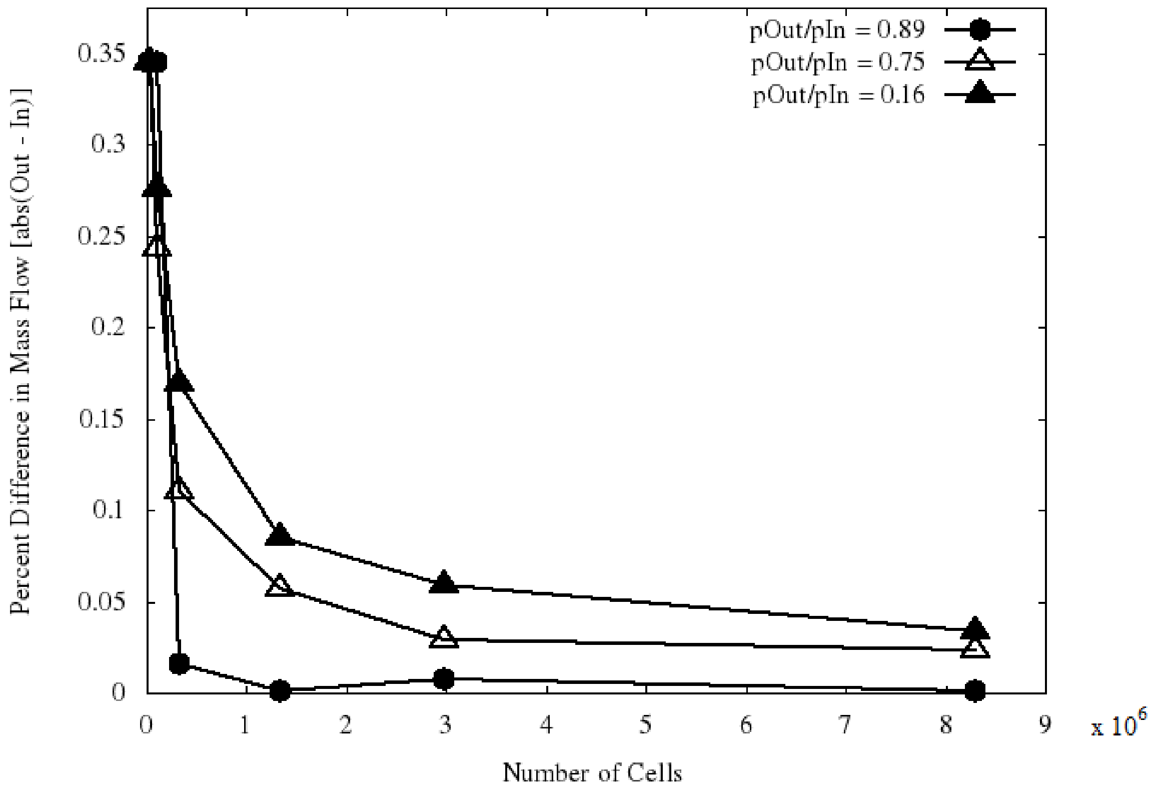
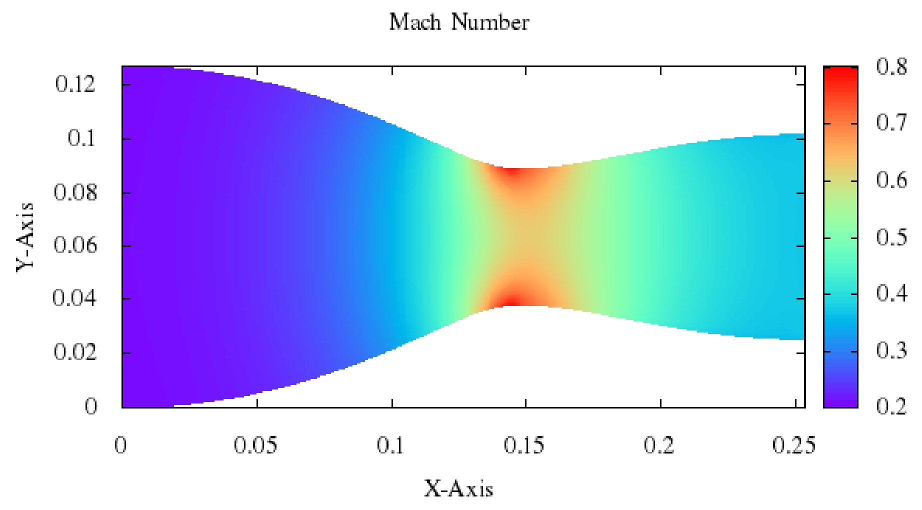
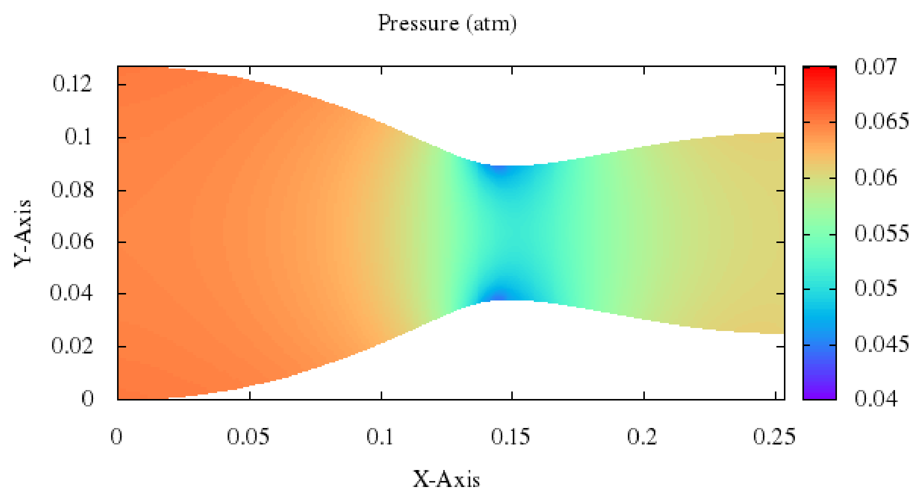


Figure 32: Differences between inflow and outflow rates for varying grid fidelity.

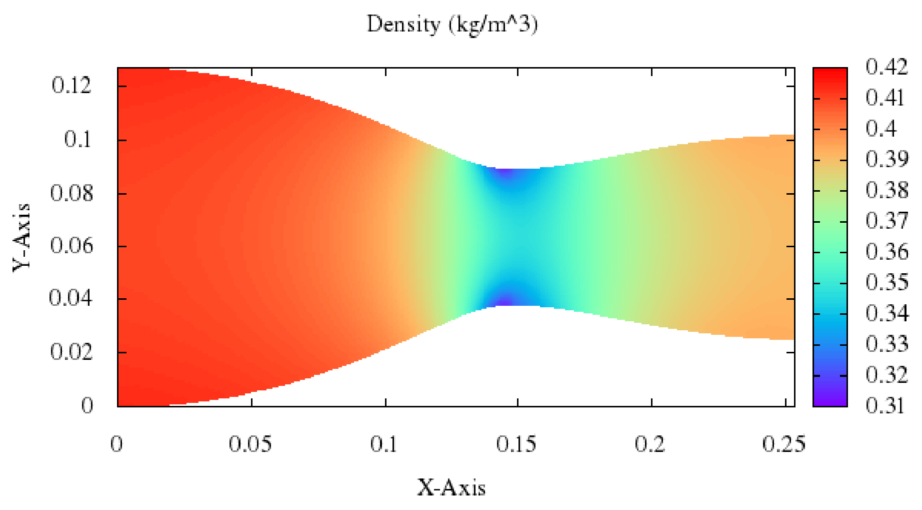
The four properties of interest are the pressure, density, Mach number, and temperature. The solutions for each pressure ratio are shown in Figures 33 through 35 for a mid-fidelity grid (331x100 nodes, with 32,760 cells). Figure 36 show the RMS error as the solutions converge for each pressure ratio. The solutions for the rest of the grids are shown in the Appendix in Figures 84 through 101 for the pressure, density, Mach number, and temperature, and Figures 102 through 107 for the RMS errors.



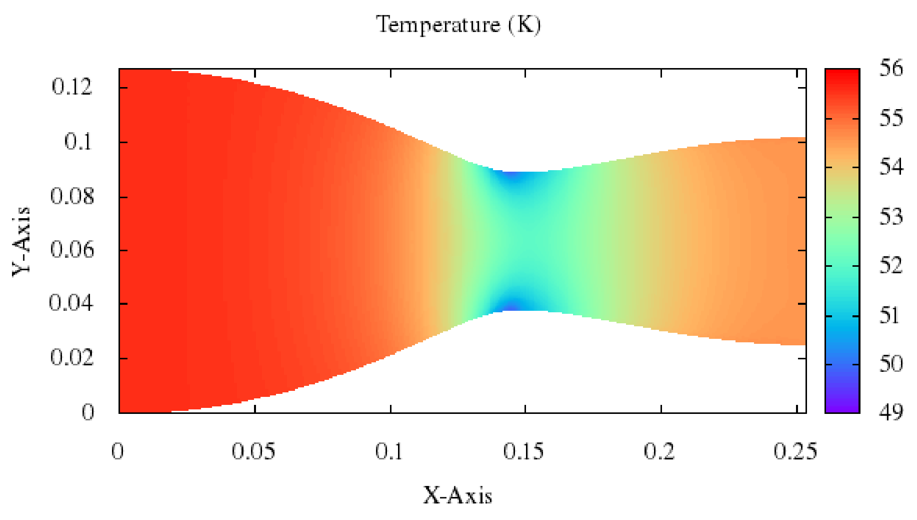
(a) Mach number contours



(b) Pressure contours

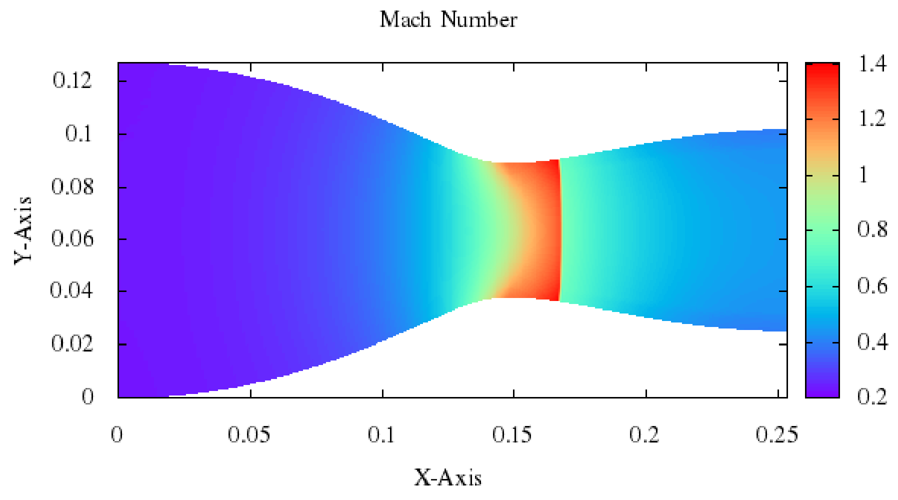


(c) Density contours

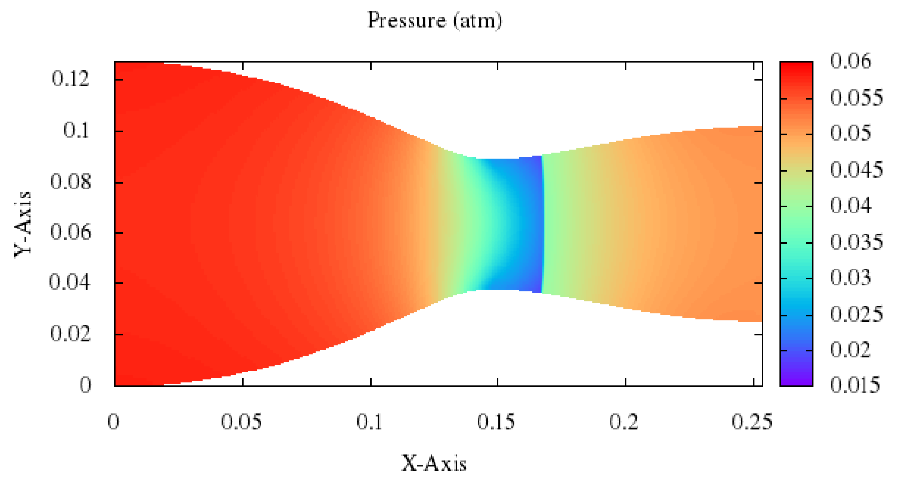


(d) Temperature contours

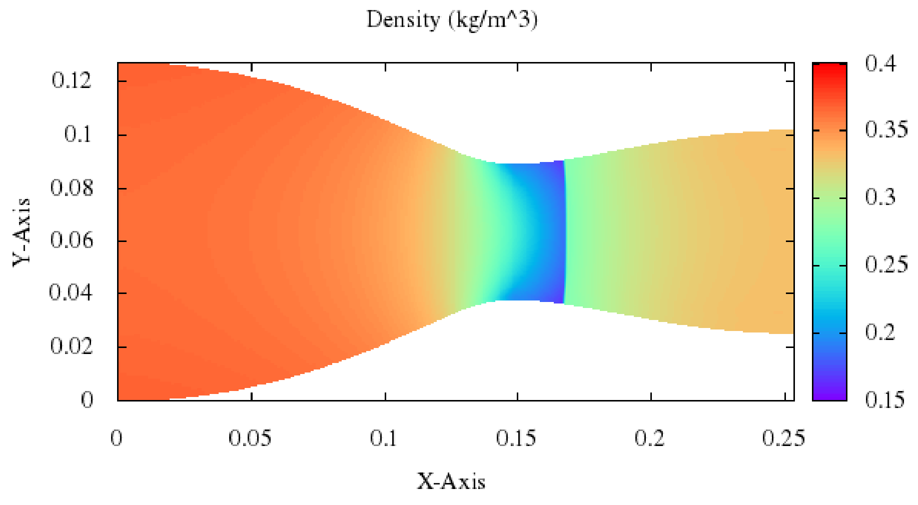
Figure 33: Property contours for converged solution with 32,760 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



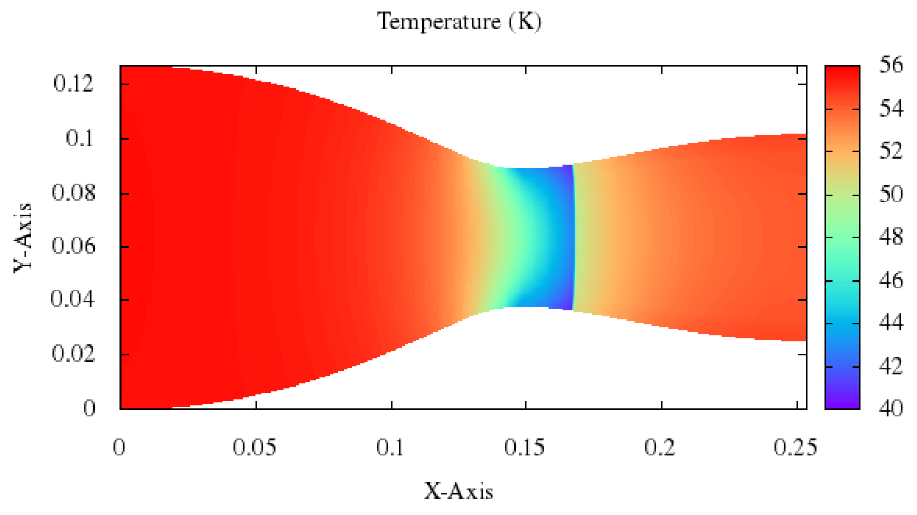
(a) Mach number contours



(b) Pressure contours

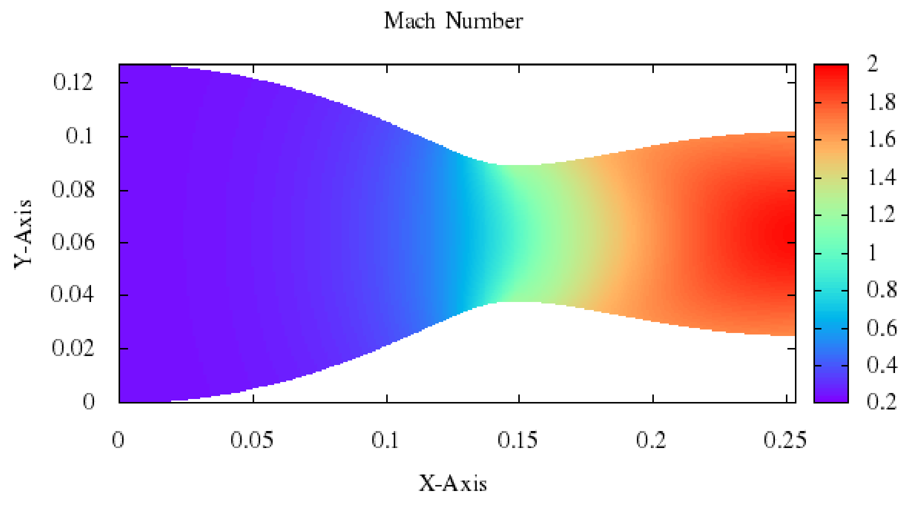


(c) Density contours

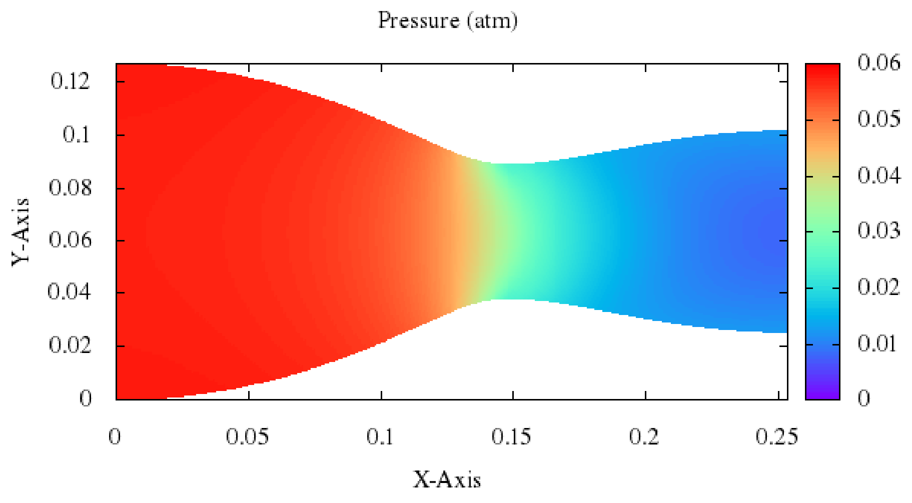


(d) Temperature contours

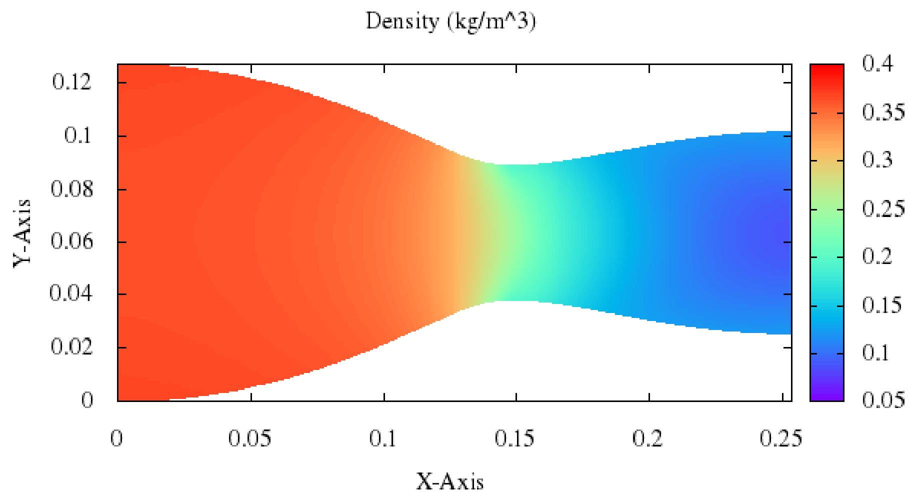
Figure 34: Property contours for converged solution with 32,760 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



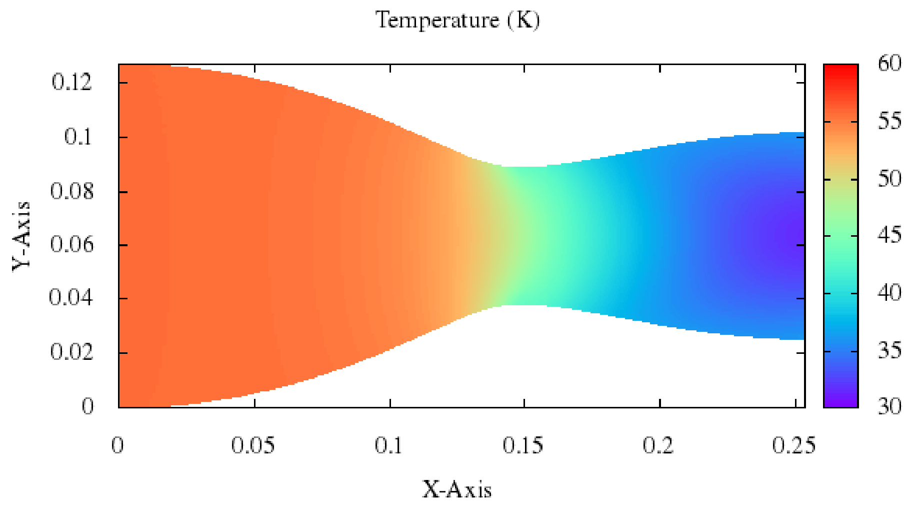
(a) Mach number contours



(b) Pressure contours

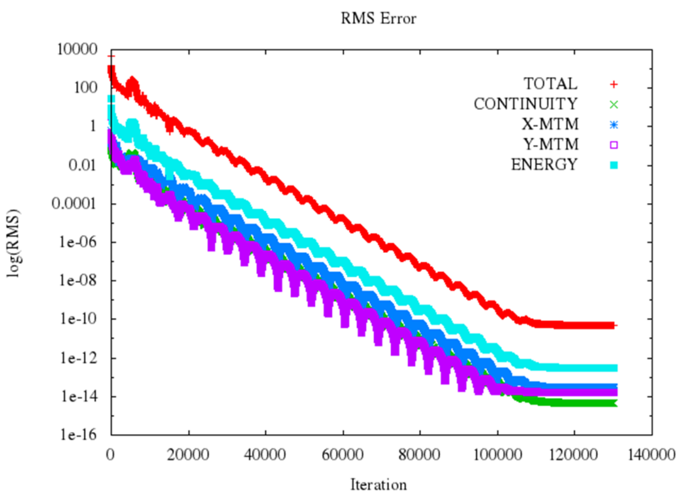


(c) Density contours

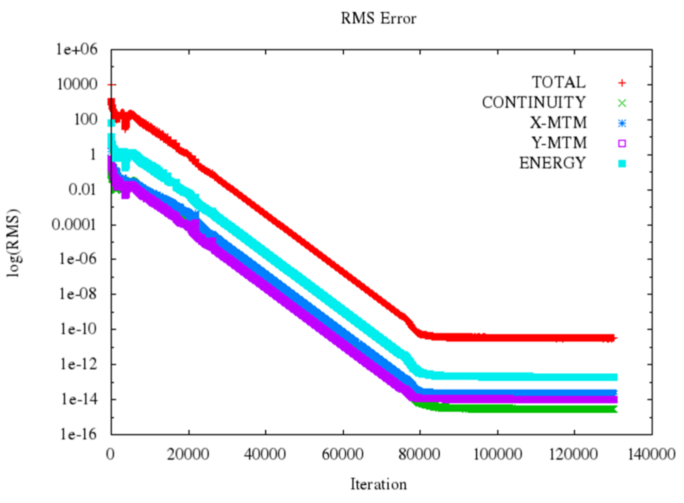


(d) Temperature contours

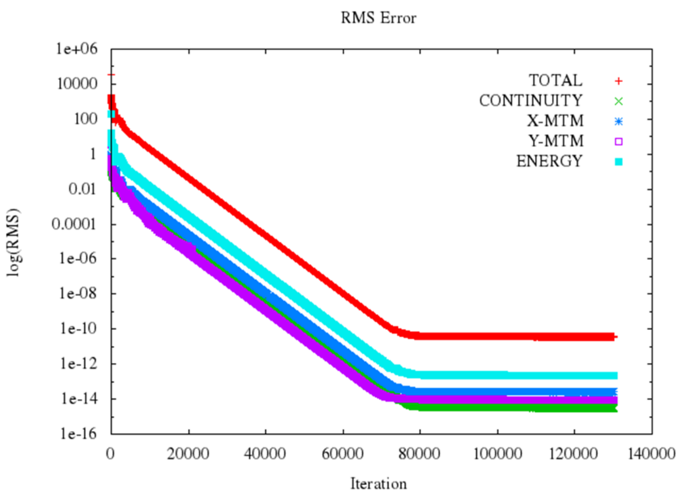
Figure 35: Property contours for converged solution with 32,760 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.



(a) $p_{out}/p_{in} = 0.89$



(b) $p_{out}/p_{in} = 0.75$



(c) $p_{out}/p_{in} = 0.16$

Figure 36: Absolute RMS Error for converged solution with 32,760 cells.

4.2 Chemical Kinetics Solver

As a general statement, the combustor is simulated using multiple reactors which tie together into a network. This chemical kinetics network (CKN) is necessary in order to obtain chemical information about the combustion process and input it back into the flow field. This network must be capable of forming a structure of any shape deemed necessary by the algorithm which searches through the converged flow field and dictates the network's architecture (Section 4.3). Additionally, between the iterations in Figure 44, the CKN must be capable of changing shape and connections, making it general enough to accept any number of reactors which may be interconnected in any different way. Also, the CKN has to handle recirculation since flameholding will be present within the combustor. Perfectly Stirred Reactors (PSRs) are chosen as the reactor model since they allow one to solve a set of coupled ODEs as opposed to solving coupled PDEs that must be solved if running reacting flow CFD[3].

Due to the high rate of mixing assumption, a perfectly stirred reactor operates in the extremely low Damkohler number limit ($Da << 1$), and is one of the simplest forms of a chemically reacting system. All thermodynamic properties are homogeneous throughout the volume, and all thermodynamic properties (but not chemical properties) change instantaneously. A PSR abides by the governing equations [95]

$$\frac{d}{dt}(\rho V) = \dot{m}_{in} - \dot{m}_{out}, \tag{41}$$

which is the conservation of mass equation,

$$\frac{d}{dt}Y_i = \tau_{res}(Y_{in} - Y_i) + \frac{\dot{\omega}_i \bar{W}_i}{\rho}, \tag{42}$$

[3]Or algebraic equations if using a steady-state PSR model.

which are the conservation of species equations, and

$$c_{p_{mix}} \frac{d}{dt} T = \tau_{res} \sum_i Y_{i,in}(h_{i,in} - h_i) - \frac{1}{\rho} \sum_{i=1}^{N} h_i \omega_i \bar{W}_i + \frac{\dot{q}_{in}}{\rho} - \frac{1}{\rho} \frac{d}{dt} p, \tag{43}$$

which is the energy conservation equation. The Equations 41, 42, and 43 are closed using the ideal gas equation of state,

$$p = \rho R_{mix} T. \tag{44}$$

In the above equations, $\dot{m}$ are mass flow rates

4.2.1 Cantera

The CKN is developed using Cantera routines [60], since this provides an easy and versatile environment for chemical reactor network development. An ideal gas sits within each PSR, which is taken to be a good approximation since the pressures and temperatures inside of the combustor are relatively high. A ”reservoir” environment is used to feed gases into the domain and collect gases out of the domain, which also have ideal gases sitting inside of them. A ”mass flow controller” is used to determine how much mass flows from the environment reservoir to the reactors. Additionally, objects are used to keep the PSR at a constant pressure. This can be done via a wall or a valve. Walls are rarely used in this study for a number of reasons. First, they do not guarantee that the volume of the reactor will stay constant, since the wall is simply a flexible membrane. Second, they may add additional stiffness to the ODEs and cause long system runtimes. Instead, valves are typically used to keep the pressure and volume of the reactors constant. This is a better choice since the the volume represents a physical space within the combustor. Figure 37 shows the simplest version of a constant pressure PSR.

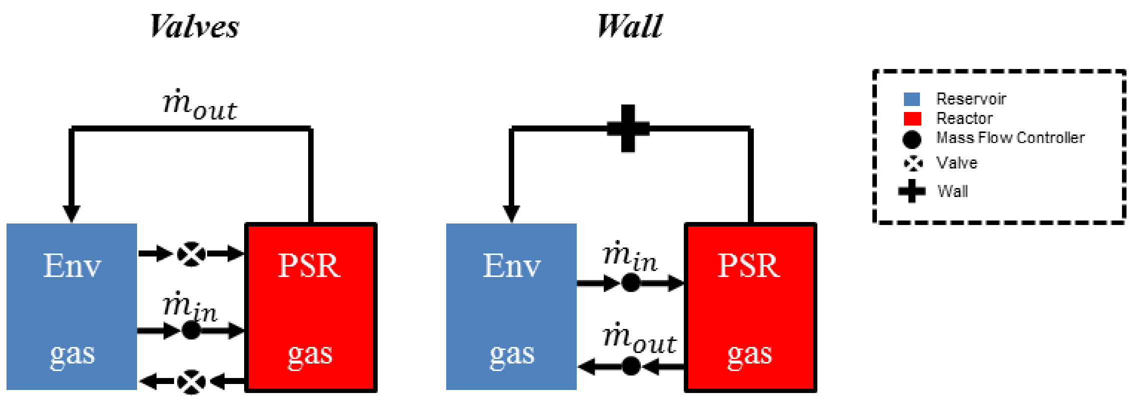


Figure 37: Simplest version of an ideal gas reactor which can be created in Cantera. An ideal gas sits within the environment (which acts as an infinite supply of this gas), and a gas sits within the reactor. A mass flow controller is used to control the flow into the reactor, and either two valves are used (one in each direction) to keep the reactor at constant pressure (left), or a wall (right).

A network can (and indeed does) get much more complicated, consisting of many different environments, reactors, mass flow controllers, valves, etc. As an example of the level of complexity a network may subsume, Figure 38 shows a reactor network with many different interconnections between environments, reactors, recirculators, etc. In this example, each of the sub-networks (environment plus PSRs going from left to right) need not have the same number of reactors. In this example, sub-network 0 has three reactors, sub-networks one, two, and five have 4 reactors, sub-network three has 1 reactor, and sub-network four has 2 reactors. The last reactor in each of the sub-networks dumps into a reservoir (which represents combustor outflow). Additionally, recirculation reactors and environments are present to mix fresh gases with previously burned gases. In the diagram, sub-network zero, reactor 0 gets recirculated products which consists of fresh air (and perhaps fuel). Lastly, there is one standard interconnection and one recirculated interconnection within the network. The standard interconnection is sub-network three, which consists of reactor zero sending flow to sub-network two, reactor zero. The recirculated interconnection consists of sub-network four, reactor one recirculating flow with sub-network two, reactor two. Note that for recirculated flow, the flow rates for recirculation maye not be the same! For example, it is perfectly acceptable to send $5\frac{kg}{s}$ of flow in one direction of a recirculated reactor, but only $3\frac{kg}{s}$ back into the recirculation stream, since the flow physics within the combustion chamber may very well separate the flow in this manner. Within Cantera, a reactor or reservoir may connect to as many reactors as one deems necessary, given they are connected with a different mass flow controller. Therefore, using a series of mass flow controllers, reactors, and environments, one can create any network one so chooses.

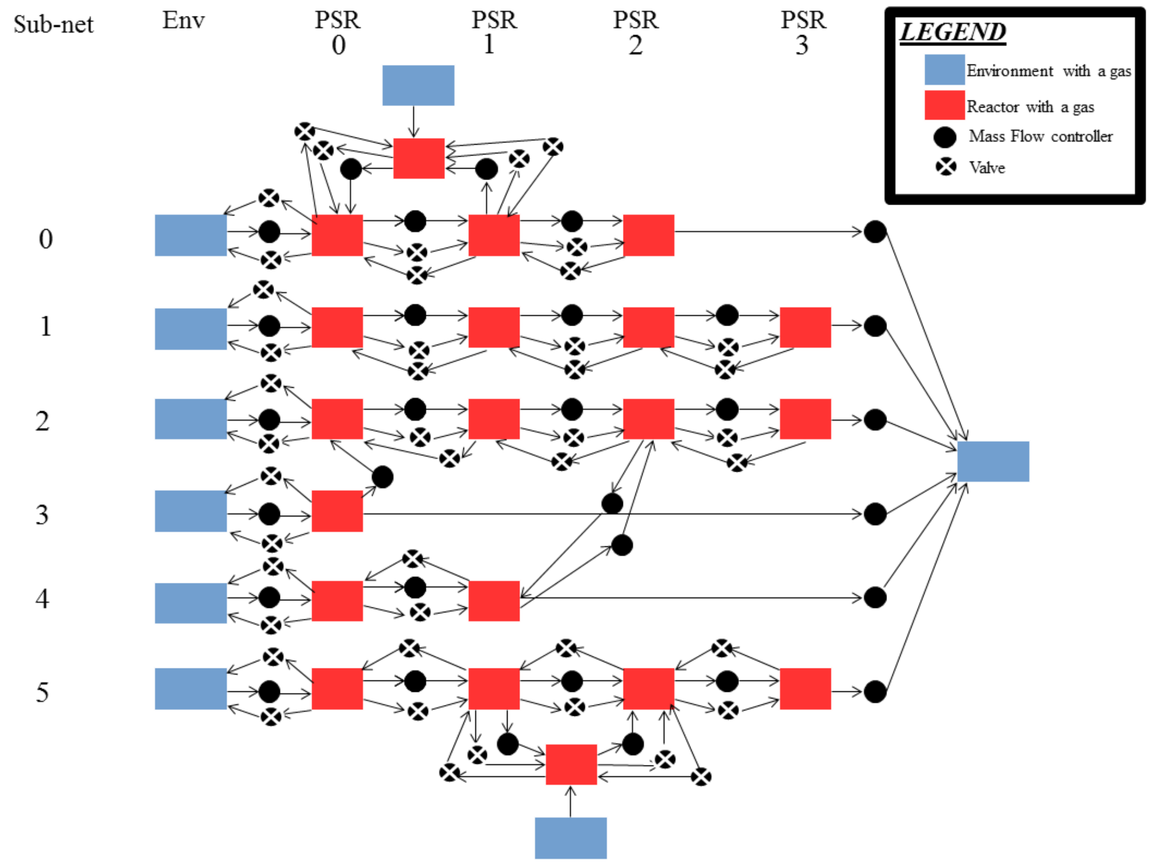
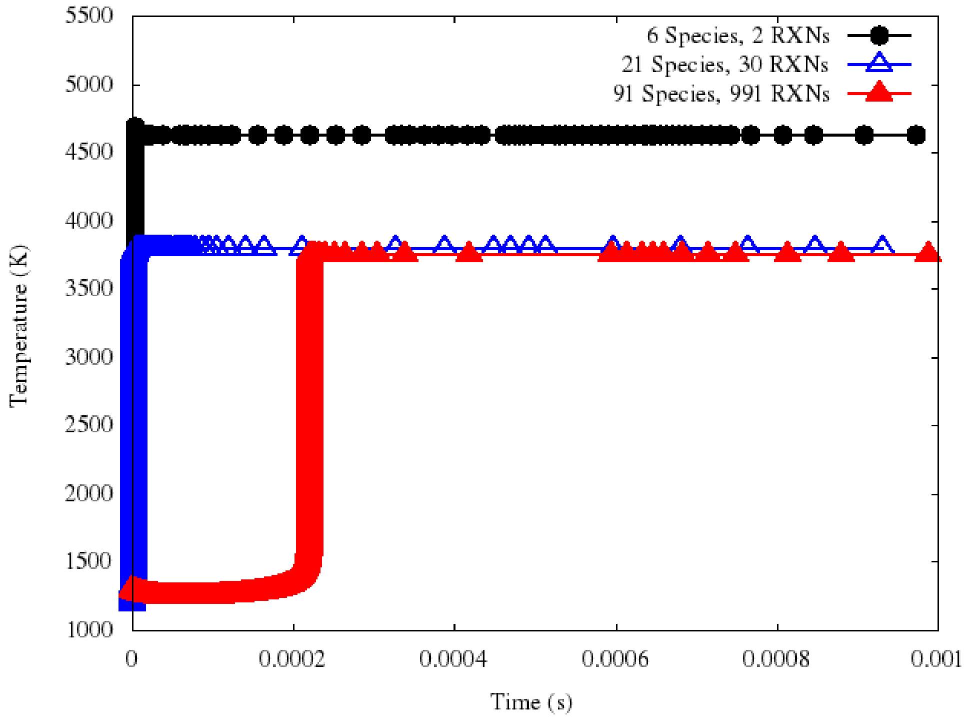


Figure 38: A complicated Cantera example. Here, many environments feed into the initial PSRs, flow is split between PSRs and other reactors, recirculation occurs within the network and between new environments and the network, etc.

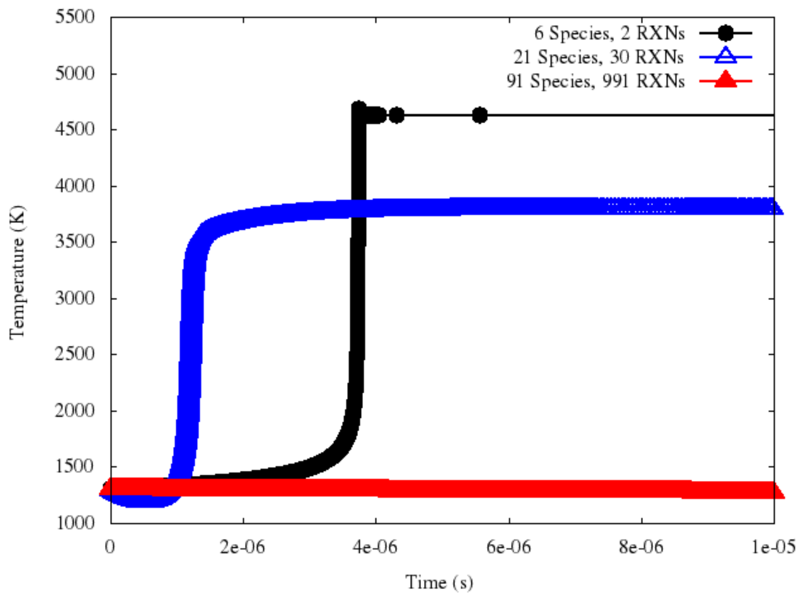
Lastly, a chemical mechanism must be chosen so that the properties of the gases in each of the reactors are known, including available species and reaction steps. It is very easy to swap chemical mechanisms in and out using Cantera, since everything just relies on an input file; so switching mechanisms does not depend on changing any algorithms or source code. Many hydrocarbon mechanisms are tested, including three kerosene type fuel models of varying fidelity, since one ultimately desires a Jet-A type fuel model. The first Jet-A kinetics scheme is a two step chemical mechanism for $C_{10}H_{20}$ involving six species called "2S_KERO_BFER" [96, 97], which is the lowest fidelity Jet-A mechanism that is tested here. The second is the Kollrack mechanism ($C_{12}H_{23}$), which consits of 21 species and 30 reactions [9]. Lastly, a high fidelity mechanism is used, which consits of 91 species 991 reactions scheme involving $C_{10}H_{22}$ (the "Luche" mechanism [98]).

Just as an initial test, these three mechanisms are run in a single PSR. Each reactor is initialized with solely oxygen and fuel in a stoichiometric equivalence ratio at $T = 1300K$, while oxygen and fuel are added at $T = 800K$ at $\phi = 1$ at a rate of $1\frac{kg}{s}$, and products are removed at the same rate. Since ignition time depends on the species present (and each mechanism has different species), they are not expected to necessarily react at the same time. However, Figure 39 shows that the two step mechanism reaches a much higher steady-state temperature, almost $1000K$ higher than the other two.

A two step model will run very quickly due to the small number of species to track. It may be used to get heat release (and therefore converge on the correct flow field) before the large mechanisms are run to give emissions results. The Kollrack mechanism can be utilized to get a good estimation of heat release and run thermal NOx simultaneously, since it is a good compromise between a relatively small reaction scheme and also containing an actual NOx mechanism. The 91 species mechanism ought to give the best heat release, since it contains more details about how the reaction proceeds, and takes many smaller molecules and other reactions into account,



(a) Temperature as a function of reactor time within a single PSR for three Jet-A models.



(b) Zoom in on smaller times in order to show the start up temperature profiles.

Figure 39: Temperature comparison of three Jet-A mechanisms.

including NO. Note that many larger mechanisms exist [99, 100], and can include hundreds or even thousands of species and almost ten thousand reactions. The higher fidelity the mechanism, the better one is able to capture the actual kinetics behavior, however this is at the expense of runtime.

These three mechanisms are the only openly available Jet-A models which were found and tested. The biggest concern here is the fact that these models differ greatly in chemical properties, specifically its initial starting composition. This is not surprising since Jet-A is typically modeled using kerosene, which is itself modeled using many different types of fuels [101]. However, it is difficult to quantify the effects of mechanism size since each mechanism represents a different starting composition. Although the user has the freedom to choose whichever mechanism is desired, this makes it difficult to determine if any results are accurate since there are so many fuel types which are considered "Jet-A" and "kerosene". To alleviate this problem, the utility of the methodology will be presented with a smaller hydrocarbon fuel. Methane is therefore used, since it's chemical formula is known (CH_4), and chemical mechanisms are much more common for methane than they are for kerosene. Additionally, the GRI 3.0 model for methane [35] is very popular and used widely throughout experimentation and literature. The focus here is on a computational method for making the design more feasible, so developing a correct and usable kinetics scheme is not of concern, since future schemes will be developed which will undoubtedly be more accurate.

Tying all of this together, the aim is a network of these Cantera reactors, where each PSR sends and/or accepts flow from another. A system of reactors is formed, where the network simultaneously solves all of the equations in 41 through 43 for each PSR. To do this,

$$n_{sys} = N \times (S + 3) \tag{45}$$

equations are formed, where N is the number of reactors in the network and S is the number of species in each reactor, since each reactor's unknowns are its mass, volume, temperature, and all species. To accomplish this, Cantera uses a modified form of equations 41 through 43, specifically [102]:

$$\frac{d}{dt}(m) = \sum_{in} \dot{m}_{in} - \sum_{in} \dot{m}_{out}, \tag{46}$$

for the conservation of mass (as in Equation 41),

$$m\frac{d}{dt}Y_k = \sum_{in} \dot{m}_{in}(Y_{k,in} - Y_k) + \dot{m}_{k,gen}, \tag{47}$$

for the conservation of species (as in Equation 42), and

$$mc_v\frac{d}{dt}T = -p\frac{d}{dt}V - \dot{Q} + \sum_{in} \dot{m}_{in}(h_{in} - \sum_{k} u_k Y_{k,in}) - \frac{pV}{m}\sum_{out} \dot{m}_{out} - \sum_{k} \dot{m}_{k,gen} u_k \tag{48}$$

for the conservation of energy (as in Equation 43). Adding flow devices does not add more equations, however it does add more terms to these equations, as do the amount of reactions realizable in the chemical mechanism [103]. Hence, the addition of some species make the equations *stiff*, causing solvers to take small steps to solve said systems.

It is very important to note that this same type of model can be used to predict other types of emissions other than just NOx, such as soot. Though it may involve more transport equations within the CFD code, Cantera has previously aided in the computation of soot formation [104]. Each of the models have different runtimes depending on the situation, since different chemical properties and thermodynamic conditions change the stiffness of the system of equations. From Section 3.2, the initial reaction mechanism is not expected to be large, since it is stated before that a global reaction mechanism is expected to be good enough to capture the correct heat release. A large mechanism is not expected to be needed until the very end when NOx is computed, however this assumption is still explored in the research questions. Specifically, Chapter 5 discusses whether a single global mechanism releases the "correct enough" heat which can be used for flow field iterations.

4.2.1.1 Chemical Kinetics Validation

The Cantera library is an openly distributed software package and therefore does not need to be validated as if it is a code written from scratch. However, Cantera allows the user to create chemical networks by tying functions together. Therefore, the method by which the network is created must be validated. Specifically, test cases must be run and analyzed to ensure the code works correctly and the correct connections have been made.

The start of the code involves defining gases at some T, P, and a chemical composition vector, $\vec{s}$. When the code starts, reactions ensue, and after a given reaction time (imposed by the solver), the gas starts to react and moves through the reactors at a rate determined by the mass flow controllers and the characteristic times (τ_{res} of the reactors. The reactors immediately start exchanging heat and radicals, so any reactor downstream will get more heat and radicals than those upstream (and those connected to reservoirs). Recirculated flow will build up radicals and heat faster as well, but this also depends on how much unreacted flow is mixing in the recirculated reactor. Figure 40 shows a simple test network, and the results using the GRI 3.0 mechanism are shown in Figure 41.

Since the incoming thermodynamic and flow conditions from both of the environment reservoirs are the same, the last reactor in the first sub-network reacts first. This is because the first reactors in each sub-network recirculate with each other,

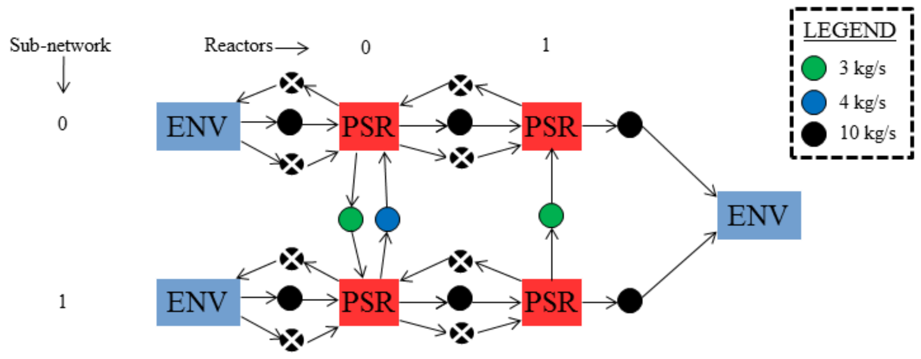


Figure 40: Example network used to run tests. This network consists of one inter-sub-network recirculator between the first reactors in sub-networks 0 and 1, and another-inter sub-network connection from reactor 1 in sub-network 1 to reactor 1 in sub-network 0.

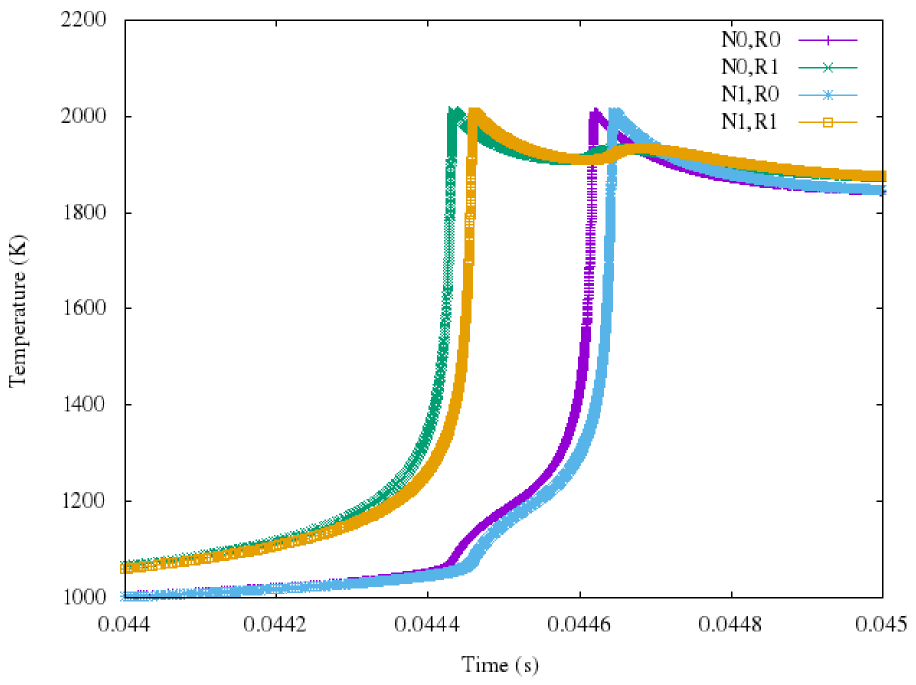


Figure 41: Results from network configuration in Figure 40. Purple: Reactor sub-network 0, reactor 0, Green: Reactor sub-network 0, reactor 1, Blue: Reactor sub-network 1, reactor 0, Yellow: Reactor sub-network 1, reactor 1. The incoming temperature is $800K$, the volumes of each reactor are $1m^3$, and reactor sits at 30 atmospheres.

however reactor $0, 1$ receives extra heat and radicals from reactor $1, 1$. One can see in Figure 41 that the temperature in reactor $0, 1$ spikes first, due to the larger amount of radicals it is receiving, but is followed closely by the only other reactor where recirculation does not occur. Next, the two first reactors in each sub-network are exchaning radicals and energy, but the reactor in the first sub-network is getting radicals and energy slightly faster and therefore reacts only slightly sooner.

To ensure that the results are what one expects, the mass flow rates for recirculation are changed between reactors 0 in sub-networks 0 and 1. Additionally, the mass flow rate between the last reactors in each sub-network is increased. This is shown in Figure 42, and the results are shown in Figure 43. A shift in the curves are observed, notably the second reactor in sub-network 0. Flow is taken out of reactor 1 in sub-network 1 much faster, which purges it of radicals and causes reactor 0 in sub-network 0 to react more quickly. The next reactor to spike in temperature is the other downstream reactor, followed by the upstream reactors.

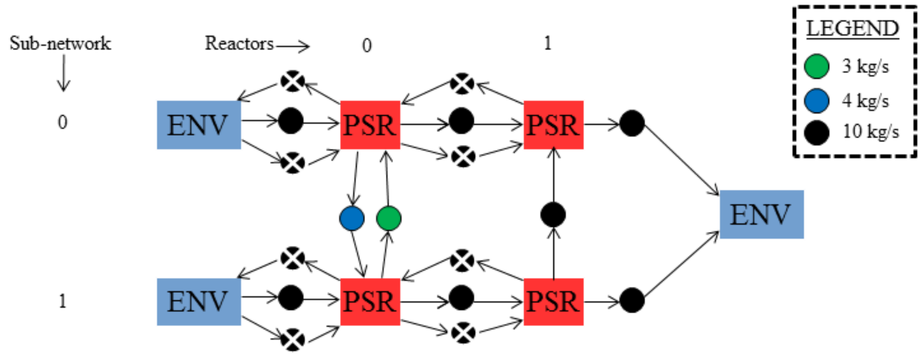


Figure 42: Second example network used to run tests. This network consists of one inter-sub-network recirculator between the first reactors in sub-networks 0 and 1, and another-inter sub-network connection from reactor 1 in sub-network 1 to reactor 1 in sub-network 0. This is the same as Figure 40, except for the fact that one of the recirculation reactor's mass flow has changed.

4.3 Method of Sparse Thermochemistry (MoST)

In this approach, the conservation of species equations (Equations 19) are NOT solved as a strictly coupled system with the rest of the flow equations, as is done in a standard reacting flow CFD simulation (see Section 3.2.2). A CKN is used instead, whose

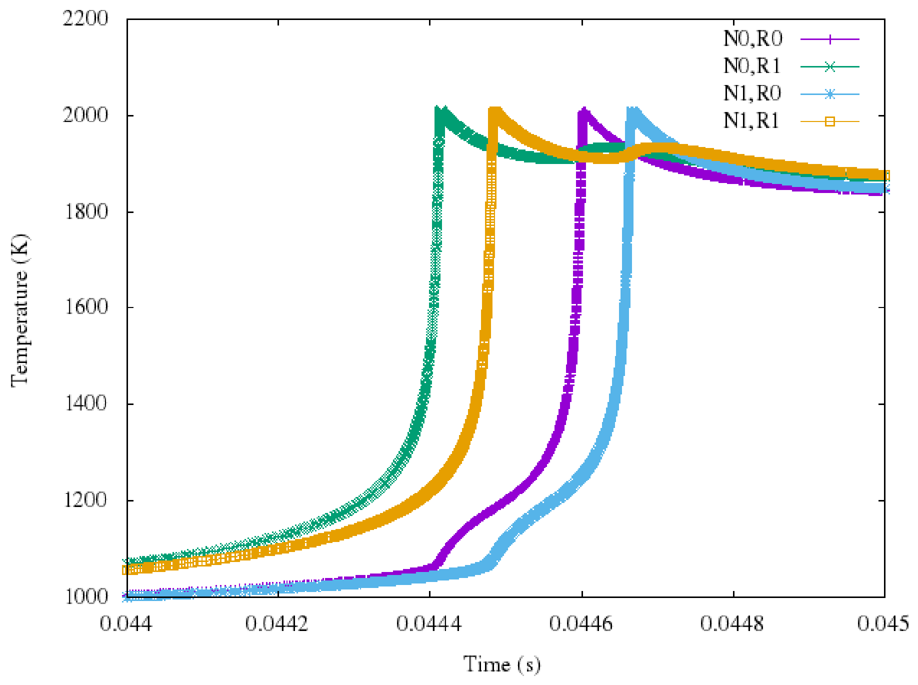
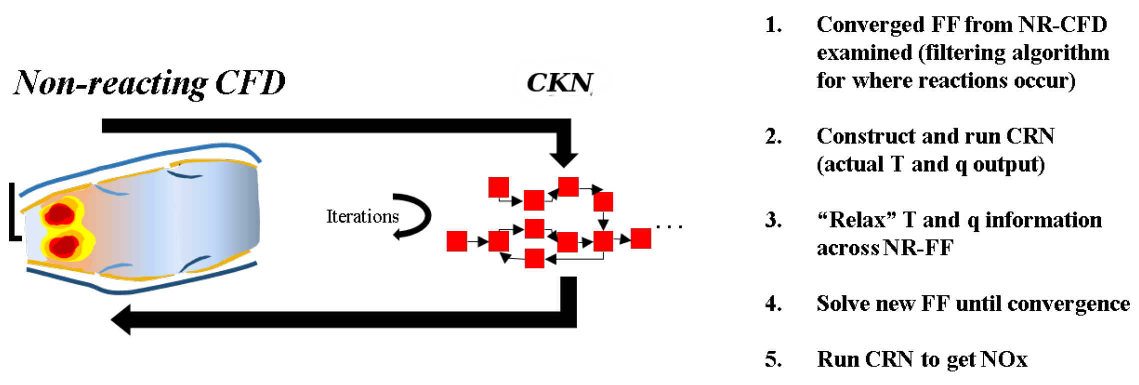


Figure 43: Results from network configuration in 42. The incoming temperature is $800K$, the volumes of each reactor are $1m^3$, and reactor sits at 30 atmospheres.

architecture is a direct result of how the flow field looks. This forms an iterative process until both the heat release and flow field (FF) converge, as in Figure 44, which is similiar to Figure 23, except the steps are listed on the right.

Tying the non-reacting CFD and Cantera codes together is not enough, since the result will still be two separate programs without an algorithm for coupling. One of the major goals of this work is to develop a method which determines where a reaction occurs within a flow field. The Method of Sparse Thermochemistry (MoST) is used to find a solution to this problem by resolving where the reactions occur. This is the main focus of the research, as described in Research Question I in Section 3.2.2. As Experiment I in Section 3.3 describes, this is a process by which numerous algorithms are attempted via trial and error in order to ascertain which properties are important for use, and which algorithms meet the task and perform best. This portion represents the bulk algorithm and the extremely novel portion of the work.



1. **Converged FF from NR-CFD examined (filtering algorithm for where reactions occur)**
2. **Construct and run CRN (actual T and q output)**
3. **"Relax" T and q information across NR-FF**
4. **Solve new FF until convergence**
5. **Run CRN to get NOx**

Figure 44: Methodology of the iterative FF/CKN technique. Heat release information is inserted into the CFD solution in the form of source terms. Non-reacting CFD is used to determine the structure of the flow field, and a chemical kinetics network is created to emulate this structure. Iterations are performed between the CFD-FF and CKN until convergence is achieved.

The first step is to run the non-reacting CFD code to completion, i.e. a converged non reacting solution which involves the correct combustor conditions in addition to tracking the mixture fraction for the fuel. This can be done with any CFD method or algorithm, as long as one has a converged non-reacting flow field solution to start with.

4.3.1 Unsuccessful Attempts Using Successive Filters

The first idea explored is that of various filters. Here, numerous filters are applied to the flow field so that areas where reactions occur are ”sought out” by the filtering process. Filters are applied until only the regions where reactions can occur are left. The filters which may be imposed are a position filter, a velocity filter, a mixture fraction filter, etc.:

Position Filter

This filter blocks out areas where reactions do not occur. For example, if one does not want reactions to take place along a wall, position filters are set to block out these areas. This may or may not be a good idea if one seeks wall interactions, etc. However, the option is available for any location desired.

Velocity Filter

The option to set a velocity threshold is available. If the flow is faster than this velocity threshold, it will not react. It will if it is slower. If one has interest in taking flame propagation speed for a premixed reaction into account, this is a good way to do it.

Mixture Fraction

The most utility comes from the mixture fraction. Fuel and air can react if they exist in certain proportions (e.g. a diffusion flame in stoichiometric proportion). In this case, the user chooses any range of mixture fractions or equivalence ratios that this filter should ignore, and only leaves points which are at acceptable fuel/air ratios when applied.

Thermodynamic Properties

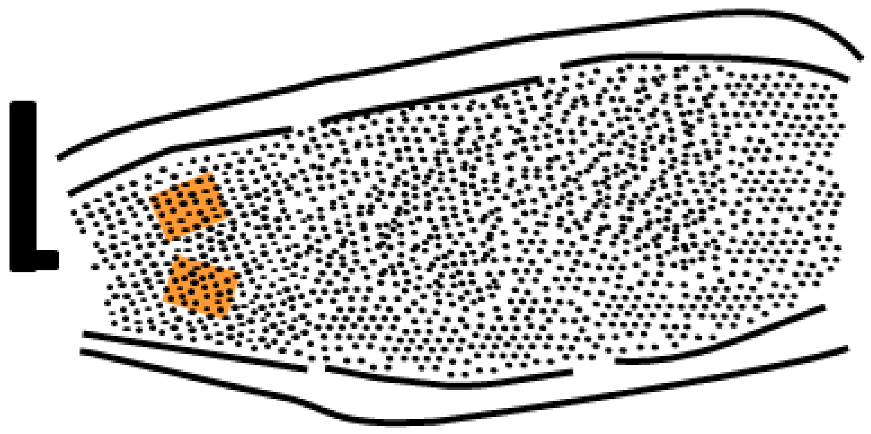
The user may filter the flow field based on any thermodynamic properties, such as pressure and density.

These filters need not be active all of the time. It is perfectly plausible to use none, one, any number, or all of these filters. They are simply turned off when necessary (e.g., the position filter may not prove any utility, so it may simply be ignored). Contrarily, if many filters prove their usefulness, more filters can be added as they become necessary and any number used. The idea here is that the regions where the reactions occur are found and reactors "placed" there within the simulation. That is, the reactors are superimposed on the flow field. Not only are reactors needed where reactions occur per se, but also at major flow features such as air/fuel inlets and recirculation regions, etc. Given the aforementioned statements, there are multiple approaches. Specifically, one may place reactors:

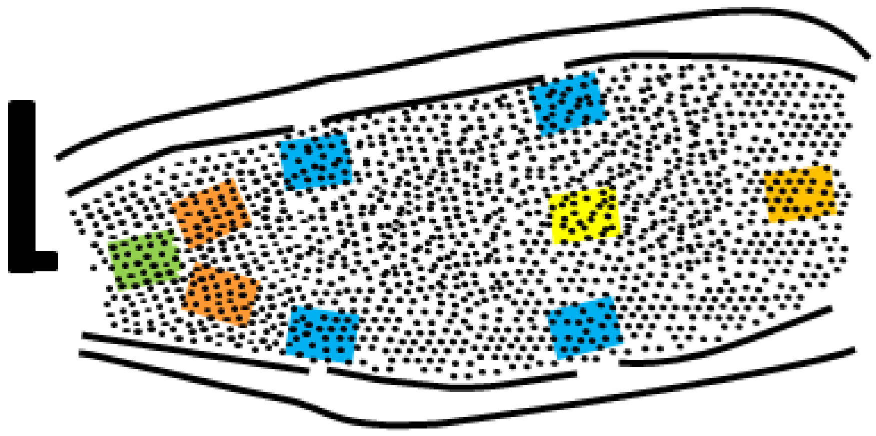
- Only in locations where reactions occur.
- At locations where reactions occur and where major flow features exist.
- Completely line the entire combustor with reactors.

Each choice has its merits and drawbacks. These ideas are displayed in Figure 45.

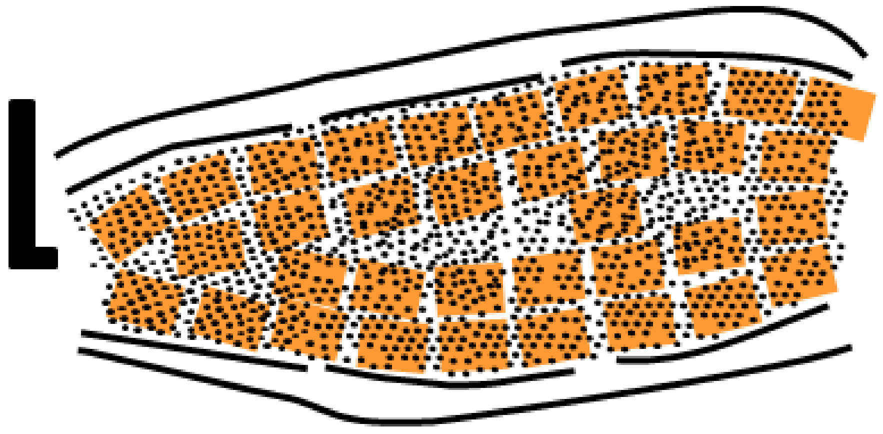
With implementation and testing, it turns out that the method to apply successive filters is not a complete idea itself. It becomes problematic if filters are applied at locations only where reactions occur, since individual cells are then used instead of groupings of cells. This is best displayed by example. Consider using a mixture fraction filter to find all CFD cells where the mixture fraction gives equivalence ratios between $0.9 \leq \phi \leq 1.1$ as on the bottom of Figure 46.
The issue here is that **each** of these point aggregates in (1), (2), and (between) in Figure 46 will have an equivalence ratio between some specified values. These cells can be grouped together, however, cells between the first and second groupings ("between" in the bottom picture in Figure 46) are not being taken into account. Cells between the two groupings may have equivalence ratios of, say, 0.8, in which case reactions will occur which are not being captured. One can move the filter to have values between say, $0.5 \leq \phi \leq 1.5$, but then many cells will be marked as a group, and the differential in equivalence ratio will not be taken into consideration. This is a specific case of the fact that the amount of NOx produced between two different



(a) Reactors only sit in locations where there are actually reactions (orange squares).



(b) Reactors sit in locations where there are reactions (orange squares) and at areas where there are major flow features (green and blue are inlet flows from the dome and from the liner, respectively, and yellow is mixing).



(c) Reactors completely line the entire combustor. Reactions occur at all locations.

Figure 45: Different ideas describing where to place reactors in the flow field. The connections between the reactors are not shown, just their placement within the combustion chamber.

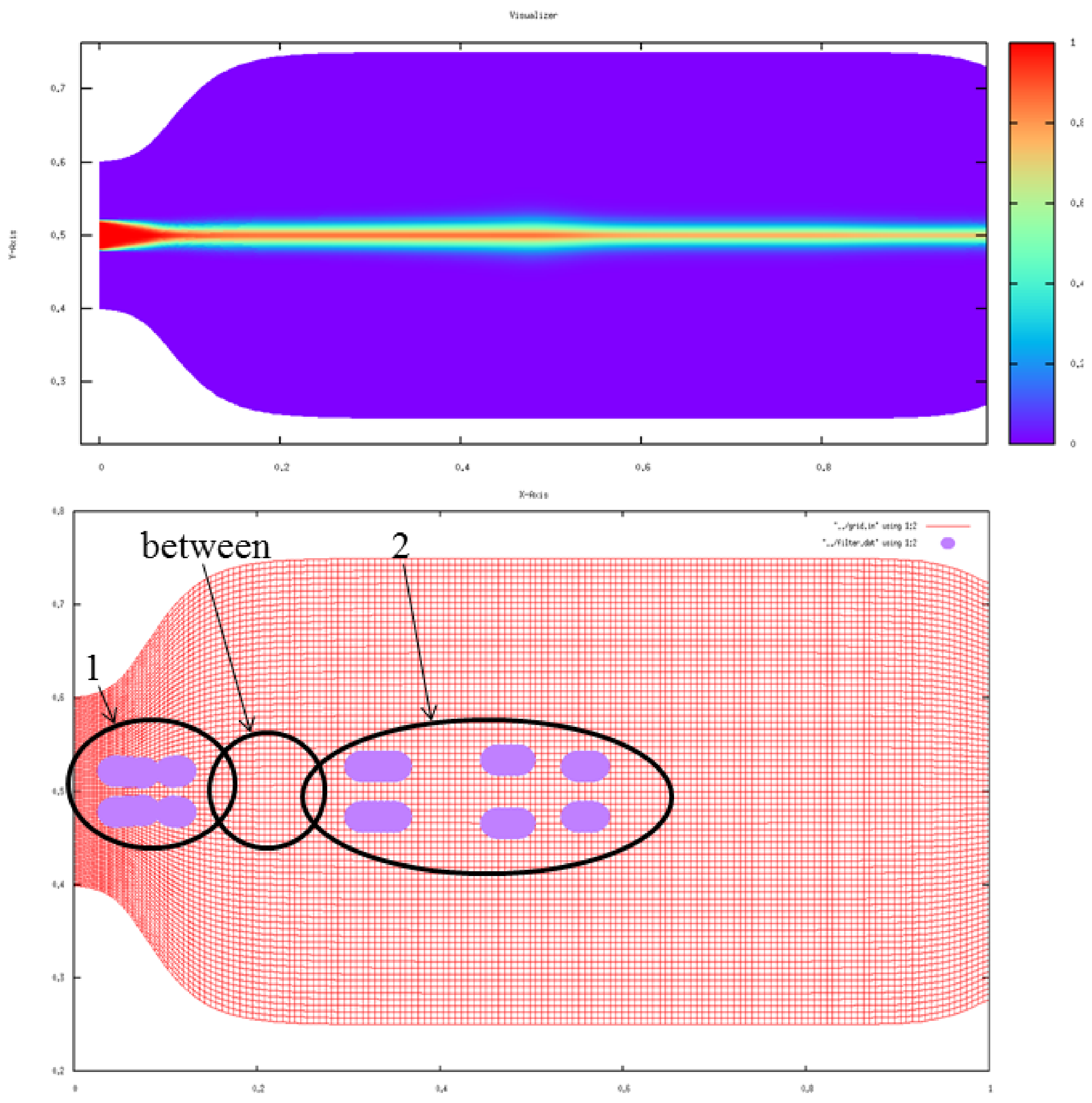


Figure 46: Putting reactors only in locations where equivalence ratio dictates reactions is misleading. The figure shows points where the equivalence ratio (Jet-A) is between $0.9 \leq \phi \leq 1.1$. Top: Mixture Fraction contour for Euler Equations solution. Bottom: Points where the mixture fraction is between the threshold. Note: The aspect ratio of the grid is warped in order to show the dots reasonably well.

combinations of reactors will be different, since oxidizer, more fuel, and products will mix in between the reactors. This is shown in Figure 47.

The previous methods discussed in Chapter 2 split the combustor into zones, also. Here however, there is an extra constraint that calibration must not be performed

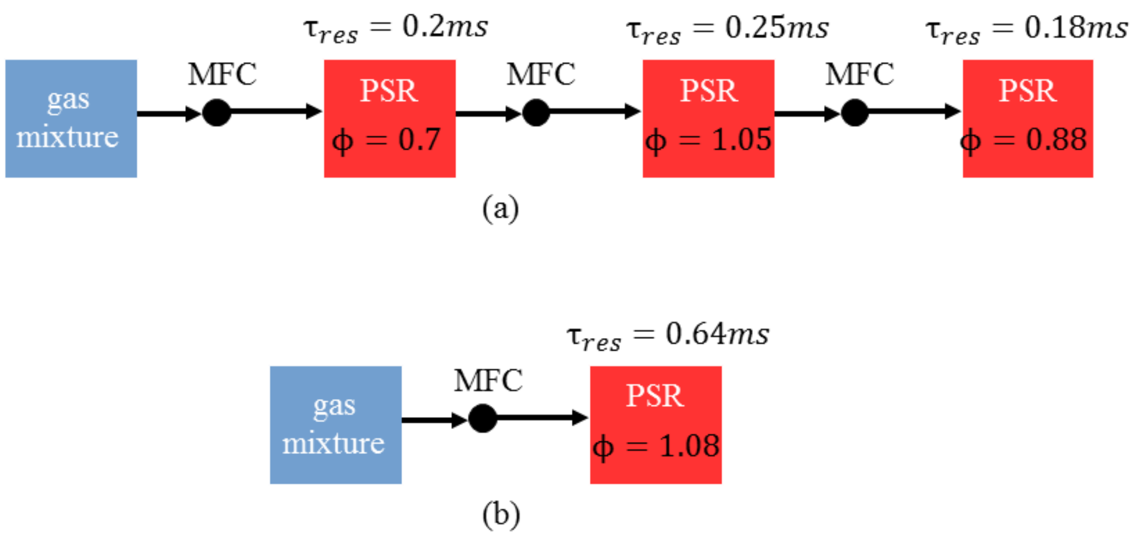


Figure 47: The amount of NOx produced between the following two cases ((a) and (b)) will be much different, since breaking down the species into different reactors will cause different reactions to proceed.

and data must not be used. In the approaches in Figure 45, calibration would be be necessary due to the fact that one has an "unstructured" kinetics mesh. For example, if reactors are only placed in regions where reactions occur, one has no direct knowledge of the flow which comes into the reactor from another reactor. Without connecting the reactors together in some structured manner with outside flow from reservoirs, the model must be calibrated to determine how much flow from each reactor goes into the others. Take Figure 45(b) as an example. Here, the model must be calibrated since flow can come from other locations and mix before hitting a given reactor.

The approach to line the entire combustor with reactors is also not a complete idea in itself, since there are some places where reactors are not necessary[4]. The final approach for the algorithm uses a combination of the above two approaches. The aim is to create a *sparse* kinetics mesh on top a CFD mesh. Moreover, some

[4]The methodology tries to avoid the limit where a reactor sits in each cell, since this is just reacting flow CFD.

of the places where reactions transpire can use **the same** reactor, as opposed to using separate reactors. This will save time, which is the whole point of the work. For instance, consider two cells next to each other. If the equivalence ratio in cell one is $\phi = 1.0045887$, and the equivalence ratio in the next cell is $\phi = 1.0045832$, there really is not much need to solve the reacting flow equations in both cells, since the answer will be approximately the same. Additionally, consider a very dense grid where thousands of cells in a certain region have very similar equivalence ratios. One can theoretically group all of these cells together into one reactor, given an accuracy threshold.

4.3.2 The MoST Algorithm

Consider a large kinetics mechanism, perhaps an accurate dodecane mechanism with 1,000 species to model Jet-A. One may choose such a kinetics model because it contains the mechanisms needed either for proper heat release, computing emissions, etc. If full reacting flow is run in 2-D using this mechanism, each cell must solve 1,003 equations: conservation of mass, conservation of momentum, conservation of energy, and 999 species equations at each grid point (closed by continuity). If running non-reacting flow only four equations are solved, conservation of mass, momentum in the x and y directions, and conservation of energy. The idea here is to group cells since high accuracy is unnecessary during preliminary design when one is running trade studies, especially when just trying to determine approximate combustor characteristics and NOx emissions.

A new approach is taken which involves combining the two aforementioned approaches: lining the entire combustor with reactors and choosing points where reactions occur. This results in the creation of a true method for "sparse kinetics". It allows one to run a dense CFD grid with a less dense kinetics grid overlaid, the point being to save time when running large mechanisms to get accurate emissions values. The dense CFD grid allows one to take specific flow features into account, such as recirculation and other flow attributes requiring many grid points to resolve, yet without directly couplying chemistry in each cell. The steps for this method include splitting the combustion chamber into KZs and "binning" the CFD cells into these zones before chemical information is obtained. The main algorithm for this method is shown in Figure 48.

4.3.2.1 Creating the CKN Architecture

Before describing each step of the algorithm in Figure 48, the method for storing the kinetics data is outlined. The KM is stored as a separate data entry consisting of numerous data structures which hold information about each cell's properties (just as you would have an array/vector/data structure which holds information about each cell in the CFD mesh). The data contained within the KZ data structure are shown in Table 4. Since the domain extends in two dimensions, a matrix of structures is used to hold information about the KG. Since this algorithm is dynamic (may accommodate any number of kinetics cells), vectors are used to store the entries adaptively, and new data structures are inserted into the matrix as necessary.

The first step in the MoST algorithm is to obtain a non-reacting CFD solution. This is described above and used as the starting point here. Once this solution is obtained, the algorithm determines the CKN architecture, which consists of getting and setting the reactor positions. This is a whole algorithm within itself, as shown in the pseudocode below.

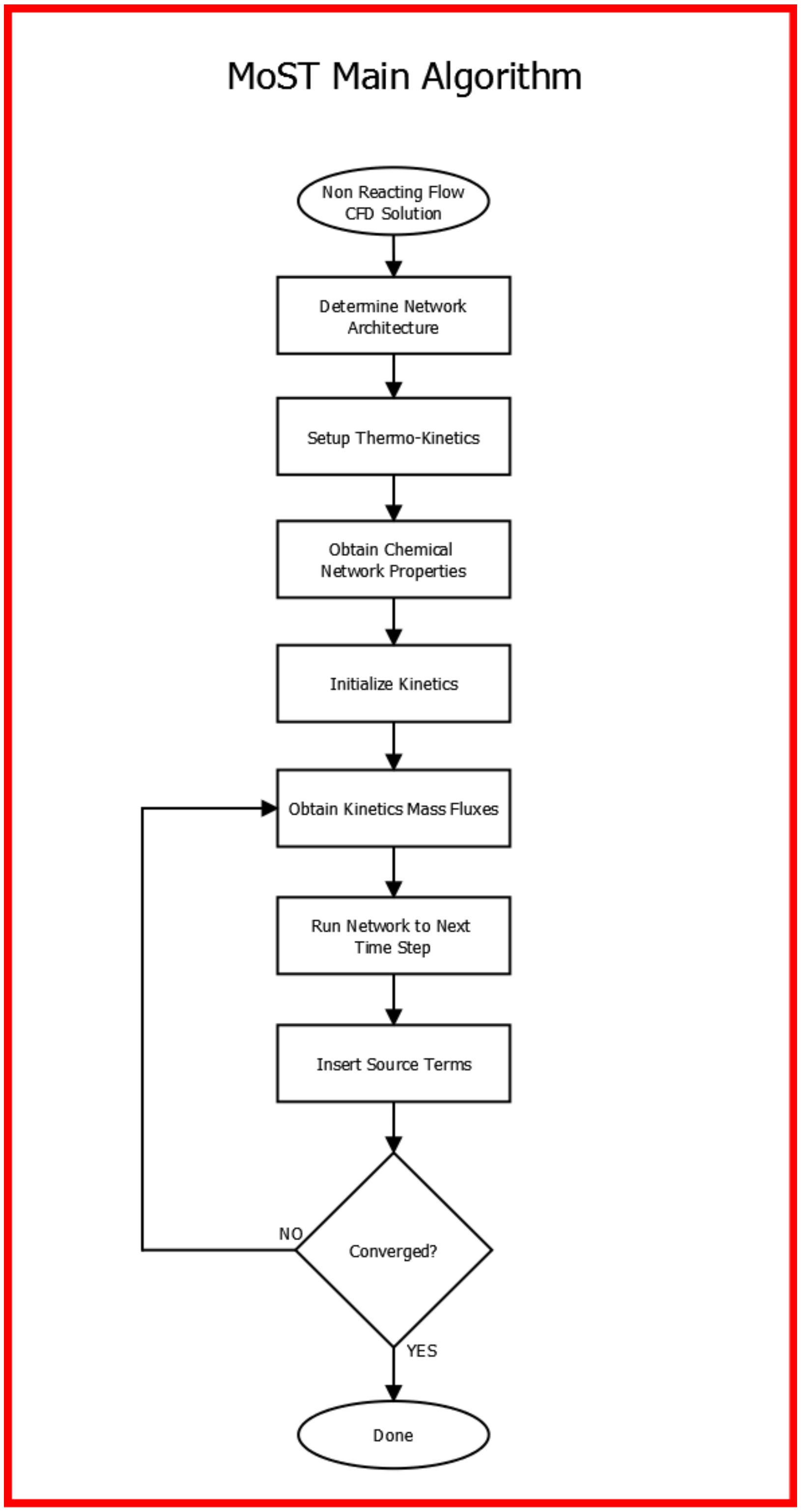

MoST Main Algorithm
Non Reacting Flow CFD Solution
Determine Network Architecture
Setup Thermo-Kinetics
Obtain Chemical Network Properties
Initialize Kinetics
Obtain Kinetics Mass Fluxes
Run Network to Next Time Step
Insert Source Terms
Converged?
NO
YES
Done

Figure 48: The MoST Algorithm. Each of these steps are sub-algorithms developed to complete a specific task.

Table 4: Information stored within each kinetics zone.

x_{Start}	Starting position of zone in x-direction
x_{End}	Ending position of zone in x-direction
y_{Start}	Starting position of zone in y-direction
y_{End}	Ending position of zone in y-direction
i_{Start}	Starting index of zone in x-direction
i_{End}	Ending index of zone in x-direction
j_{Start}	Starting index of zone in y-direction
j_{End}	Ending index of zone in y-direction
EqRatio	Average Equivalence Ratio in the zone
rho_{Avg}	Average density in the zone
T_{Avg}	Average temperature in the zone
p_{Avg}	Average pressure in the zone
numPts	Number of points in the zone
MFR_W	Mass flow rate coming in / leaving from the west side of the zone
MFR_N	Mass flow rate coming in / leaving from the north side of the zone
MFR_E	Mass flow rate coming in / leaving from the east side of the zone
MFR_S	Mass flow rate coming in / leaving from the south side of the zone

```
/* Converged NRCFD Solution is obtained */

/* Determining reactor positions / locations */
initialZoneSplit();

while(mixtureFracGradients) {
  splitKineticsZones();
}

getAverageZonalProps();
```

The first step is to come up with an initial zone split. These splits should be at locations where gradients are high and reactors are likely to be located. If the equivalence ratio is going to greatly vary from one point to another, it can be due to the presence of dilution jets with fresh air coming in. When the initial zone split function is called, it does its first split in the y-direction by looking for fuel jets. It splits the combustion chamber down the middle of each fuel jet (i.e. one fuel jet gets split among two zones).

The next split is done in the x-direction, where the algorithm looks for liner air (ex. dilution air) entering the combustor, and splits the combustor at the beginning of each dilution jet in the x-direction. In the example here, there are three jets on the top and three on the bottom (symmetric) so that it starts with four zones on the top and bottom, splitting the grid into kinetics zones at these locations. Figure 49 shows the initial splitting of the zones with a single fuel jet down the center.

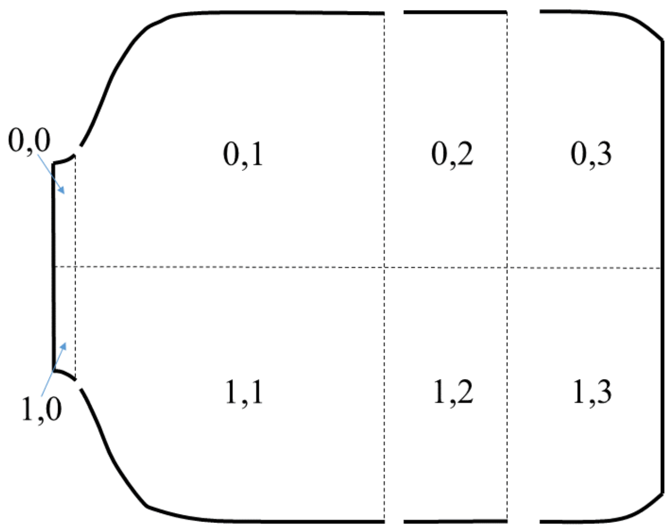


Figure 49: Initial splitting of the zones at the start of the algorithm (initialZoneSplit() function). After a converged CFD solution is obtained, the initial split is based on the number of fuel jets and dilution jets. Here, there is one fuel jet and three dilution air jets, so there is an initial split into four zones in the x-direction and two in the y-direction. The fuel jet is located in the center, while the centers of the dilution jets are located at $x = 0.05$, $x = 0.5$, and $x = 0.8$

After the initialZoneSplit() is complete, a while loop is entered and the existing zones continually split until there is no more mixture fraction gradient anywhere within the combustor (to within the specified threshold). Here, the code iteratively calls the same function as the flow field is searched for gradients (with the most important gradient being the the mixture fraction, since this defines how different one cell is from another in terms of the fuel/air fraction). The code stops iterating on the splitting function when no gradient is present that is larger than the specified threshold. If the mixture fraction differs too much from one kinetics zone to another (Figure 50), the zone is then split again, which is due to the fact that cells must be grouped with similar fuel/air ratios. This is where the method's utility comes in- it will give as accurate a result as possible while saving time due to the fact that cells with similar mixture fractions are grouped into KZs.

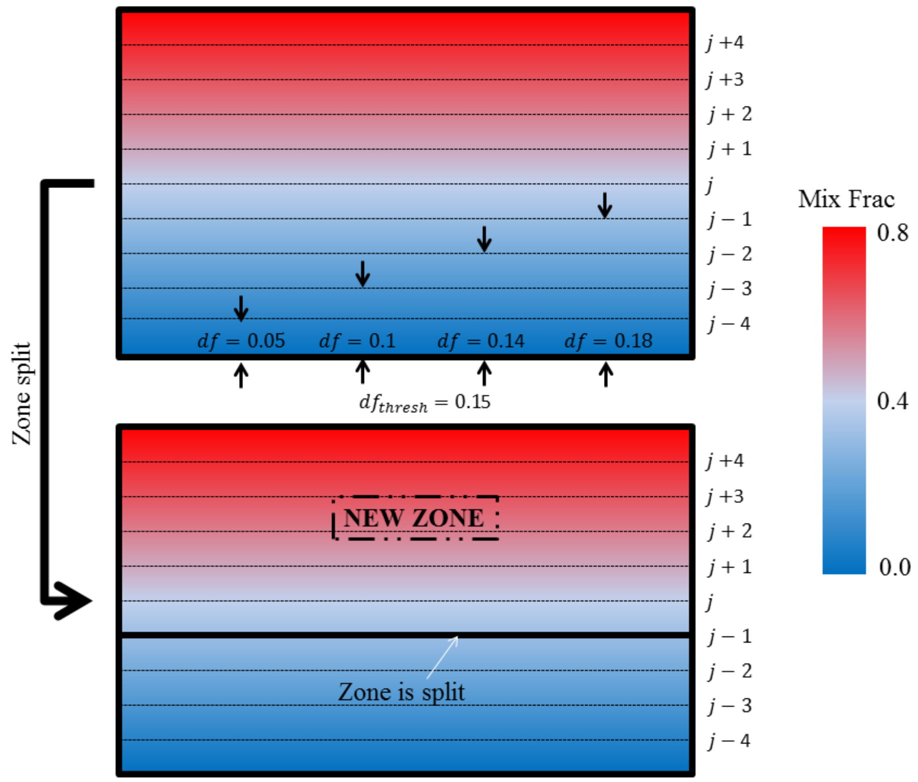


Figure 50: Zoom in of a square field showing mixture fraction contours. This shows an example search through a notional Cartesian mixture fraction field for the gradients. The searching is done in the y-direction using finite differences. The threshold is set to mixture fraction gradients of 0.15, and a new zone is created at $y = j - 1$ when this mixture fraction gradient is encountered or exceeded.

As shown in Figure 50, the gradient search starts from the beginning of each cell (bottom left). The algorithm successively checks each CFD grid point within each KZ until the gradient threshold is reached or exceeded. If the threshold is exceeded, a new KZ is created at that location. This is shown in the y-direction in Figure 50, and occurs the same way for the x-direction. The order in which the checks are done are the positive x-direction (left to right), the negative x-direction (right to left), the positive y-direction (bottom to top), and then the negative y-direction (top to bottom). Note that this check are carried out twice in the same direction (starting at different locations) in order to unbias the directionality of the mixture fraction gradient. An example of this splitting within a combustor in the x-direction is shown in Figure 51. Assuming this is the only gradient in the x-direction, the y-direction is then searched, and rezoning occurs in the same way. This is shown in Figure 52, and is done for all kinetics zones. These new zones are created by temporarily storing the old data structure, changing variables within the vectors, and inserting the zone in the correct location within the matrix where the structures are stored. Programatically, structures representing the new kinetics cells are dynamically allocated. These data are filled in as the zones are created by inserting these structures into the already existing matrix of structures.

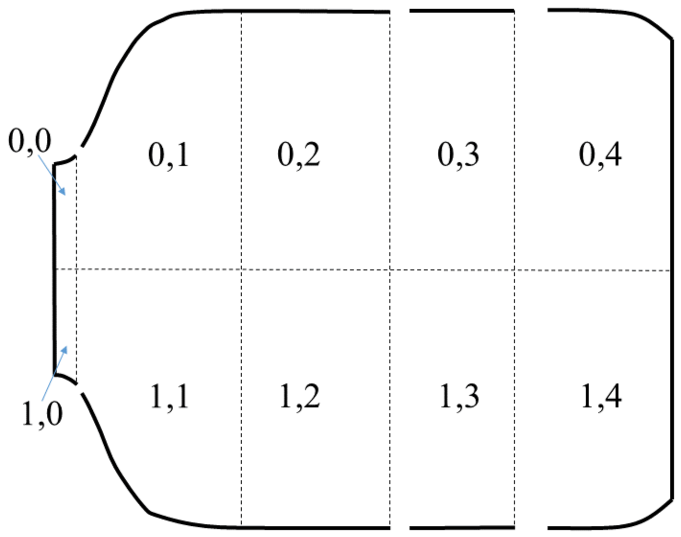


Figure 51: After the initial split is done, gradients in mixture fraction are found. If the gradient between kinetics zones becomes too large, this zone is split. This shows the split in the x-direction after gradients in Figure 49 are searched.

The overall algorithm for further splitting the zones can be done different ways. The approach taken in this work is to split and create new zones, and then recheck

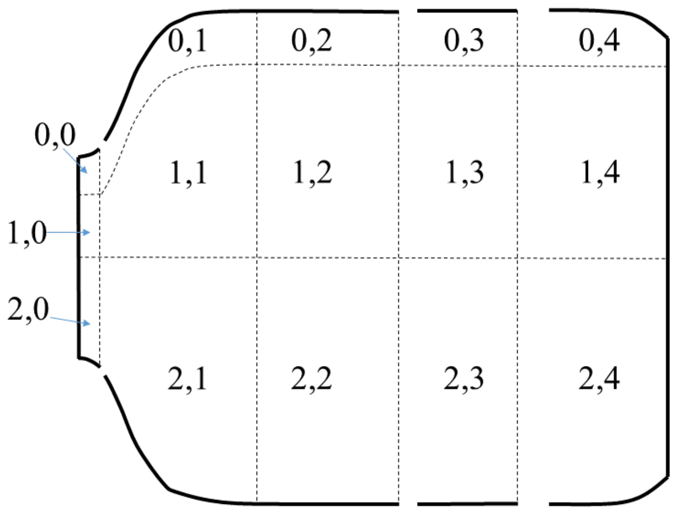


Figure 52: After the initial split is done, gradients in mixture fraction are found. If the gradient between kinetics zones becomes too large, this zone is split. This shows the split in the y-direction after gradients are searched in the x-direction in Figure 51.

all of the split zones after each bifurcation is completed. For example, if searching through zone "0,2" in the x-direction in Figure 51, the gradient search starts at the dotted line to the right of "0,1" and to the left of "0,2" and proceeds to the dotted line to the right of "0,2" and to the left of "0,3". The pseudocode for this is shown below.

```
/* Splitting the kinetics zones */

bool splitKineticsZones() {
  if (positiveX) {
    for (all vertical zones) {
      for (all horizontal zones) {
        for (sweep zone +y direction ) {
          for (sweep zone +x direction) {
            df_dx = mixFrac_{i+1} - mixFrac{i}
            if (df_dx > threshold) {
              checkIfPointTagged();
              if ( pointTagged() ) {
                found = true;
                break;
              }
              else {
                rezone();
                tagPoint();
                return true;
              }
            }
          }
        }
      }
    }
    positiveX = false; //If here, finished all sweeps in +x
  }

  if (negativeX) {
    /* Repeat above, sweeping in the -x direction */
  }

  if (positiveY) {
    /* Repeat above, sweeping in the +y direction */
  }

  if (negativeY) {
    /* Repeat above, sweeping in the -y direction */
  }

  return false; //Exit rezone loop-- no more gradients!
}
```

If this mixture fraction is too large (above the threshold), a rezone occurs. The original zone "0,2" becomes smaller, and the new zone, "0,3" is inserted on the right. Then, the old zone "0,3" becomes "0,4", etc. This rezone function moves the numbers around until all of the zone values line up. Importantly, it keeps it in order, so that the kinetics zones stay next to each other and the code knows which zones pass flux information during iteration. After this new zone is created, gradient searches recommence (at the beginning of the new zone), and gradient checks continue. It is important to note that the gradient threshold can be set to whatever the user desires. As stated above, the inspection direction does not proceed unless every zone no longer has a gradient present. Once this is the case, the rezone() functions stops executing, and the KM for the present CKN iteration has been found, with the starting, ending, etc. locations of each mesh point stored.

The approach taken here results in a structured KM since each time a new KM cell is added, the entire grid shifts (Figures 51 and 52). This algorithm can be used to create an unstructured KM for use with unstructured CFD meshes simply by relaxing the constraint that all cells must shift when a new zone is created. However, having a structured KM is more simple for the same reasons why having a structured CFD mesh is simple: there is no need to store connectivity since that information is known based on array indexes. At this point, everything is known about the CKN, so the next step is its construction.

In general, the Cantera routines desire knowledge of number of reactors, how the reactors are connected, which gas each reactor uses, etc. at the time of program commencement. However, this information is unknown until the first converged CFD solution is obtained and the above filtering algorithm is executed. Therefore, this information is created on the fly using dynamic memory allocation and occurs during program execution. Each reactor, gas, mass flow controller, valve, etc. are declared as empty containers with nothing placed in them until after the CFD run and filtering algorithm executes. Specifically, each of these containers are declared as vectors of vectors (in C++, effectively forming a matrix) so that they are populated by pointing to their memory addresses at a later time. Additionally, the user has the option to write out kinetics data as frequently as each solver step[5], so vectors of output file pointers are also created. The overall gist of the setup algorithm is shown below.

```
/* Number of reactors for each network comes from CFD code */

/* "*" denotes that these are pointers to memory locations */
for (i=0; i<numReactorNets; i++) {
  for (j=0; j<numReactorsInNet; j++) {
    *allocate_gas_{i,j};
    *allocate_PSR_{i,j};
    *allocate_valve_{i,j};
    *allocate_massFlowController_{i,j};
    *allocate_initGasStrings_{i,j};
    *allocate_initProps_{i,j};   //T, P, etc.
    *allocate_ReactorVols_{i,j};
    if (viewKinetics) {
      *allocate_outStreamFile_{i,j}
    }
  }
  *allocate_inletReservior_{i};
  *allocate_feedGasString_{i};
}

for (i=0; i<numReactorNets; i++) {
  for (j=0; j<numReactorsInNet; j++) {
    createReactorConnections_{i,j}
  }
}

for (i=0; i<numReactorNets; i++) {
  for (j=0; j<numReactorsInNet; j++) {
    initializeReactors_{i,j}
  }
}
```

At this point (after the filtering algorithm has been run), the size of these vectors are allotted by using the KM information. Specifically, allocations are carried out for the number of reactors necessary (each KZ is one reactor), the types of connectors,

[5]Writing out data for each reactor as each solver step causes a severe increase in runtime

valves, mass flow controllers, etc. Afterwords, these Cantera objects exist in memory, however they must be arranged and connected in order to simulate the CFD flow field. Each reactor is connected by two valves and any number of mass flow controllers depending on where and how the reactor is situated. The two valves always get connected in both directions between reactors, however the mass flow controller direction depends on the the bulk flow orientation between the two reactors. In order to get this information, integration is performed along all boundaries as in Figure 53 (a).

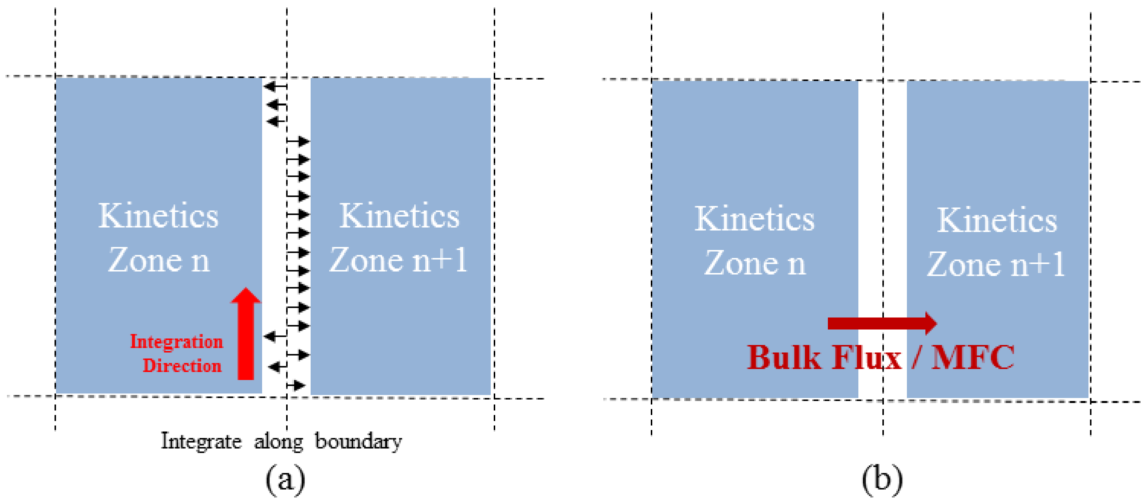


Figure 53: Example integration performed along a boundary to find the orientation of the mass flow controller. (a) In order to configure the CKN, the direction of each mass flow controller must be determined by integration between each reactor. (b) The bulk flow direction gives the orientation for the mass flow controller.

The completion of this step results in a CKN which emulates the CFD flow field. This is shown in Figure 54.

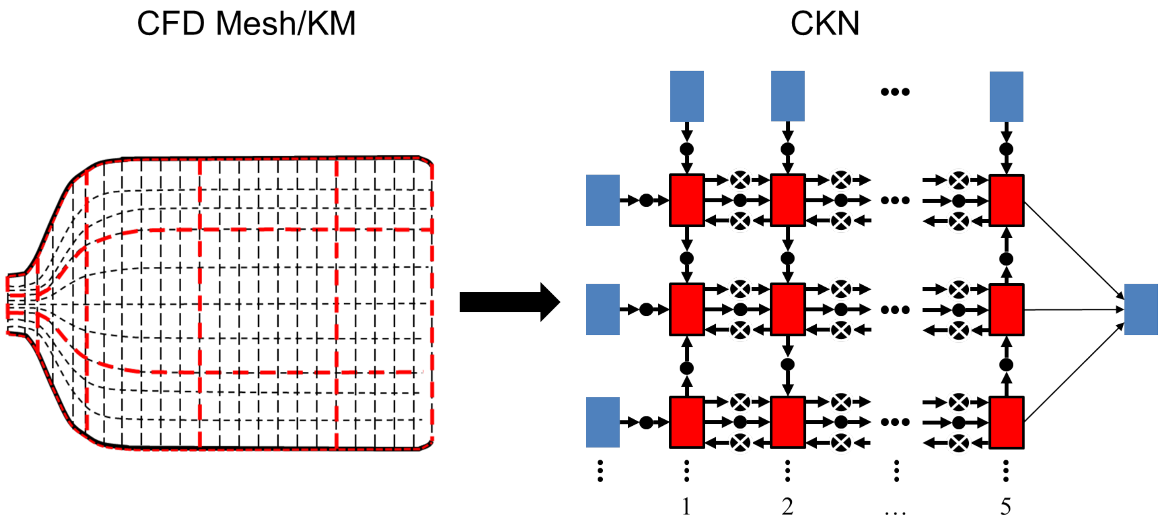


Figure 54: Notional depiction of the KM overlaid on top of the CFD mesh after the filtering algorithm is run. (Left) The black lines are the CFD mesh and the red lines are the KM. (Right) The orientation of the mass flow controllers are now known, so the CKN is constructed.

The last step in determining the CKN architecture involves computing the average properties for each zone in the KM. This is done in a few steps, starting with the algorithm shown below.

```
/* Total zone values */
for (i=iStart; i<=zone[][].iend; i++) {
  for (j=jStart; j<=zone[][].jend; j++) {
    zone[][].totalMass += zoneVolume[][] * rho[][];
    zone[][].fuelMass  += totalMass * mixFrac[][];
    zone[][].oxMass    += totalMass * (1.0 - mixFrac[][]);
    zone[][].rhoAvg    += rho[][];
    zone[][].pAvg      += p[][];
    zone[][].TAvg      += T[][];
    numPointsInZone++;
  }
}
z[][].mixFrac       = zone[][].fuelMass / zone[][].oxMass;
zone[][].rhoAvg   /= numPointsInZone;
zone[][].pAvg     /= numPointsInZone;
zone[][].TAvg     /= numPointsInZone;

/* Searching inlet fuel/air ratio */
for (k=0; k<z.size(); k++) {
  fuelMass  = 0.0;
  oxMass    = 0.0;
  totalMass = 0.0;
  for (j=z[k][0].jStart; j<z[k][0].jEnd; j++) {
    totalMassCell = cellVol[0][j] * cellRho[0][j];
    totalMass    += totalMass;
    fuelMass     += totalMass * mixFrac[0][j]);
    oxMass       += totalMass * (1.0 - mixFrac[0][j]);
  }
  z[k][0].fInlet = fuelMass / totalMass;
}
```

The most important part of this algorithm is to determine the mass flow of fuel and air. The fuel/air ratios of the reactors do not matter at program start, only their actual inflow rate, and the flow rates between reactors. As long as the reactor network is run for longer than a few residence times of the reactors, anything initially inside will filter out, leaving only the correct species at the end. Therefore, stoichiometic mixtures at high temperatures are inserted initially so that reactions are ensured to begin. To correctly capture the inflow of fuel and oxidizer, the inflow reactors are searched as shown at the bottom of the algorithm above. Each reactor may have fuel, oxidizer, or a combination of the two. Dilution air from the liner only consists of oxidizer and therefore its composition does not need to be checked. Lastly, approximations are made for the pressure. Since combustors operate in the low Mach number deflagration regime, the pressure is approximated as the inflow pressure into the combustor [105]. This approximation is often made, since doing so drastically reduces the runtime while solving the coupled ODEs of the system.

To display the utility of the algorithm, various thresholds for the mixture fraction gradients are used to create kinetics zones. First, a gradient of very close to 1 ($\partial f \approx 1$) is used so that the mixture fraction has to change enormously before a new kinetics zone is created. This is as sparse of a kinetics grid as can be found. Gradients all the way down to ($\partial f = 0.01$) are tested for both the Euler and NS equations. For the NS equations, an artificial diffusivity is used in order to get the fuel to spread out for testing purposes[6]. The reason for this is to check that the algorithm works in general for any fuel/air mixture, no matter the diffusivity. The mixture fraction contours for the actual CFD cases are shown for the Euler and NS equations in Figures 55 and 56. The contours in Figures 57(a) through 57(f) and 58(a) through 58(f) show the varying KGs for the Euler and NS equations, respectively.

[6]The Euler equations represent the case when the diffusivity goes to zero.

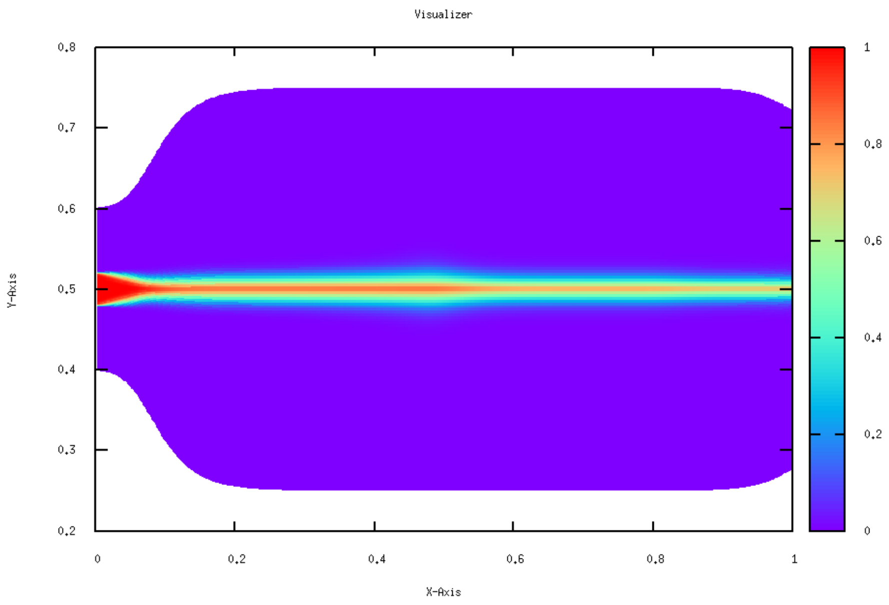


Figure 55: Converged solution of mixture fraction (Y_{fuel}) on a grid with 166x75 nodes for the Euler equations. Each CFD grid point has a computed mixture fraction, so the reacting flow equations would be run on all 12,210 cells.

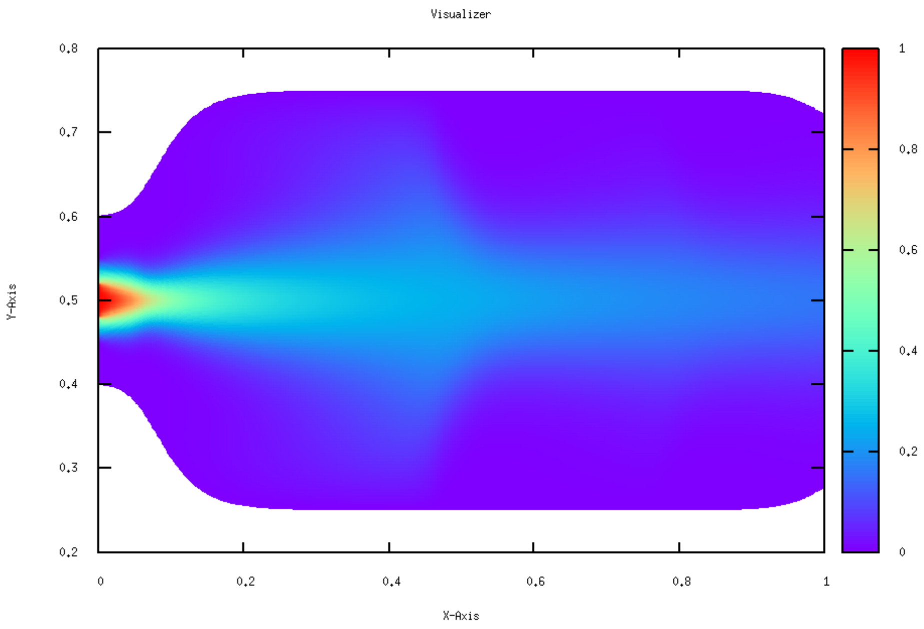
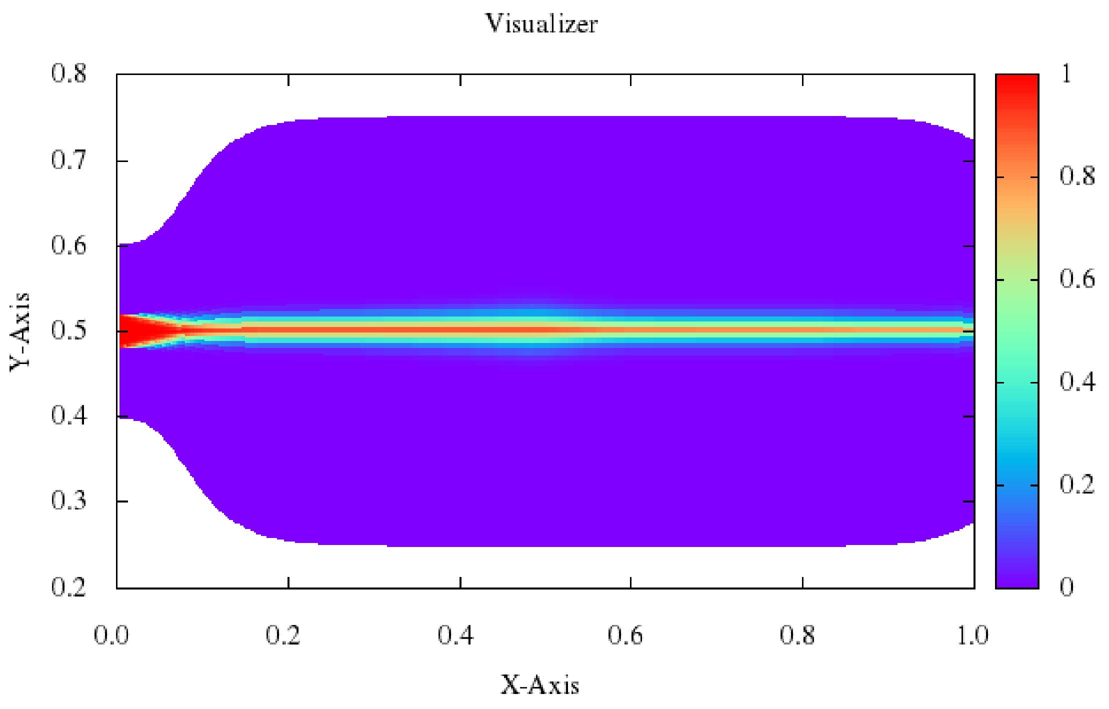
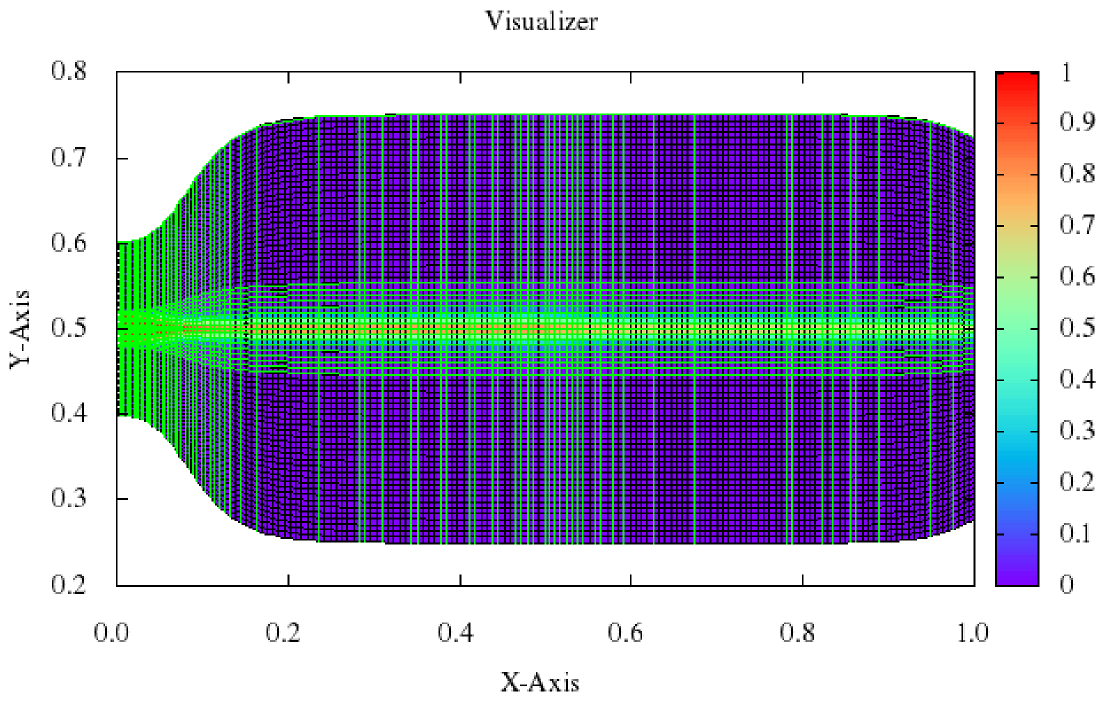


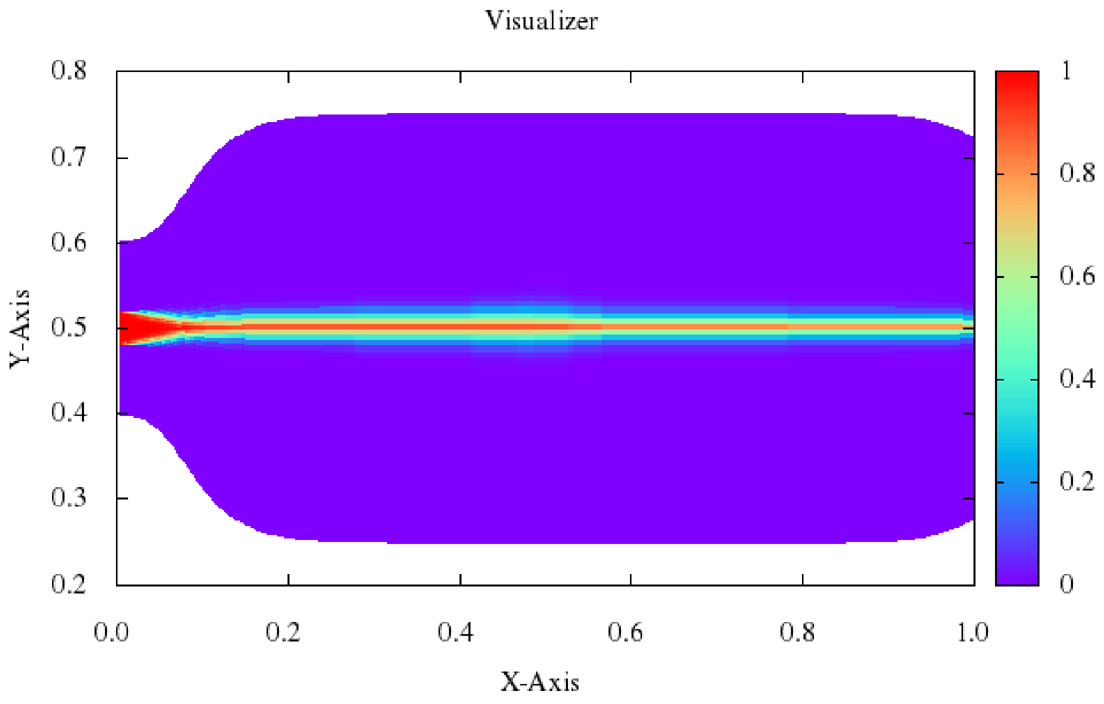
Figure 56: Converged solution of mixture fraction (Y_{fuel}) on a grid with 166x75 nodes (solving the Navier-Stokes equations with artificial viscosity, in order to get large diffusivities). Each CFD grid point has a mixture fraction, so the reacting flow equations would be run on all $12,210$ cells.



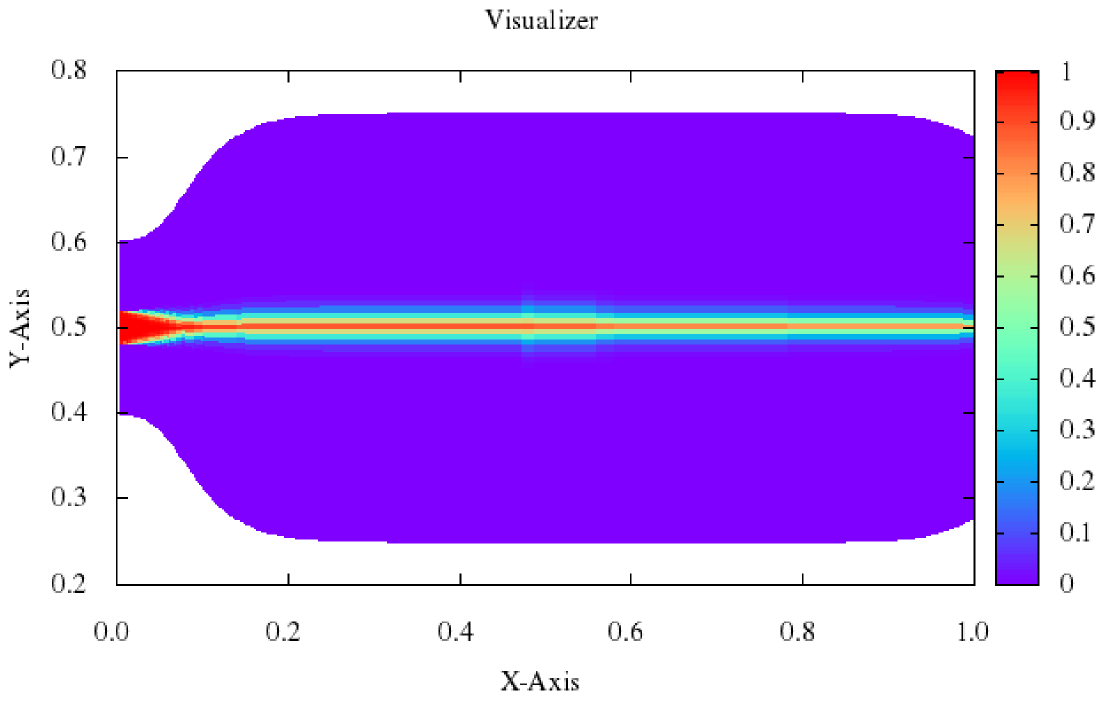
(a) Using a mixture fraction gradient of $\partial f = 0.01$, which creates 1,386 reactors.



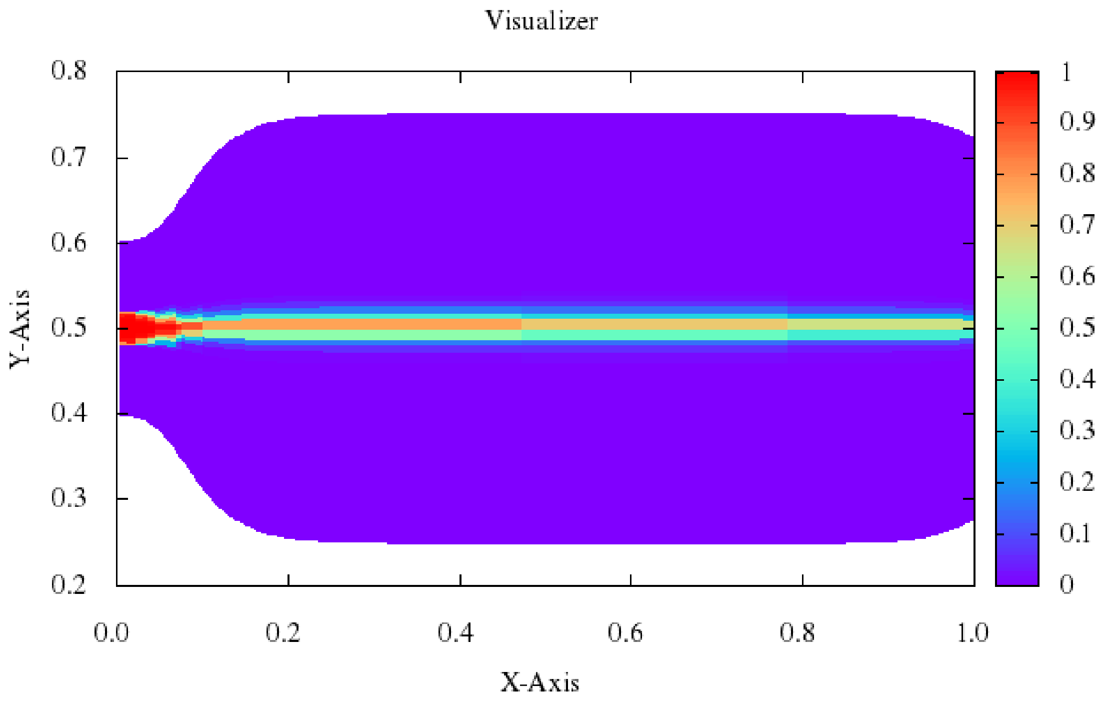
(b) Lines showing the split between zones for a mixture fraction gradient of $\partial f = 0.01$. Since small gradients cause the cells to look like the result from reacting flow, one can see here that there is an actual split between zones.



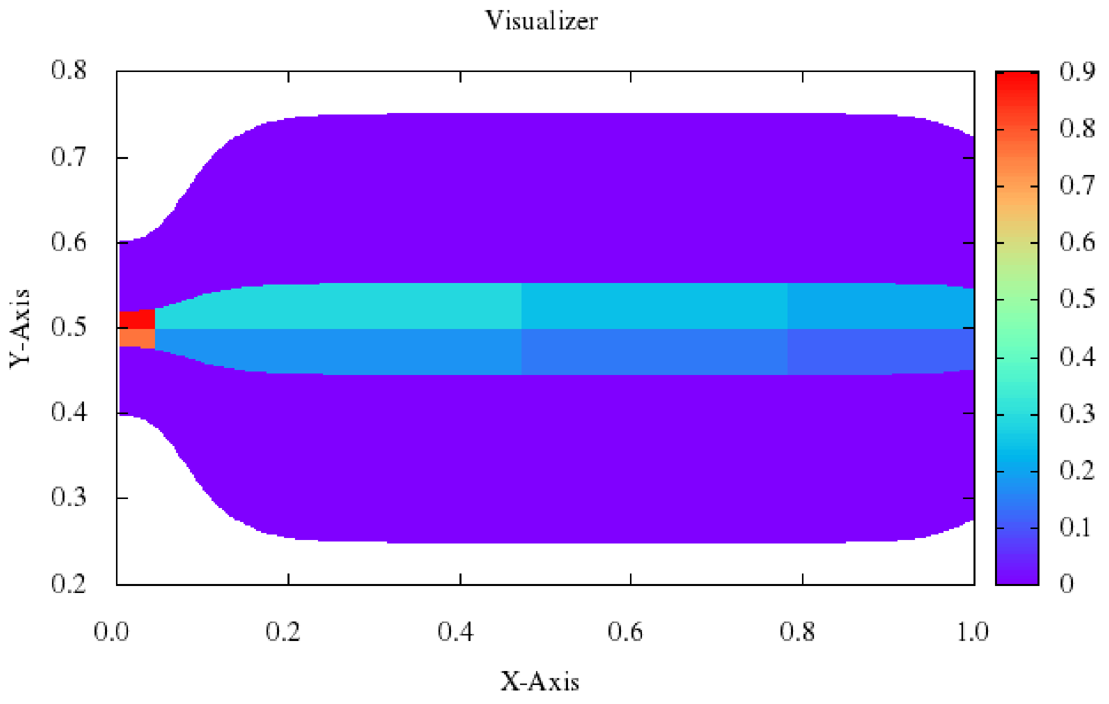
(c) Using a mixture fraction gradient of $\partial f = 0.05$, which creates 810 reactors.



(d) Using a mixture fraction gradient of $\partial f = 0.1$, which creates 396 reactors.

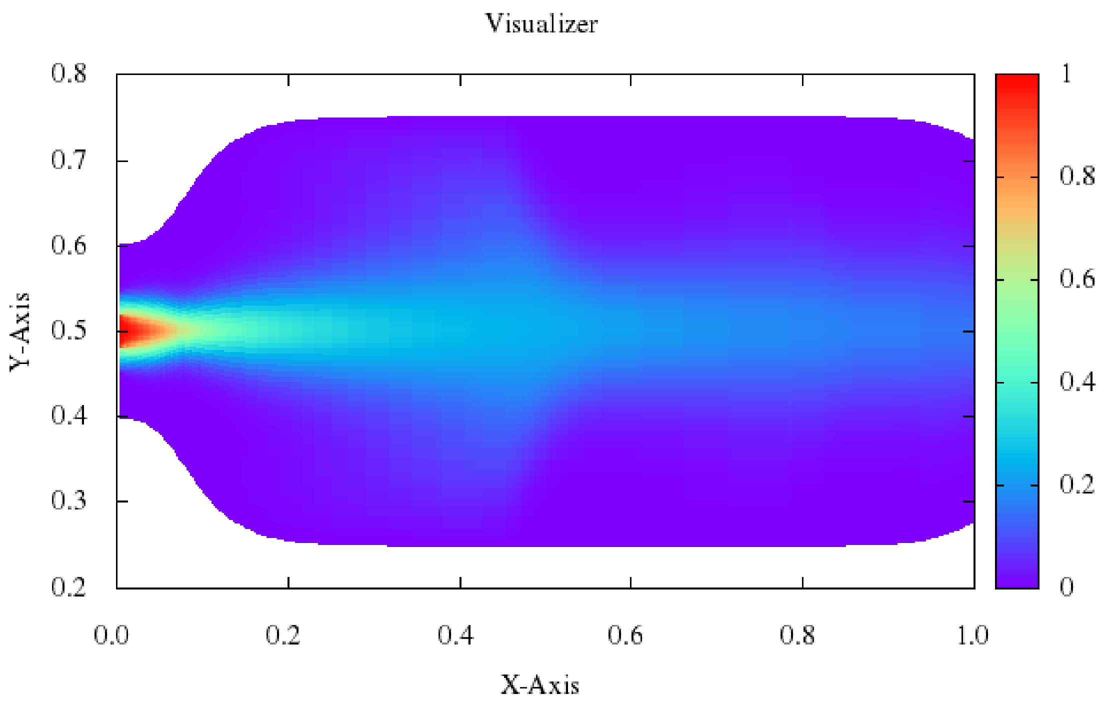


(e) Using a mixture fraction gradient of $\partial f = 0.5$, which creates 135 reactors.

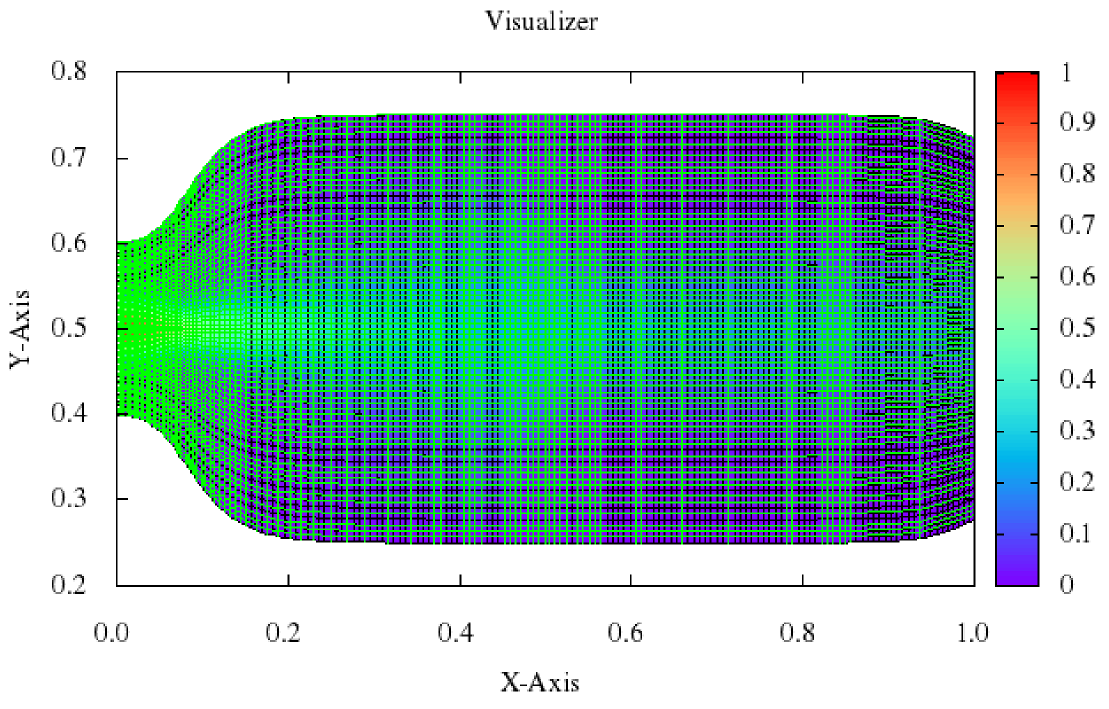


(f) Using a mixture fraction gradient of $\partial f = 0.999$, which creates 16 reactors.

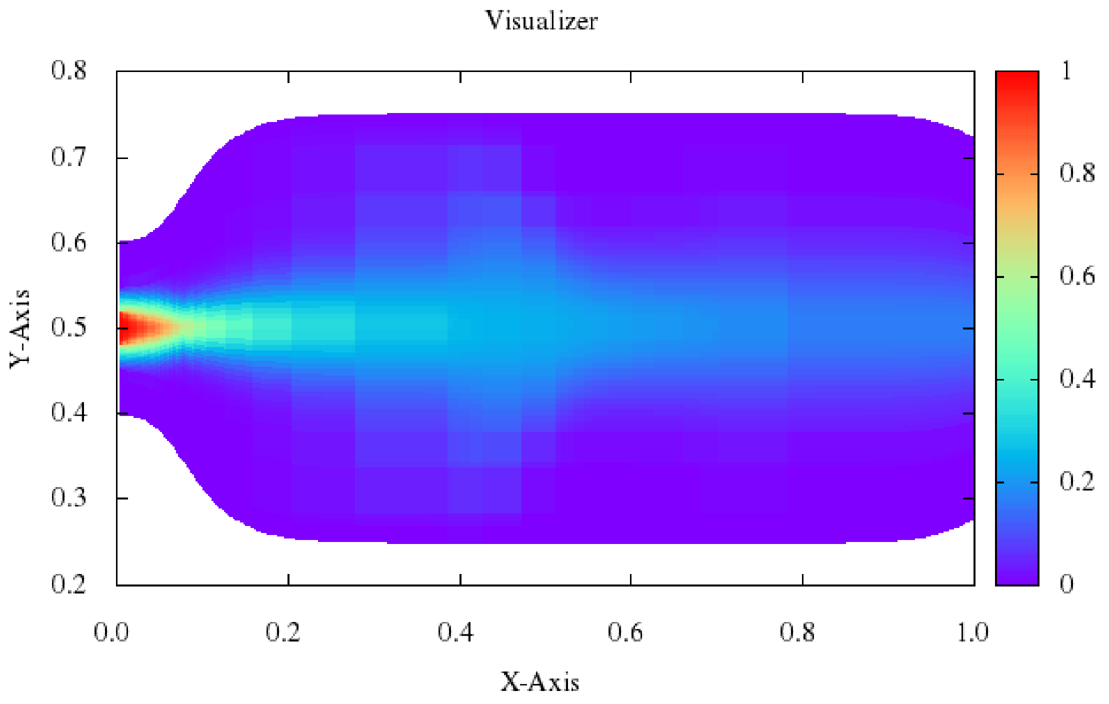
Figure 57: Converged solution of mixture fraction (f, or Y_{fuel}) using Euler equations on a grid with 166x75 nodes. Each CFD grid point has a mixture fraction which has been binned using the mixture fraction derivative. The flow equations need be solved in each cell (12,210 cells), but cells are grouped together and kinetics is only done in certain places.



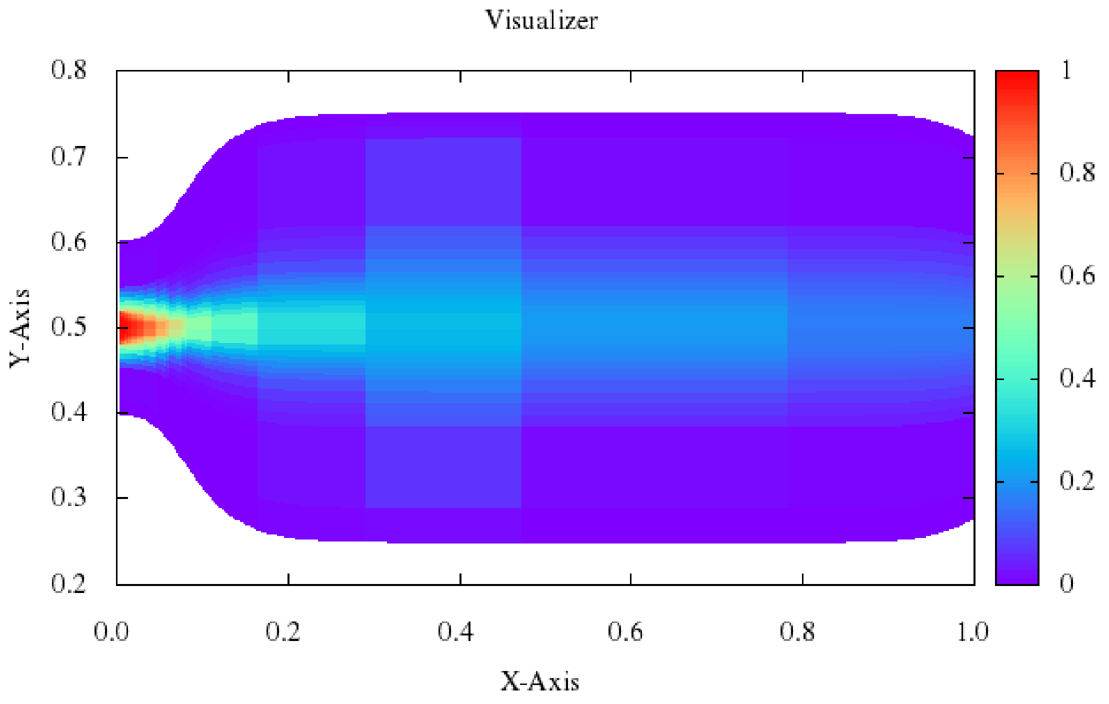
(a) Using a mixture fraction gradient of $\partial f = 0.01$, which creates 5,590 reactors.



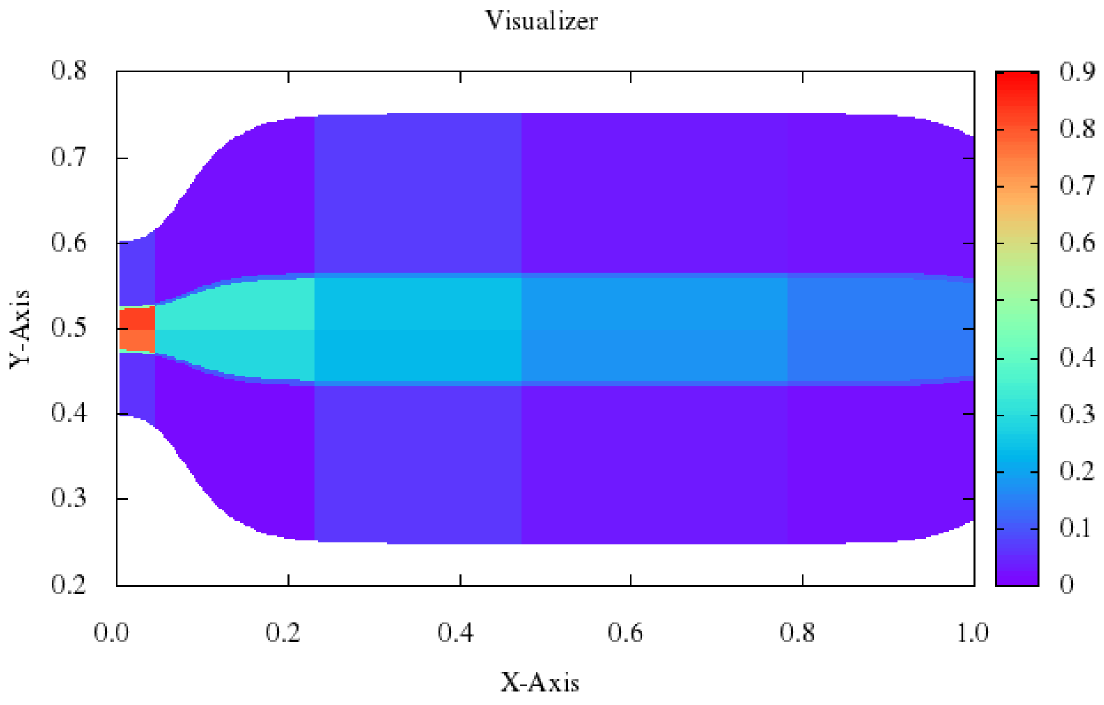
(b) Lines showing the split between zones for the fraction gradient of $\partial f = 0.01$. Since small gradient causes the cells to look like the result from reacting flow, one can see here that there is an actual split between zones.



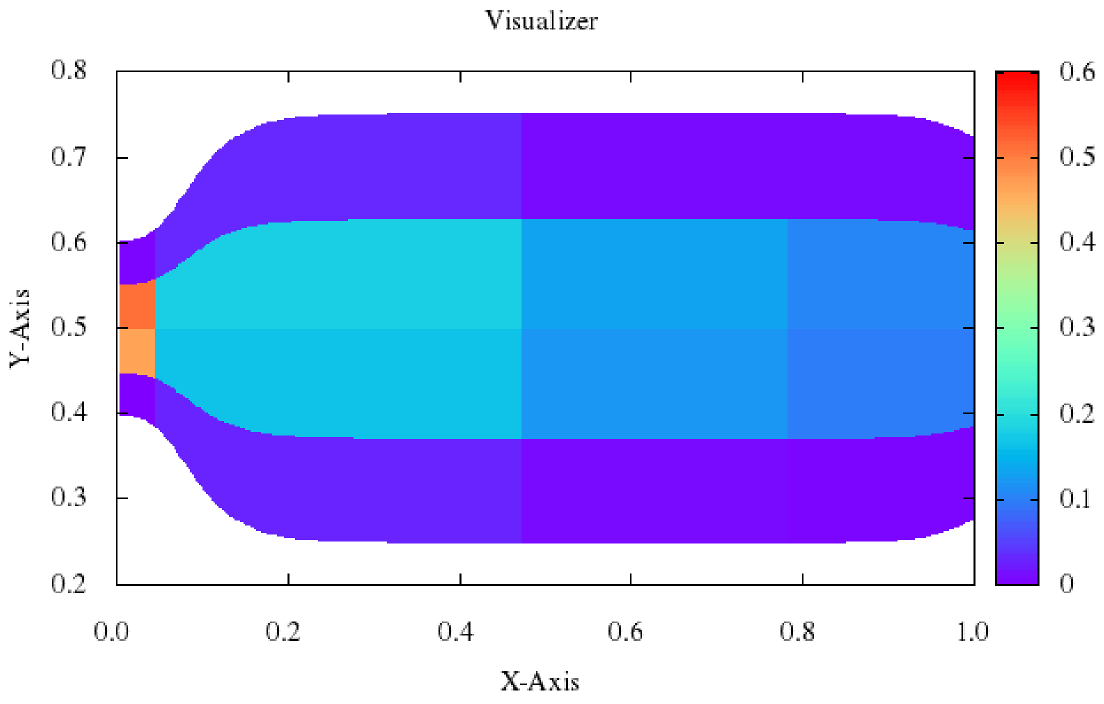
(c) Using a mixture fraction gradient of $\partial f = 0.05$, which creates 2,064 reactors.



(d) Using a mixture fraction gradient of $\partial f = 0.1$, which creates 570 reactors.



(e) Using a mixture fraction gradient of $\partial f = 0.5$, which creates 30 reactors.



(f) Using a mixture fraction gradient of $\partial f = 0.99$, which creates 16 reactors.

Figure 58: Converged solution of mixture fraction (f, or Y_{fuel}) using NS equations on a grid with 166x75 nodes. Each CFD grid point has a mixture fraction which has been binned using the mixture fraction derivative. The flow equations need be solved in each cell (12,210 cells), but cells are grouped together and kinetics is only done in certain places.

As a summary, the Euler and Navier-Stokes equations are run on the same grid ($12,210$ cells). Table 5 shows how many CFD cells must solve reacting flow equations as opposed to how many kinetics cells must solve them.

Table 5: Results for the savings between full reacting flow Euler and Navier-Stokes runs and the MoST algorithm at different mixture fraction thresholds on a $12,210$ cell grid. ∂f denotes the threshold for the change in mixture fraction that one is willing to accept (which will effect the accuracy), and n_f is the number of reacting cells.

Run	∂f	n_f (Reacting CFD)	n_f (MoST)
Euler	0.01	12,210	1,386
Euler	0.05	12,210	810
Euler	0.1	12,210	396
Euler	0.5	12,210	135
Euler	0.9999	12,210	16
(Forced) Navier-Stokes	0.01	12,210	5,590
(Forced) Navier-Stokes	0.05	12,210	2,064
(Forced) Navier-Stokes	0.1	12,210	570
(Forced) Navier-Stokes	0.5	12,210	30
(Forced) Navier-Stokes	0.9999	12,210	16

In reality, one may exploit an algorithm which does not allow reactions if the mixture fraction in a given cell is too high or low. This can be taken into consideration using the MoST, and may influence the number of reactors. This is especially true for the Euler equations where the fuel and air do not diffuse into each other, since the flow is completely advection driven. These parameters exist as variables in the code, and one may perform studies to see how much savings occurs before reacting flow CFD is run using either conventional methods, or the MoST algorithm.

4.3.2.2 Connecting the NRCFD and Kinetics

At this point in the simulation, the CKN architecture is known, but its properties are not. The next step involves finding and setting these properties, and then tying together the flow field and chemical network for iteration. The mass flow rates between the zones are found by integration (just like finding the directionality of the mass flow controllers) along the boundaries of each of the kinetics zones within the CFD grid. Integration occurs along the left, right, lower, and upper faces of each zone to find the mass flow rate along its edges. This is shown in Figure 59.

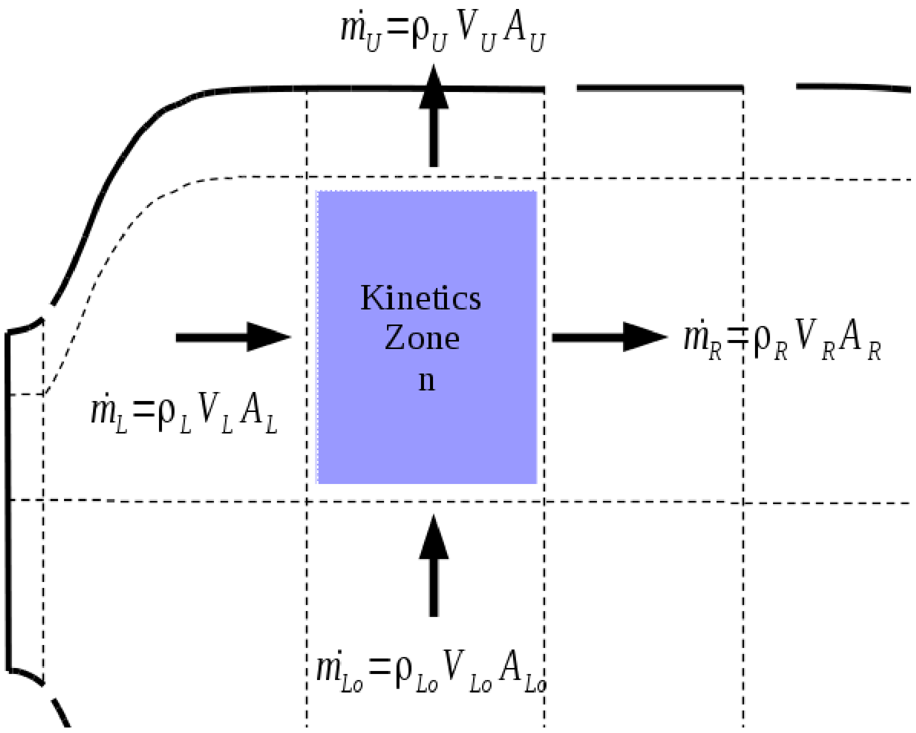

$\dot{m}_U = \rho_U V_U A_U$
Kinetics Zone n
$\dot{m}_R = \rho_R V_R A_R$
$\dot{m}_L = \rho_L V_L A_L$
$\dot{m}_{Lo} = \rho_{Lo} V_{Lo} A_{Lo}$

Figure 59: Finding the mass flow rate into and out of each cell by integrating the converged CFD solution along the KM boundaries.

Each of the CFD grid points used for integration have a density and velocity vector, in addition to areas on each face. The standard equation

$$\dot{m} = \rho V A \tag{49}$$

is utilized where the areas, A are found by exploiting the grid Jacobian, or computed strictly using geometry. Note that if there is one kinetics cell per grid cell then integration is unecessary since the fluxes into and out of each CFD cell are simply used. Typically, there will be many more CFD cells than kinetics cells, which is why this method is useful.

The kinetics grid is set to its initial properties (pressure, tempearture, kinetics cell volume, etc.) now that they are known. Mass flow integration is also complete, so the mass flow controllers are initialized to the inagural non-reacting mass flow rates between KZs. Here, the need for calibration has been eliminated by the use of numerous cells which directly connect to each other. In the previous approaches of this section, the mass flow rates between reactors are unknown since the reactors do not directly connect to one another, making flow features such as recirculation difficult to capture. In general, recirculation is an *extremely* important part of combustor operation and a feature which must be captured (a combustor is no good if it can not hold a flame since the ignitor should not continuously fire). In this approach, mass flow is simply advected between zones, so the flow proceeds in directions where the flow field mandates. This allows recirculation to occur without the code "forcing" it through a separate recirculation reactor, and correctly takes the flow field into account based on the CFD code. With a sparse KM, the approximation gets better as the fidelity increases, just as it does with an actual CFD grid.

Based on the fact that the CFD grid and KM are structured, the Cantera code produces a reaction network (displayed in Figure 38) which is too general. A much simpler CKN assembly is used since the flow field computed from the CFD solution takes care of capturing the flow features and each reactor only needs to communicate with other reactors or environments around it in an organized fashion. In order to do this, mass flow controllers are utilized between each and every reactor, as in Figure 60.

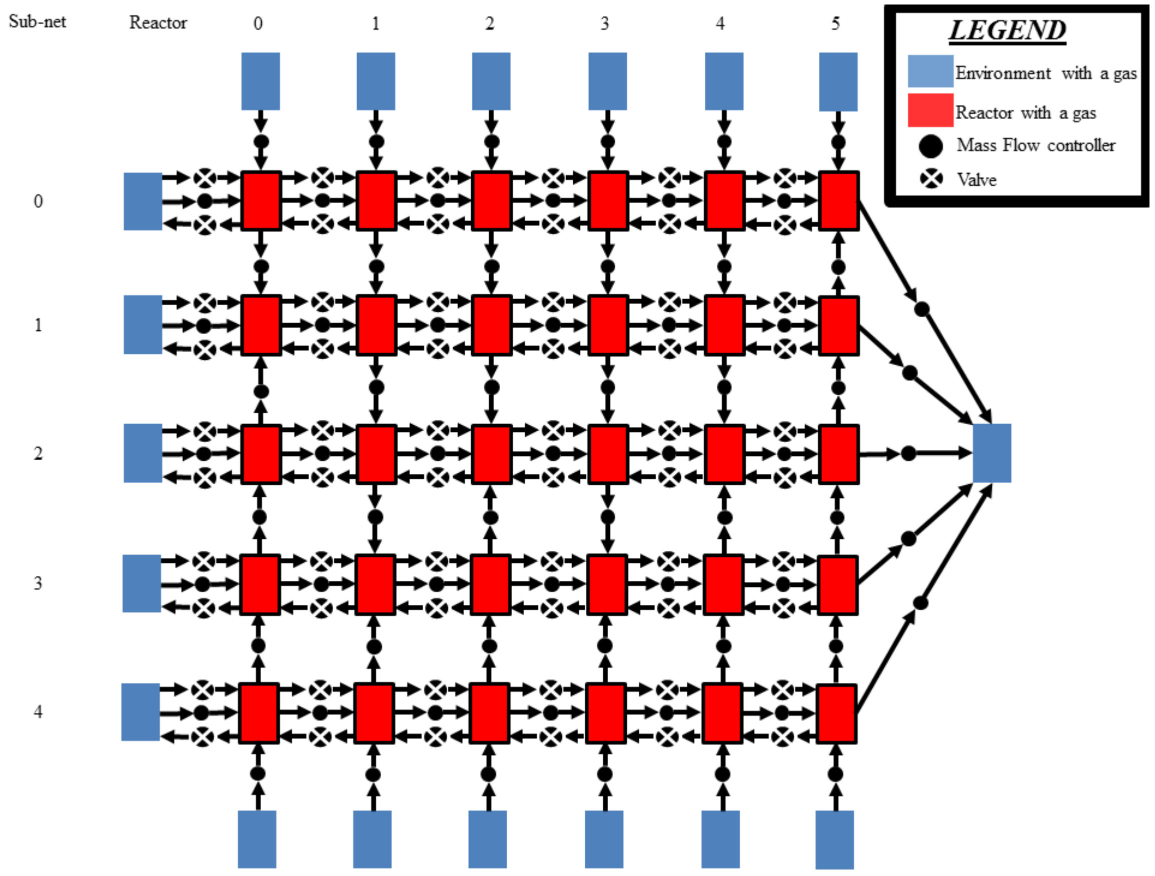


Figure 60: The form of the Cantera network which is utilized. It contains environments at the flow inlet, in addition to environments above and below (for dilution air). Each of the reactors are connected to it's neighbors based on the mass flow rates calculated from the CFD fluxes.

Note that Figure 60 only shows the Cantera network in one configuration, although numerous configurations are possible. The configuration typically only changes in size, however the mass flow controllers may change orientation, as well.

Going back to Figure 48, the next step is to run the system for a single time step. There is a disconnect between the time steps within the CFD flow solver and the CKN. In general, the time step taken by the CFD code is much longer than that taken by the CKN solver, so the total time step is determined by the CFD code (the larger of the two). The CFD code takes time steps which are a function of the grid cell spacing, the local speed of sound, etc. (as given by Equation 32[7]), while the CKN time steps depend on chemical species creation and destruction rates. Contrary to the CFD time, the physical time that the reactor steps through to solve stiff nonlinear ODEs is set internally to the chemical network solver during runtime, and may differ between CFD solver time steps. This is all done behind the scenes within Cantera using the Sundials library [106]. As such, depending on the integration time step of the solver, it is near impossible that the CKN chooses the same time step as the CFD code. Therefore, the CKN network solver takes time steps until the CFD code time is reached.

During the CKN time step, the species and other reactor properties are computed. Once the two times match, each reactor contains the correct temperature, density, species, etc. for the given physical time. Each reactor usually releases heat during the exothermic combustion process, however the flow field does not know how much heat is dispensed. This information is input into the flow field before the next iteration because the flow field influences the kinetics, and vice-versa. For example, the temperature rise changes the density, which changes the mass flow rate into the given reactor. This is how the two algorithms couple together, since the flow field is updated on each iteration in response to the temperature imbeded by the CKN. Doing this on each iteration allows the CKN and CFD codes to converge together simultaneously. This is accomplished by use of source terms, which are entered into the flow field on each iteration and represents the amount of heat release by the reactors. The iteration and convergence process is shown in Figure 48 with the addition of the convergence loop that points to the "Obtain Kinetics Mass Fluxes" step. Here, mass flow rates are recomputed to ensure that the proper responses to temperature changes are captured. This feedback mechanism is what separates the MoST algorithm from

[7]Note that this is the non-dimensional time, since the CFD code is nondimensionalized. This is put back into a dimensional time by multiplying by $\frac{L}{v}$, where L is the characteristic length by which the grid is normalized, and v is the characteristic velocity by which the grid is normalized.

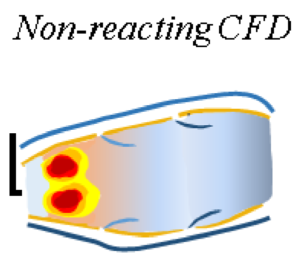
previous methods, as previous methods such as the hybrid CFD approach presented in Section 2.5 do not take any information from the CKN into account.

It is important to understand that inserting reactor information into the flow field can be done in different ways, two of which are presented here as possible methods. The first is to directly obtain heat release values given by the CKN. Each of the species in all of the reactors have enthalpies associated with them, which are used along with the net production/destruction rates to determine the actual heat release. This is given by

$$q = R \times \int_{t=0}^{t=t_{res}} dt \times V \times T \times \sum_{i=1}^{i=N} h_i \times S_i, \tag{50}$$

where h_i is the enthalpy of species i, and S_i is the "source" term, meaning the production rate of species i minus the destruction rate of species i. Therefore, the reactor network runs for a single CFD code timestep and an integral is taken over the reactor time until the CFD time is hit. This is exemplified in Figure 61.

As seen on the left hand side of Figure 61, the CFD time steps vary, and the right hand side of the figure shows the three integrations which take place to get the source terms for each of these CFD time steps. This all assumes that the heat release is computed and inserted into the CFD code directly. However, this is a computationally expensive method, since it involves computing and summing the enthalpy and net production rates of each species in every reactor for all CKN time steps. It is obvious why this becomes costly when dealing with large mechanisms (and many reactors), since the enthalpies and production rates of each species in all reactors must be multiplied and summed. Additionally, this approach inherently assumes that one has more information about what is occurring inside of the reactor than is actually known. Consider Figure 62.



EXAMPLE NUMBERS

Iteration	dt (ms)	t (ms)
1	1.7	1.7
2	0.315	2.015
3	0.4	2.415

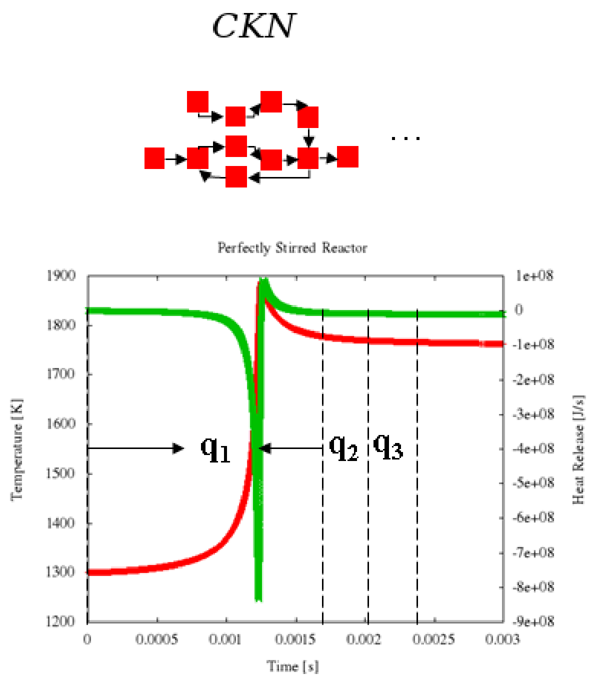


Figure 61: There is a disconnect between the CFD solver time steps and the chemical kinetics solver time steps. For each CFD time step taken, numerous time steps are taken in the chemical kinetics solver until the time is the same as the CFD time step. The total heat released is then found by integration over the smaller time steps. The times in the figure are fictitious, and are only for illustration purposes. (Left) The temperature rise is in red and the instantaneous heat release is in green.

Inputting source terms evenly is the best thing to do since nothing else is known about what occurs inside a homogeneous reactor. However, the fluxes also interact with the energy input and cause the cells downstream to become hotter due to flux advection, even though they are in the same kinetics zone. This inherently assumes that the CFD code knows more about the reactor's chemistry than the CKN, even though all of the chemical information comes solely from the CKN. This approach can still be used, but the source terms must be redistributed in order to create homogeneity within the CFD cells that represent the kinetics cell. Therefore, in addition to making unwanted assumptions, the method to input chemical information from heat release computations is a long process since it involves so many intermediate

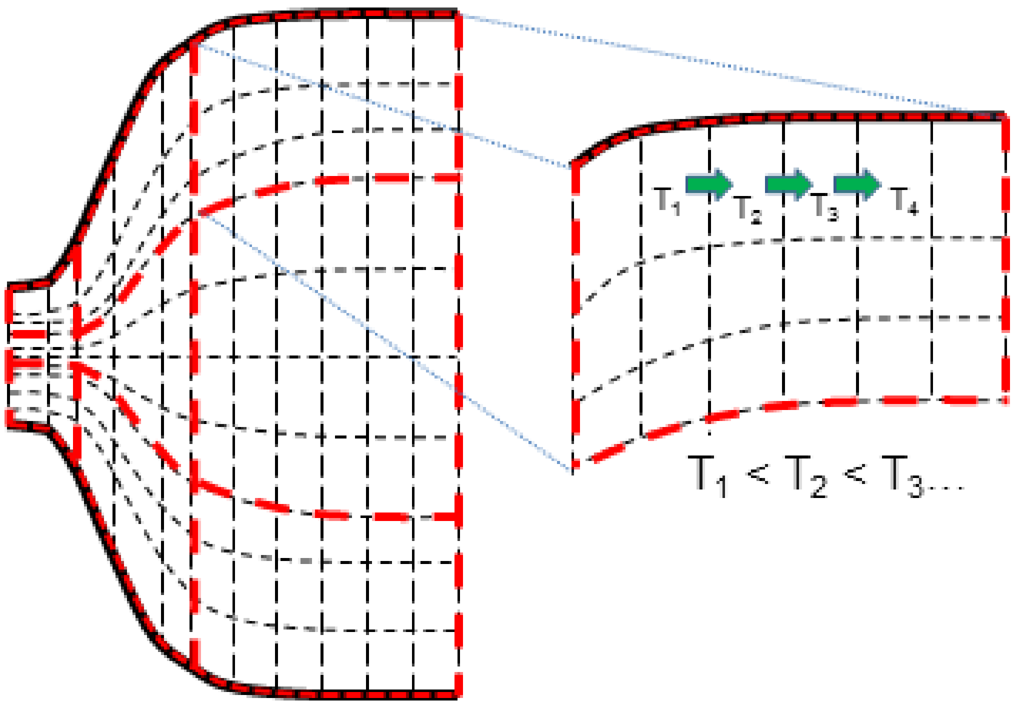


Figure 62: Inserting the heat release evenly within each segment of the CFD mesh inherently assumes that more is known about the reactor than is known in reality. The green arrows are the energy flux vectors. Equal heat releases means that cells to the right will have a higher temperature since energy flows in that direction. The red cells are the KM boundaries and the black cells are the CFD mesh cells.

computations. Lastly, utilizing the heat release information may be a great method if an implicit CFD solver is employed. This may help distribute heat release to the correct places irrespective of fluxes since all cells are solved concurrently. Since an explicit time stepping method is used here, an easier (and faster) way is sought.

Instead of using source terms, the heat release which is needed to attain the correct temperature inside of the reactor may be calculated for each CFD cell. This is done in a few steps. On the first iteration, one assumes no heat release so that the source term is zero, ensuring that the temperature is that which came from the nonreacting CFD solution. On all subsequent iterations, the source term becomes proportional to the temperature difference between the reactor temperature and the temperature of the cell in question as shown in Equation 51.

$$S_{i,j} = F \times \frac{T - T_{react}}{T}, \tag{51}$$

where T is the current temperature of the cell, T_{react} is the temperature of the PSR to which the cell belongs, and F is a scale factor which can be used to aid in convergence. Note that if F is too large or small, either too much or too little heat will be released into the cell, and a steady-state may be found with the fluxes before the correct temperature is attained. Ergo, care must be taken when choosing this factor.

Other equations may also be used to insert source terms, such as one utilizing previous source terms in addition to temperature differences. In this case, one uses a modified form of equation 52, as in

$$S_{i,j} = S_{i,j,old} + F \times \frac{T - T_{react}}{T}, \tag{52}$$

so that a change in the source term only occurs if the correct temperature has not been found yet. One must be careful using this method, also, since it can cause the temperature to jump very high and very low within small amounts of iterations due to over corrections. Over corrections result in a sharp change in the flow field if small amounts of reactors are used, since the high temperatures want the flow to spread out before the source term corrects itself on the next iteration. A limiter can be used on the temperature correction, but these results typically fall back to the case when the source term from the previous iteration is not used (Equation 51). Other methods of bringing kinetics information without biasing the flow field may be examined in future work.

Even the insertion of source terms is not straight forward, and requires a great deal of care. As soon as a source term is inserted into the flow field, the flow field is no longer in steady-state (as it was when it converged during the NRCFD run). Each reactor does not necessarily have an inflow which matches the outflow in an unsteady-state. This *must* be the case however, otherwise a reactor may be drained of mass and the code will throw errors and crash (i.e. since more flow is pulled out than is available within the reactor). Two methods have been utilized to get around this problem. The first method utilizes optimization. Here, two cells are assumed to have a correct mass flow rate (the mass flow rate in, plus one more side), and the extra side plus mass flow rate out are unknown. In this approach, a series of simultaneous equations are formed. This set of simultaneous equations have more unknowns than equations ($2n$ unknowns and n equations, where n is the number of cells), and an optimizer is used to solve the two missing mass flow rates simultaneously. Experimentation shows this works very well for small reactor systems, however large systems contain too many design variables, causing this approach to become infeasible. To make this process speedy and achievable, only the largest outflow value for the cell is assumed unknown, and a system is solved with n equations and n unknowns. This works well, and allows the user to approximate the steady-state flow solution at times when steady-state values are unknown.

The last step in the algorithm is to continuously integrate forward in time while inserting source terms until the system converges. Eventually, the CKN hits a steady-state (i.e. the temperatures are no longer changing so that a constant heat is introduced into the flow field, and the species concentrations in each of the reactors are no longer changing). This ultimately causes the source terms to stop changing (when the temperature stops changing) and the CKN converges with the flow field to the specified convergence tolerance. This is the ideal situation, so all of these parameters should theoretically be checked (temperature, species concentrations, etc.). An abundance of methods may be used to approximate when the system is converged, a few of which are considered in this work. First, convergence can be taken as the point where the CKN is run when the temperature is basically constant, but still may be changing slightly. For example, the CKN may be considered converged when the temeprature in each reactor is no longer changing by more than one tenth or hundredth of a degree for ten flow field iterations (or any user specified temperature difference or number of iterations). Secondly, the residence time of each reactor is considered. The network should be run for more than one residence time of the reactor with the largest τ_{res}. This can also be added in combination with the above temperature criterion (i.e. look for temperature changes after a residence time, or a few residence time cycles have been completed). Further consideration may be taken by checking all of the species fractions. For example, the temperature may not change by more than a certain amount, however minor species (as in the NOx of interest) may change drastically with only small temperature changes, so one must be careful. For large reaction mechanisms (e.g. 300 species), it is very expensive to store the previous ten concentrations for each reactor over numerous iterations). Here, it is much easier to check and store the temperatures, and the combination of a steady temperature over a a prescribed number of residence time cycles should be enough to consider the CKN to be at a steady-state.

No matter how convergence is achieved, NOx emission values are obtained after the flow field and CKN converge. The idea here is to find the amount of NOx flowing out of the combustor per unit time. This can be done with any gaseous species, i.e. one may want to check the carbon monoxide outflow rate to compute efficiency values, or any excess fuel, etc. NOx is of concern in this work, so it is taken as the species of consideration here. In order to get the NOx flow rate at the outlet, the last reactors in each sub-network are used, as shown in Figure 63.

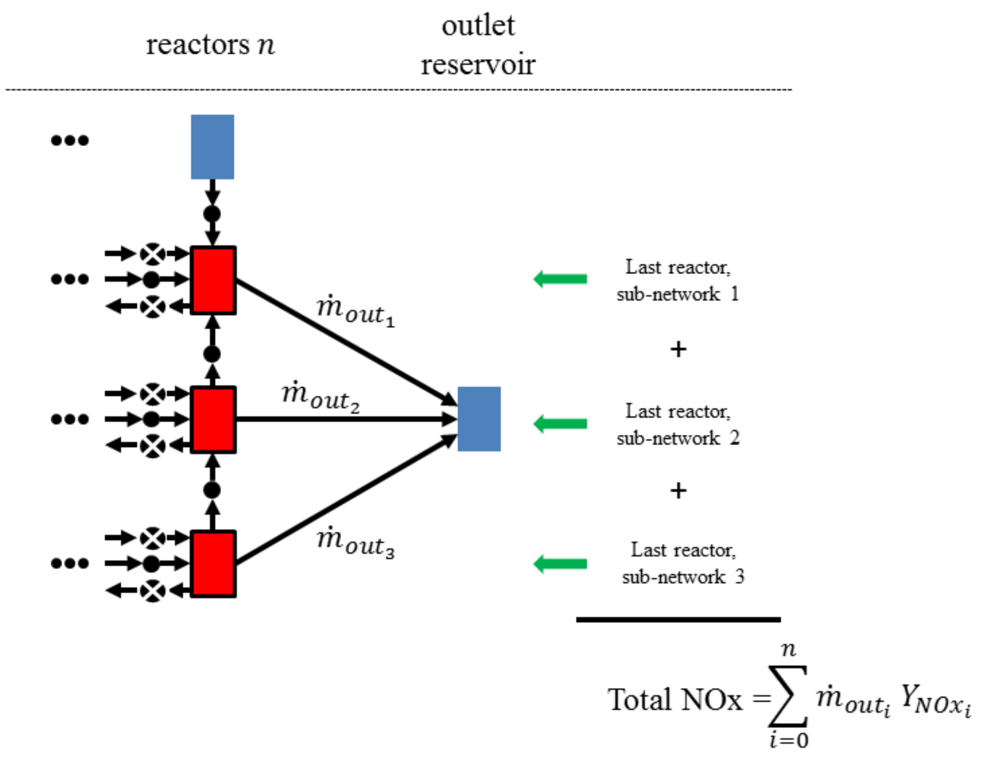

reactors n
outlet reservoir
$\dot{m}_{out_1}$
$\dot{m}_{out_2}$
$\dot{m}_{out_3}$
Last reactor, sub-network 1
+
Last reactor, sub-network 2
+
Last reactor, sub-network 3
Total NOx $=\sum_{i=0}^{n} \dot{m}_{out_i} Y_{NOx_i}$

Figure 63: Obtaining NOx values from the CKN.

These reactors contain information such as temperature, volume, species concentrations, etc. The species mass fractions are used in combination with the outlet mass flow rate from each reactor to the outlet reservoir and summed. This gives the total mass flow of NOx emissions leaving the system. Since each reactor contains the mass fractions of all species which are present, the mass fractions of any gaseous species in combination with the mass flow rates can be used to obtain the outflow rate of any gaseous emissions.

Lastly, the approach for tying (and iterating between) the non-reacting CFD code and reactor network to get a converged reacting flow solution is a very important point. Figure 48 shows an iteration loop, where kinetics information is exchanged with the CFD portion. As the methodology is presented above, this data exchange occurs between the reactor network and the CFD solver at *every* iteration. This capability makes the MoST algorithm general enough to handle both unsteady flows (information exchanged every iteration) and steady flows (information exchanged much less frequently, perhaps every hundred or thousand iterations). For this reason, it is very important to obey the requirement that all variables are kept in memory so as to avoid reading and writing to and from files every time an iteration takes place (Chapter 4). Further work can be carried out with different types of iteration schemes to determine the best fit for given applications (see Section E.3).

CHAPTER V

RESULTS

The most accurate emissions results one can obtain is from experimental testing, since this gives the "true" NOx value. Without these data, it is difficult to assess the method's accuracy. Ideally, one would like to utilize experimental results from an engine which has been tested (i.e., from the ICAO databank, where emissions results are publicly available [107]). Though emissions data are in the public domain, the combustor designs themselves remain highly proprietary and are not typically released to the public. This makes comparisons difficult, since the geometry is needed to run the MoST algorithm. When either generic correlations or correlations for a specific architecture are used [108, 109, 110, 111], all aspects of the combustor design are not taken into account. For example, using different correlations to predict NOx will give vastly different results (Figure 64). This method may be compared to NOx approximations using other techniques such as reacting flow CFD, though the geometry of the combustor is still necessary. Even in reacting flow CFD, large reaction mechanisms are not widely available, nor are they typically run (Section 2.4). Since reacting flow CFD runs are too slow to be used to accurately obtain NOx by direct computation, various NOx approximation techniques are employed.

5.1 Reacting Flow CFD and Emissions Approximations

Direct Numerical Simulation (DNS) has become somewhat uncommon in reacting flow CFD due to the wide range of time and length scales of turbulent features [112]. This necessitates too many cells in order to capture the flow across the multitude of length scales. Instead, turbulence models are used [113, 114] in combination with combustion models, such as a the eddy breakup model (EBu) [114], presumed probability density

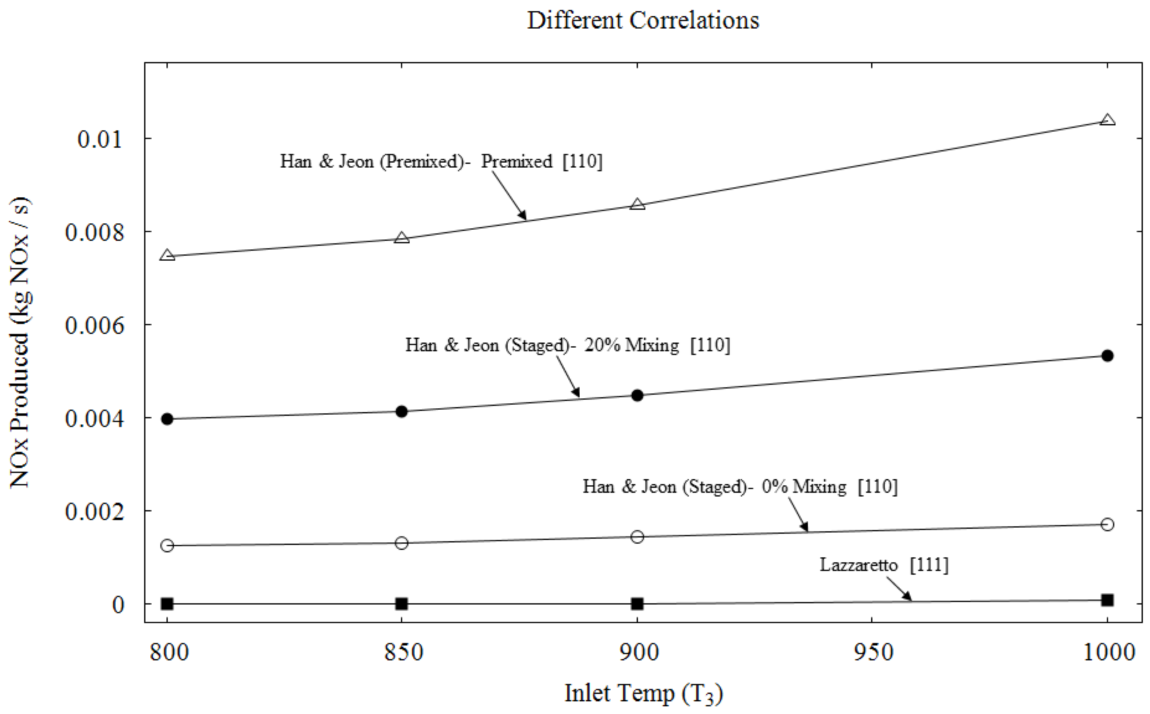


Figure 64: Blindly using generic correlations (or correlations specific to a given architecture) can give vastly different results. Here, four correlations are used for the notional combustor shown in Figure 66, and results differing by orders of magnitude are obtained.

approach (PPDF) [114, 115], and flamelet generated manifold (FGM) model [114, 116, 117]. The EBu model does not give as good temperature predictions since it computes the rate of fuel consumption via a characteristic mixing time. This is determined as a ratio of turbulent kinetic energy to dissipation rate, and leads to unphysically high temperatures and high reaction rates in the present situation [118]. Narrowing the choices to the PPDF and FGM models, a software package which handles these must be chosen for validation. STAR-CCM+ is selected [119], however the use of the FGM model requires the DARS chemical package [120, 121]. Because of this, the PPDF model is chosen for use in the validation study.

In contrast to the EBu model which tracks all species, the PPDF equilibrium model only tracks the mean mixture fraction while assuming the flow is in instantaneous chemical equilibrium conditions [119][1]. The time averaged value of the mixture fraction is given by

$$\bar{\phi} = \int_{0}^{1} \phi(f,h), P(f,h) df dh, \tag{53}$$

where f is the mixture fraction, h is the enthalpy, and ϕ is the mixture fraction probability density function [119, 122]. In the PPDF model, these integrals are already calculated using a 1-D opposed jet calculation as in [123], and the chemical species and reaction information come directly from a look-up table which is precomputed before runtime. In actuality, much theory goes into the plethora of CFD combustion models which have been developed over the decades, but only a small overview of the specific model which is used is given.

Additionally, many emissions predictions models exist to combat the fact that CFD models do not track or even include all of the species necessary to compute emissions. This includes models for soot, NOx, and other types of emissions. Each main NOx formation mechanism has its own approximation techniques, such as prompt NOx, thermal NOx, etc. Since thermal NOx is of interest here, popular estimation approaches include the equilibrium and partial equilibrium assumptions [124, 125], though other approximations are sometimes used (e.g. an exclusion assumption). A partial equilibrium assumption is used in the present RFCFD simulations, since this model is available within STAR-CCM+. This approach only simulates an estimated NO production rate based on the O, H, and OH (radical) concentrations. The reaction rate coefficients for these reactions have been tabulated, and are available using this software [119, 126]. Note that this really is an approximation, since extremely

[1]Though equilibrium conditions are not the ideal case, it is the only option since finite-rate chemical effects are not available without the DARS package.

large mechanisms (their molecules and interactions) are important to truly take NOx production into account (see Section 2.4).

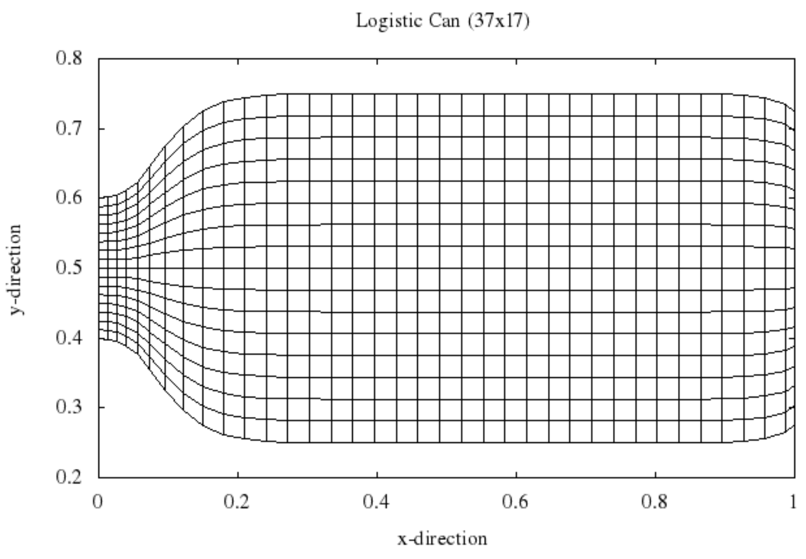
5.2 *Canonical Problem*

With no actual combustor geometry, a simple canonical combustor is used to test the method. This is run using full reacting flow software which has been validated and released (STAR-CCM+), where NOx approximations are used to obtain estimates. The overall strategy for comparison is to therefore develop a mesh which resembles a combustor to run in the MoST algorithm, and then use the same case to run in a full reacting flow environment to obtain NOx approximations.

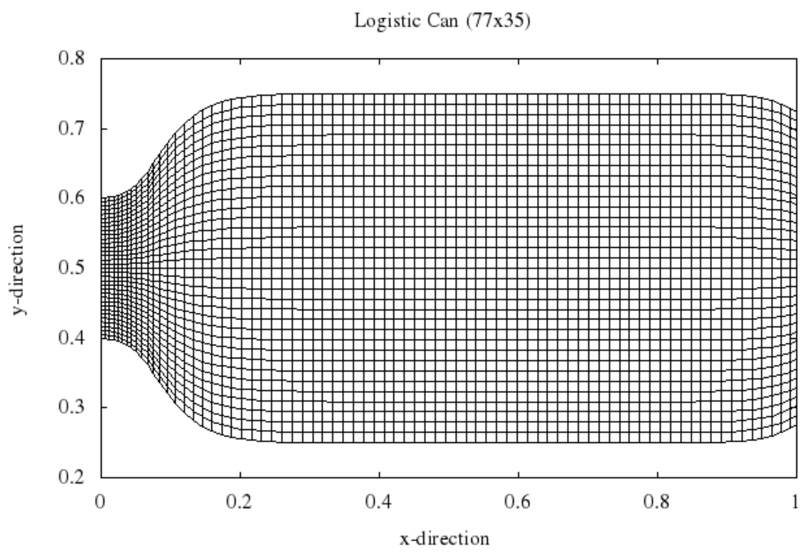
The test case that is used here is a simple can combustor, whose shape is defined with logistic functions given by

$$f(x) = \begin{cases} \frac{0.6+0.4}{\frac{8}{3}+e^{-30(x-0.2)}}, & \text{if } x < 0.5m \\ \frac{0.7+0.3}{6+e^{40(x-0.95)}}, & \text{otherwise} \end{cases} \tag{54}$$

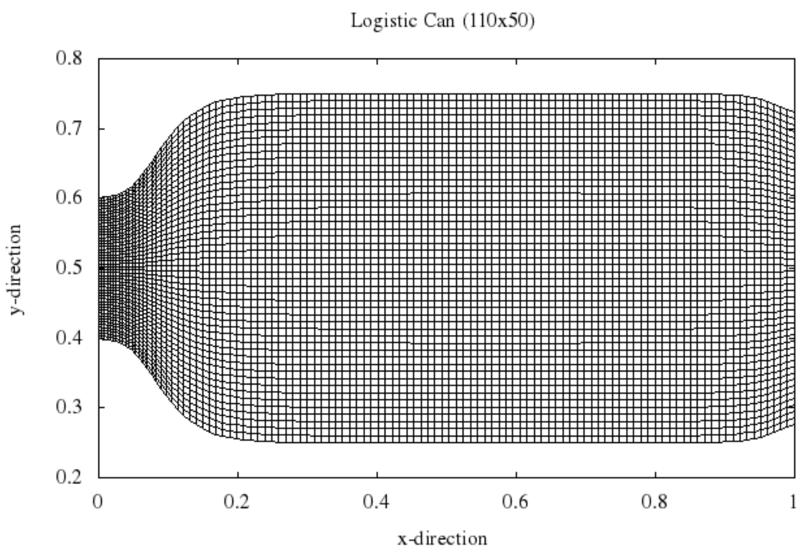
Numerous grids of this combustor are created and tested using the code described in Section 4.1.1, ranging from low to high fidelity. The grids are shown in Figure 65. Three grids (222x100 nodes / 21,879 cells, 445x200 nodes / 88,356 cells, and 1116x500 nodes / 556,385 cells) are left off due to the density of the mesh, which pictorially looks like a single filled object. The grids are all run with the same flow properties and boundary conditions, as shown in Figure 66.



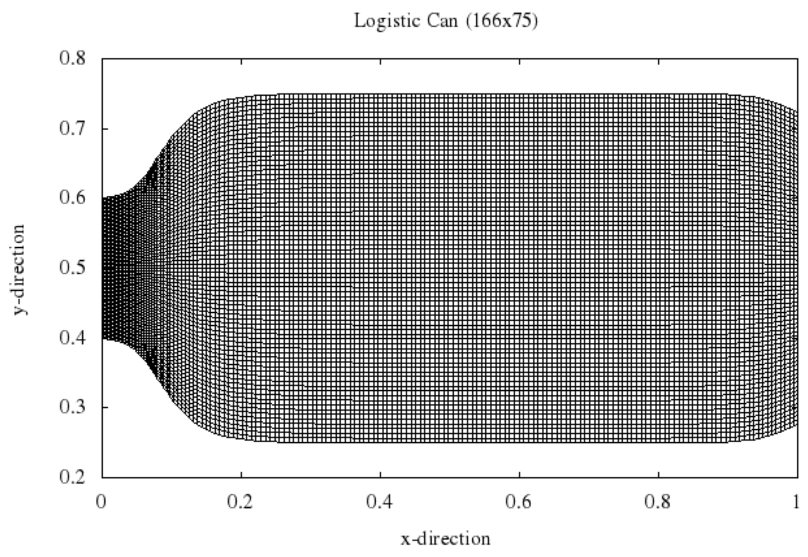
(a) Logistic Can with 576 cells



(b) Logistic Can with 2,584 cells



(c) Logistic Can with 5,341 cells



(d) Logistic Can with 12,210 cells

Figure 65: Numerical meshes for canonical test case.

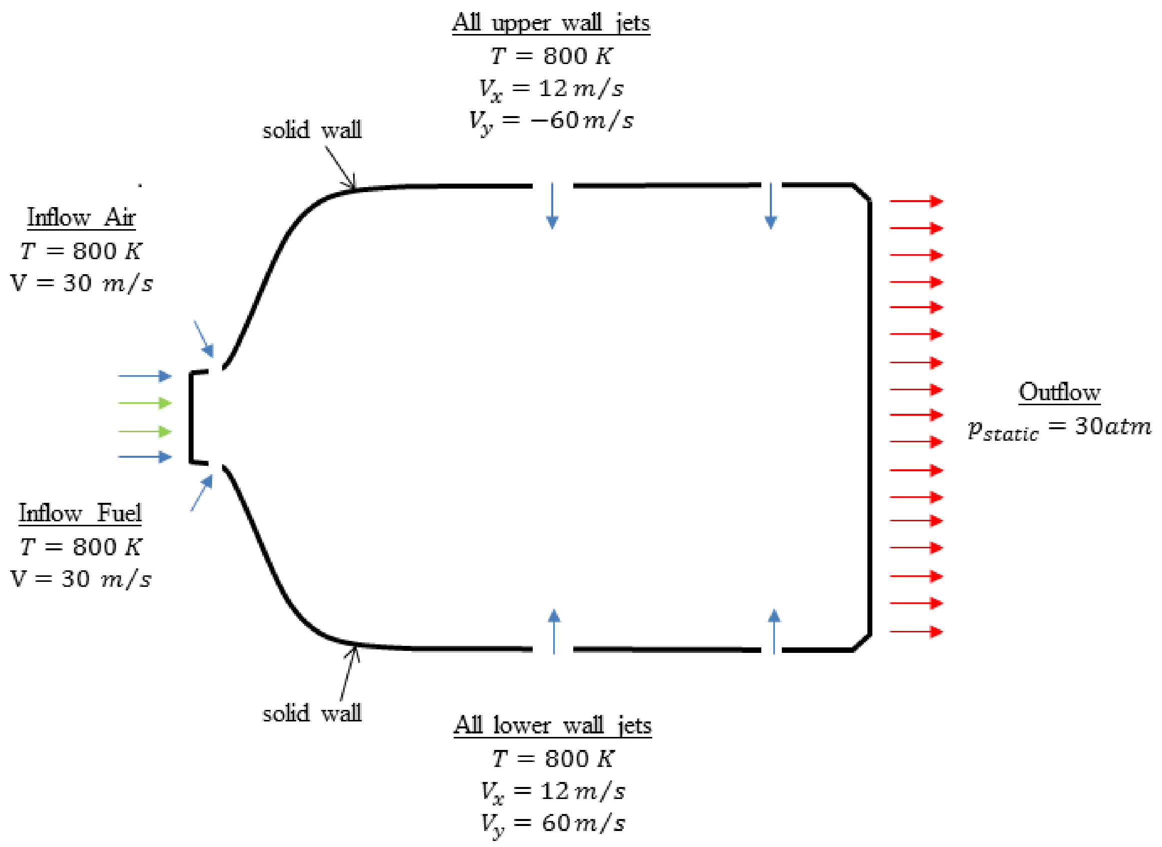


Figure 66: Boundary conditions and flow properties for the combustor can which is run to demonstrate proof of concept. The grids are shown in Figure 65.

5.2.1 Canonical Problem Using Reacting Flow CFD (PPDF)

The reacting flow CFD PPDF validation model is run on five grids (the four grids shown in Figure 65 plus the 222x100 grid) using ten species, CH_4, N_2, CO_2, O_2, H_2O, OH, CO, H, H_2, and O. Ten species are chosen because a smaller number of species (global reactions) do not converge very well, as the flame has trouble reacting without radicals. The simulation is run using the standard k-epsilon turbulence model (with viscosity turned down to a very low value) and a steady, non-adiabatic PPDF model. Additionally, the thermal NOx (Zeldovich) model is used to obtain NOx approximations (other NOx formation mechanisms such as prompt NOx, etc. are neglected). The expectation is to converge on the correct NOx value with an increase in number of grid cells (an increase in mesh fidelity). Figure 67 shows the variation in NOx estimates with grid size for four inlet temperatures. Here, the boundary values for the canonical problem in Figure 66 are used, with the inlet temperatures being the only property that is changed.

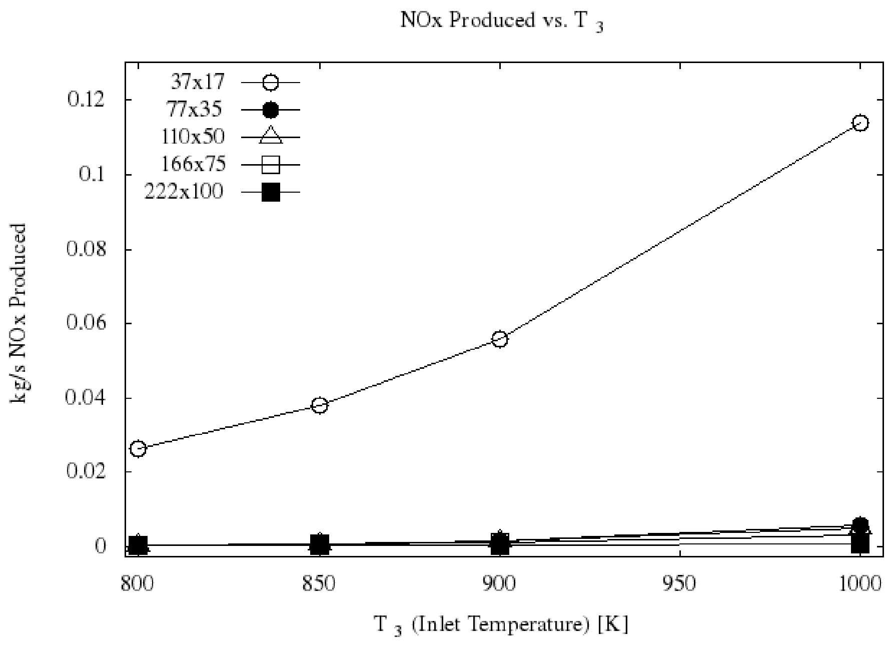


Figure 67: NOx vs. T_3 using STAR-CCM+ PPDF combustion model for varying grids. This plot shows NOx as a function of inlet temperature ($T_3 = 800K$ to $1000K$, inclusive). These lines are not those of best fit, but simply connecting the points to show which data points belong to the same group.

If the grids are perfect (i.e. represent the actual physical combustor can and combustion process), the lines and points for each size grid in Figure 67 should all fall on top of one another. As may be inferred, the 37x17 grid is extremely course, and is unable to give good approximations based on the fact that it is unable to resolve flow features and thermodynamic properties of the flow field very well. Removing these data due to the courseness of the grid, four of the higher fidelity grids (77x35, 110x50, 166x75, and 222x100) are left. The NOx data as a function of temperature are shown again in Figure 68 for these grids only.

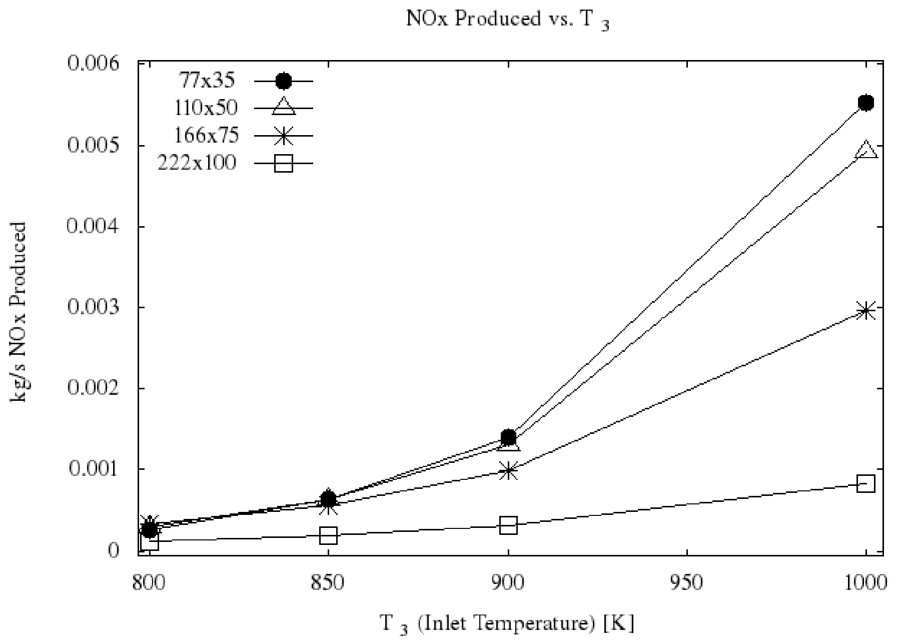


Figure 68: NOx vs. T_3 using select STAR PPDF combustion models for select grids. These lines are not those of best fit, but simply connecting the points to show which data points belong to the same group.

The CFD NOx model does not necessarily give the predictions one expects. As the grid fidelity increases, the data points should converge, however they only seem to get lower. This may occur if the "true" NOx value is actually smaller, or may be due to the fact that the code is unable to capture the temperatures and reaction rates correctly in the PPDF model. Thermal NOx production is extremely heavily dependent on flame temperature, so if the CFD code does not compute this accurately then the NOx values will be off. This is better shown with the addition of Figure 69, which displays that the maximum flame temperature value does not necessarily converge across all conditions.

NOx emissions are extremely difficult to capture without correlations, and the numbers are extremely small; thus making it difficult to determine which CFD result is the correct one. This is evident by the fact that validated software does not capture the correct NOx values or converge very well on the correct temperature. Even though

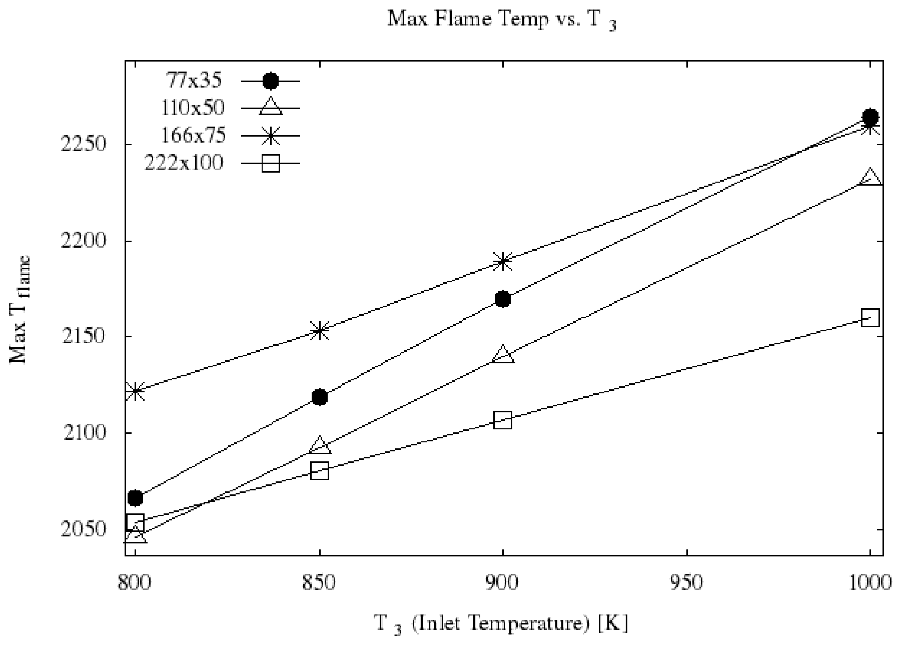


Figure 69: Max T_{flame} vs. T_3in select STAR PPDF combustion models for varying grids. These lines are not those of best fit, but simply connecting the points to show which data points belong to the same group.

all of the NOx values for a given temperature do not sit on top of one another, there is a clear trend that NOx increases as inlet temperature increases. Specifically, increasing the inlet temperature increases the maximum attainable flame temperature within the combustor. The increase in flame temperature causes an increase in thermal NOx [127], which is seen in the output.

Theoretically, one should achieve an exponential increase in NOx formation with inlet temperature [128, 129, 130, 131]. To be exact, thermal NOx follows an exponential curve with increasing air preheat temperature [128]. To ensure the CFD model behaves correctly, exponential curves are fit, as shown in Figure 70.

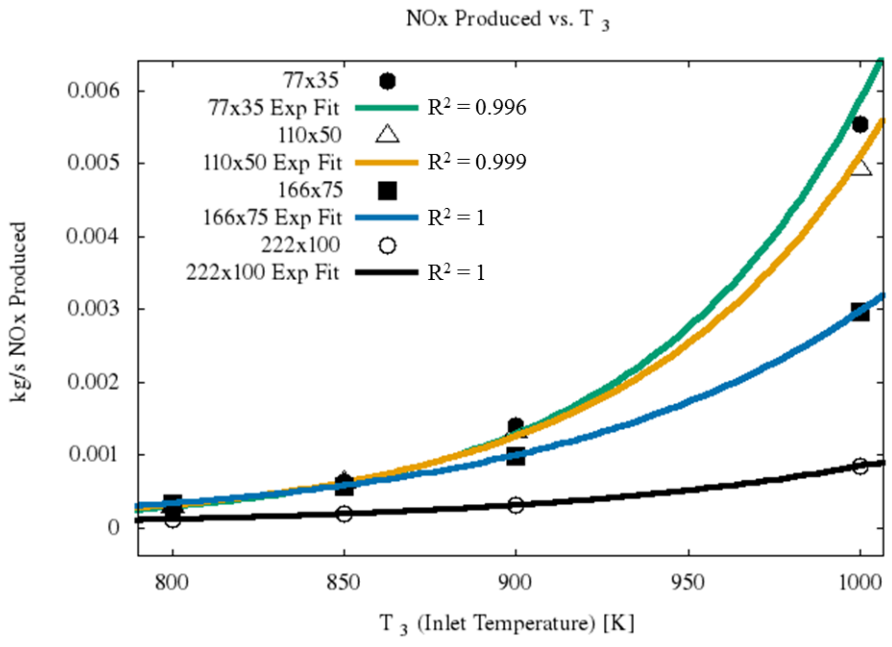


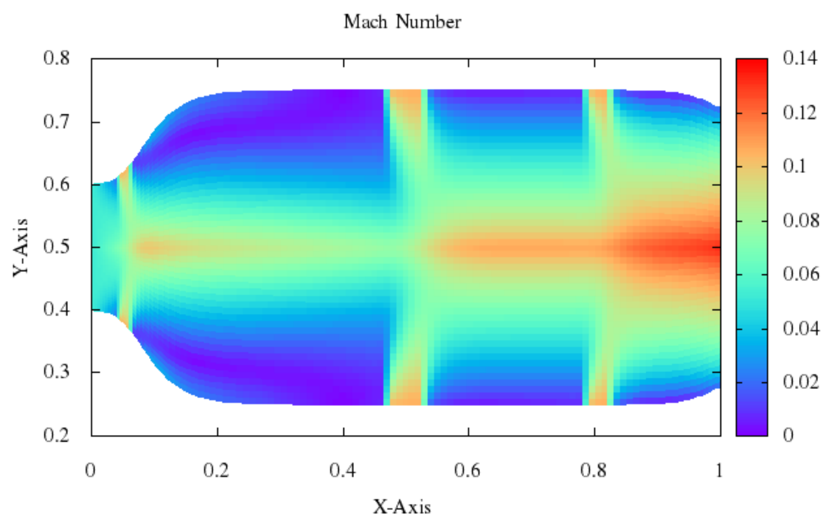
Figure 70: Theoretically, thermal NOx should follow an exponential increase with preheat temperature. The coefficient of determination (R^2) shows that this is the case. The lines here are the lines of best fit.

The amount of NOx produced behaves as expected. The flow field contours and RMS error values for the 222x100 node grid are shown in the Appendix, Chapter C. The MoST algorithm is run next so that a comparison can be made between the STAR-CCM+ PPDF RFCFD solution and that obtained using the Method of Sparse Thermochemistry.

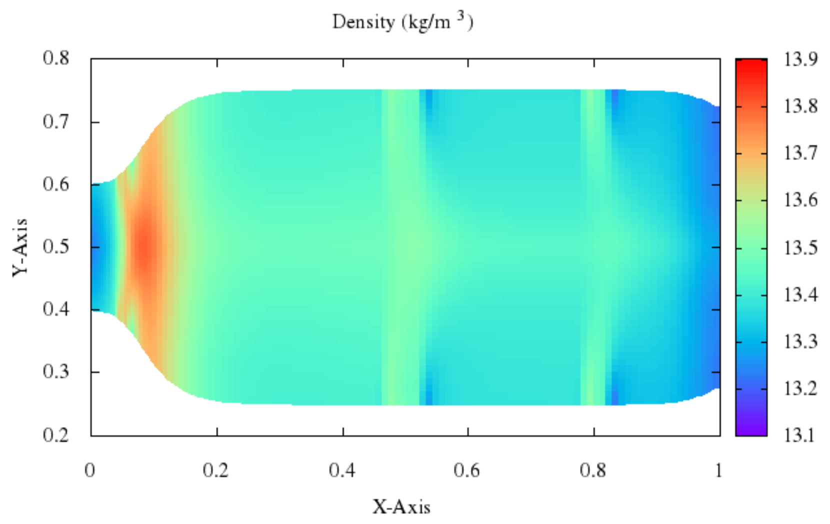
5.2.2 Canonical Problem Using the MoST Algorithm

Based on Figures 67 and 68, the grid fidelity which gives decent results starts with the 77x35 node grid. The most accurate RFCFD results are arguably obtained with the 222x100 node grid since it is the highest fidelity, so this is used for comparison. MoST results are presented in this section using the 77x35 and 110x50 node grids as canonical test cases. The 77x35 grid should give good results, and the 110x50 node grid is a good compromise between low and high fidelity, so it should yield results for a good overall comparison of how the MoST method performs. Running two grids also allows one to obtain emissions sensitivities to varied grid sizes.

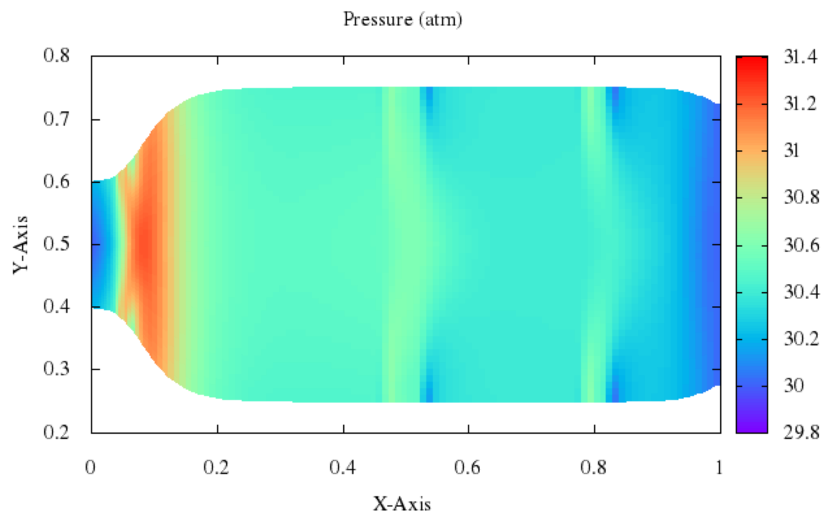
The first step in the MoST algorithm is running the non-reacting CFD solution (Section 4.1.2). The equations which are run are all inviscid (Euler equations), since the turbulence intensity is very low in the RFCFD simulations so the flow is driven almost purely by advection. The non-reacting CFD solution is shown in Figure 71 for the $T_3 = 800K$ case. Additional solutions using varying inlet temperatures and varying grid sizes at $T_3 = 800K$ are shown in the Appendix, Chapter D.



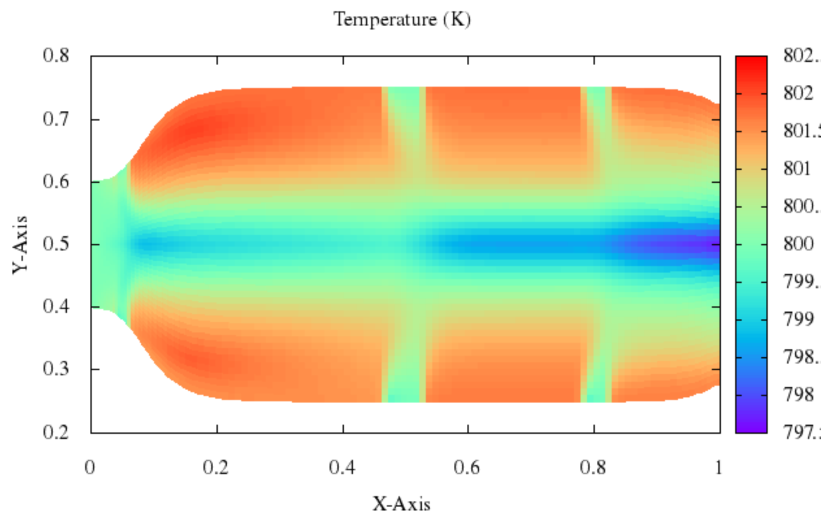
(a) Mach number contours



(b) Density contours



(c) Pressure contours



(d) Temperature contours

Figure 71: Non-reacting flow property contours for 110x50 node grid of the combustor can concept used for the runs ($T_3 = 800K$). This was run to an absolute RMS of 1.0×10^{-6}.

Section 4.3.2.2 discusses that a convergence criteria must be chosen in order to determine when iterations within the MoST algorithm are complete. It was stated that storing and checking all species over many iterations is expensive, and that an approximation for steady-state can be made by looking at temperature. This approximation is made here, with the addition of another constraint: Each simulation which is compared should be run to the same residence time. This ensures that species production is not being biased in one simulation over another. Although temperature changes may subside after a large number of iterations, some minor species, such as NO_2, may continue to change. This simply means that when comparing a $900K$ simulation to a $1000K$ simulation for example, the NOx production should be compared at the same physical system time. Therefore, the runs here are carried out with the criteria that a run is complete when:

1. The temperature within *each* reactor stops changing to within one thousandth of a degree for ten iterations, and
2. The same residence time has been reached for all reactors in the simulations which are being compared.

As discussed and displayed in Section 4.3.2.1, different mixture fraction gradient thresholds are used to determine where the reactors sit. Setting the threshold equal to unity means that only eight reactors are used in the configuration because of the initial split (Figure 49), since the mixture fraction only varies from zero to one. As an initial test, the 77x35 and 110x50 node grids are used with eight reactors to predict emissions using the same inlet temperatures as those used for the RFCFD results. These reactors are placed in accordance with the steps in the Most algorithm described in Section 4.3, and contain two subnetworks with four reactors in each. After the flow controllers are inserted, and the input and initial properties are obtained, iterations ensue until convergence. After convergence is achieved, the NOx emissions are computed. The NOx emissions results are shown in Figures 72 and 73 for each inlet temperature.

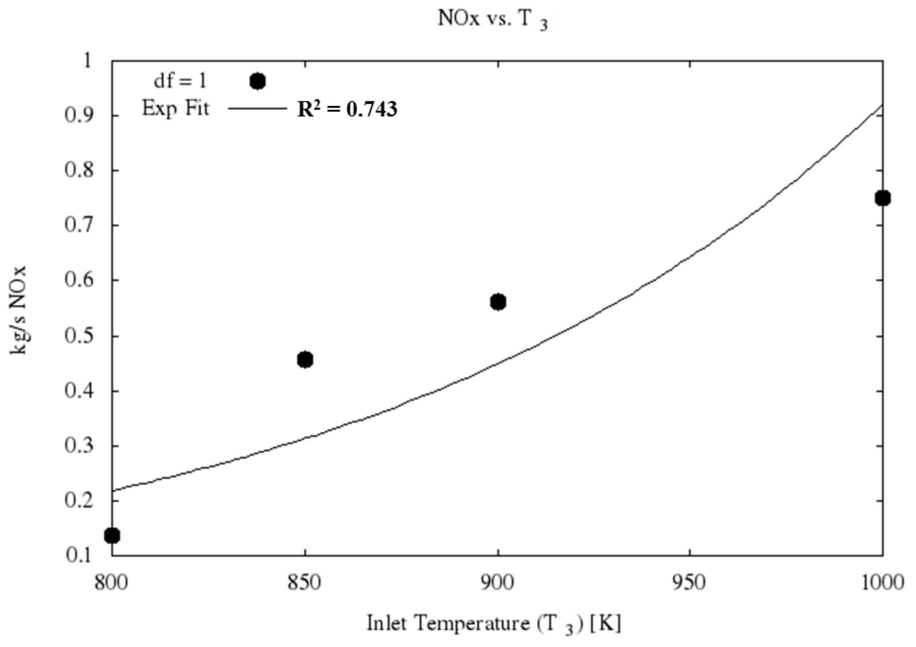


Figure 72: Emissions results using the MoST algorithm on a 77x35 node grid. Here, $\partial f = 1$ for varying inlet temperatures, which creates 8 reactors.

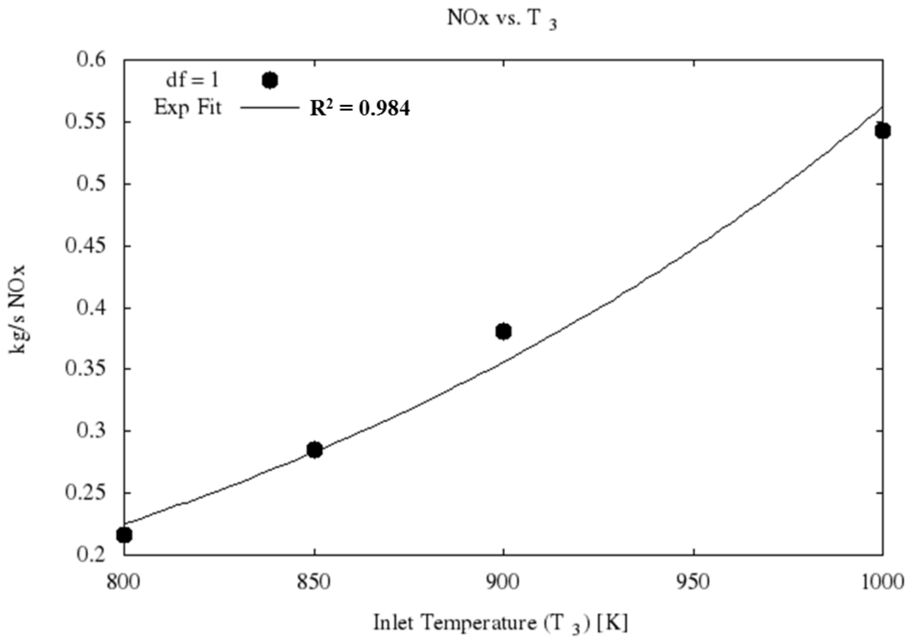
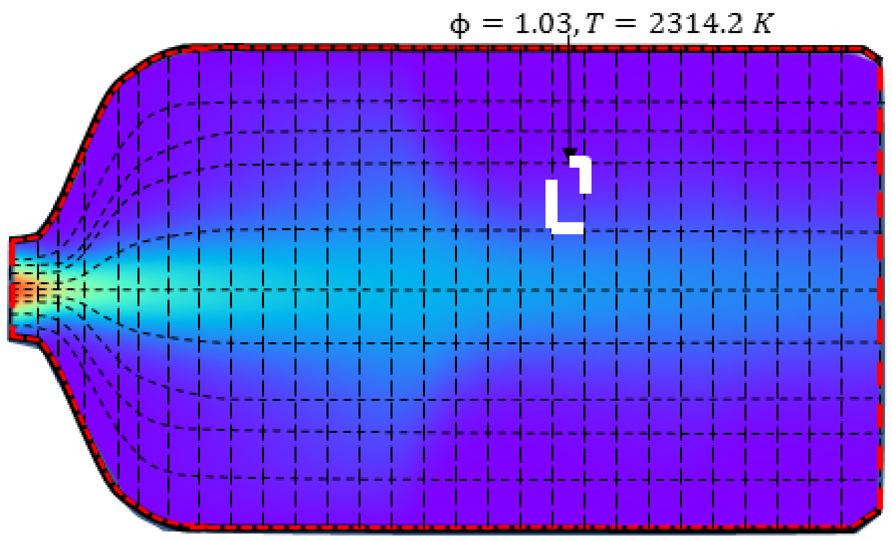


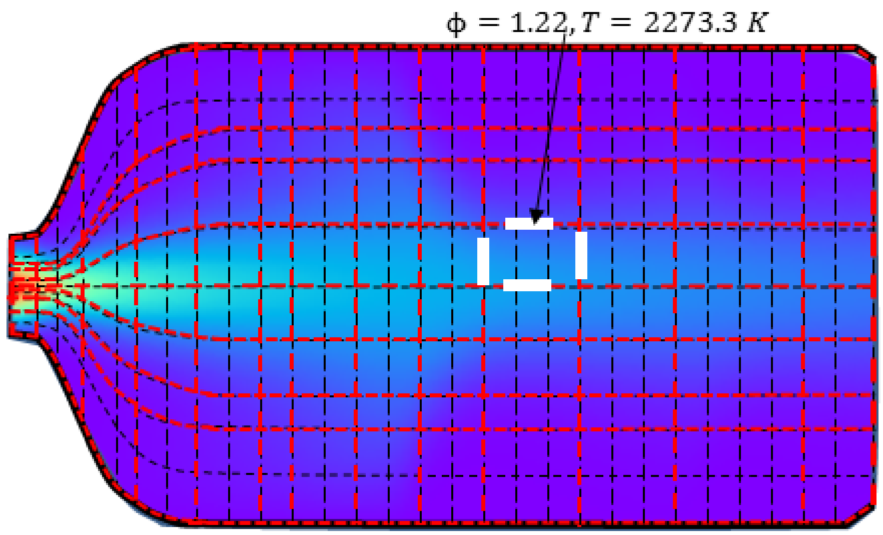
Figure 73: Emissions results using the MoST algorithm on a 110x50 node grid. Here, $\partial f = 1$ for varying inlet temperatures, which creates 8 reactors.

A few conclusions may be drawn from this. First, the 110x50 node grid out performs the 77x35 node grid (in terms of theoretical considerations to produce NOx exponentially with inlet temperature) at this mixture fraction threshold. The data should be predicted by an exponential, which is the case for the more dense grid, but the coefficient of determination value is quite bad for the lower fidelity grid. This is due to the fact that the sum of fuel and air in each reactor are not resolved as well (compare Figures 117 and 71), and this has a much larger impact when the reactors are sparsely placed. The second conclusion to draw is that the NOx values are over predicted by about two orders of magnitude in both cases. This is not surprising, and motivates a very interesting discussion point for this method.

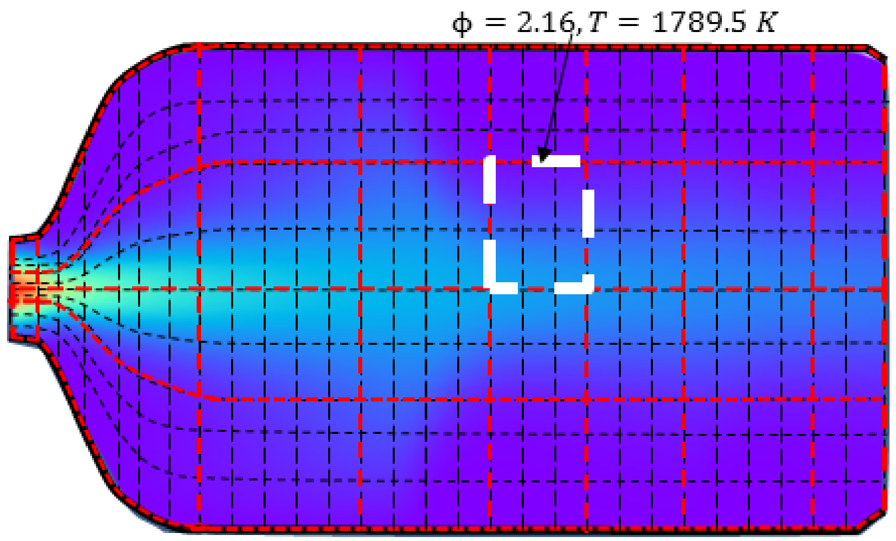
For the results where $\partial f = 1$, many CFD cells get approximated by one kinetics cell, which is the most extreme case and represents the sparse limit. In this case, one eighth of the combustor can is approximated by one reactor. As inferred, this is a pretty bad approximation, since very large differences in mixture fraction exist between the CFD cells contained within each kinetics cell. This is a very important point in general, and is displayed in Figure 74. In the case where $\partial f = 1$, the reactors are placed such that the fuel and air mix in proportions closer to stoichiometric than exist in many of the individual cells for the RFCFD run, making the temperature higher and therefore the thermal NOx higher (the split for $\partial f = 1$ is shown in Figure 49). Better approximations are expected when using lower mixture fraction gradients, since the mixture fraction in each reactor gets closer to that in each RFCFD cell. Note that this can also go the other way. Using this approach, the temperature can also be lower if the grid is too sparse, so an appropriate number of cells must be utilized.



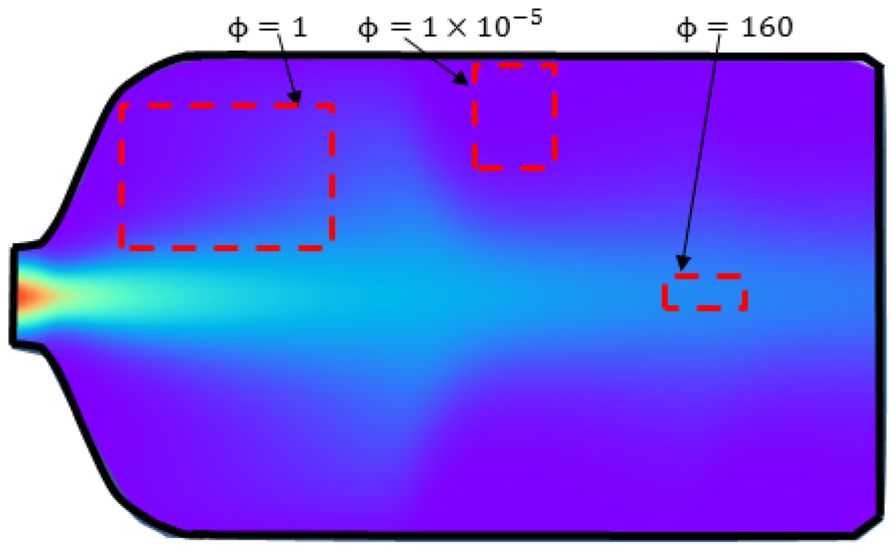
(a) Notional equivalence ratio for a given cell captured by CFD



(b) A notional good mixture fraction captured by the MoST



(c) A notional bad mixture fraction captured by the MoST (∂f is too large)



(d) Creating reactors with any equivalence ratio (notional)

Figure 74: Notional comparison of equivalence ratio when segmenting the flow field into KZs. Figure (d) shows that theoretically, a reactor with any equivalence ratio can be created depending on where the reactor is placed, so caution must be used when determining a mixture fraction threshold. Black lines are CFD cells, red lines are kinetics cells.

Next, the mixture fraction gradient is changed for both grids to see how much of an effect this has on NOx, and to deduce how far the results are from the CFD results. These data should maintain the exponential relationship with T_3, but should also get closer to the CFD results. The mixture fraction gradient is changed to $\partial f = 0.9$ (which gives sixteen reactors) and then to $\partial f = 0.8$ (which gives twenty-four reactors). The MoST algorithm is then run until convergence, at which time the NOx results are obtained. These results are shown in Figures 75 for the 77x35 grid and 76 for the 110x50 grid, respectively.

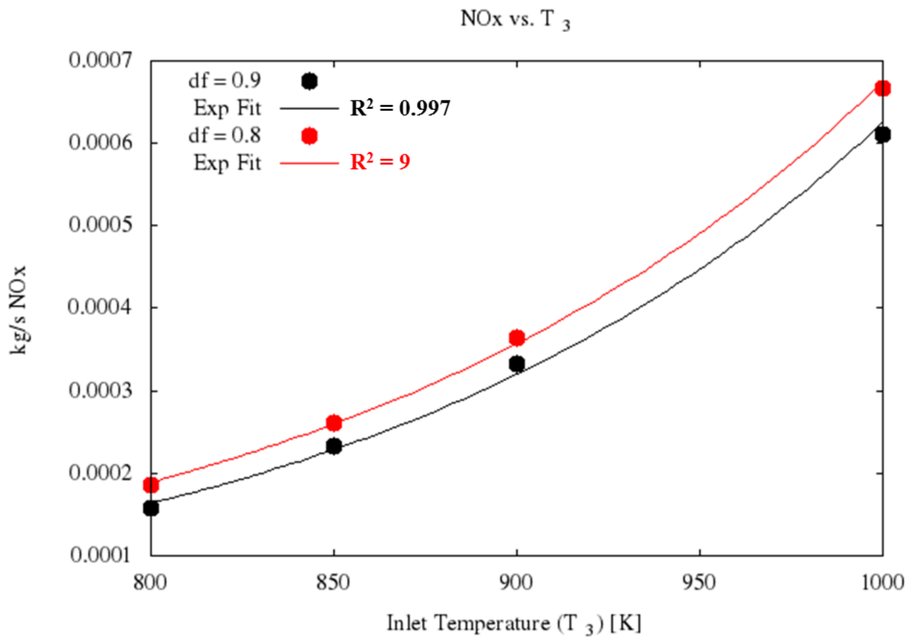


Figure 75: Emissions results using the MoST algorithm with a 77x35 node grid. Here, $\partial f = 0.9$ and 0.8 for varying inlet temperatures, which creates 16 reactors and 24 reactors, respectively.

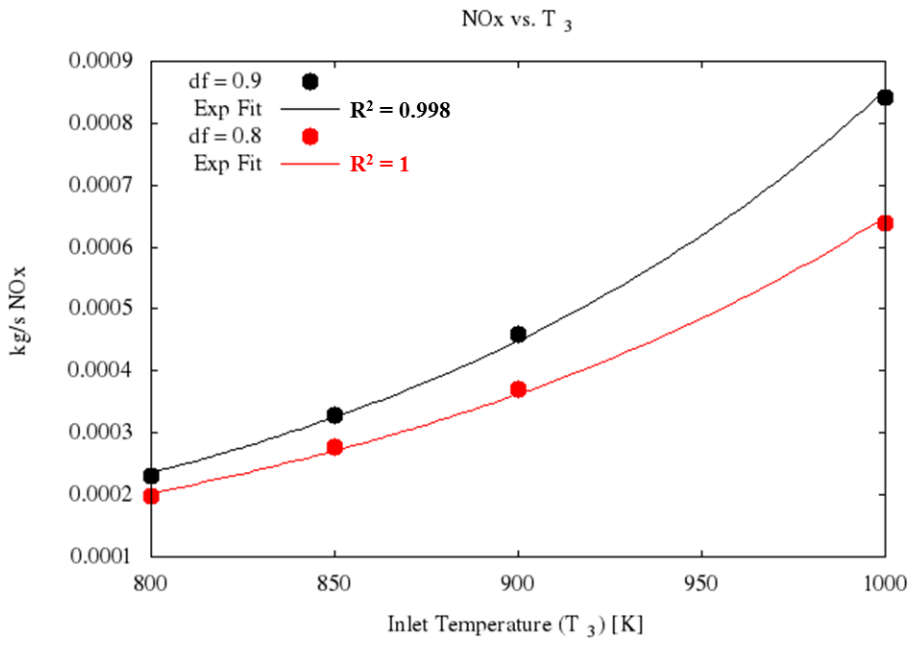


Figure 76: Emissions results using the MoST algorithm with a 110x50 node grid. Here, $\partial f = 0.9$ and 0.8 for varying inlet temperatures, which creates 16 reactors and 24 reactors, respectively.

A few things are immediately noticed. First, there is an large drop in the magnitude of the NOx results; they are much closer to those orders presented in Figure 70 for the RFCFD results (especially for the most dense grid). Second, all of the fits are very good. The R^2 values are extremely close to unity showing that the model does an excellent job predicting the theoretical trends. To directly compare to the RFCFD results, the 222x100 node grid is used and these RFCFD results are plotted along side these MoST results in Figure 77.

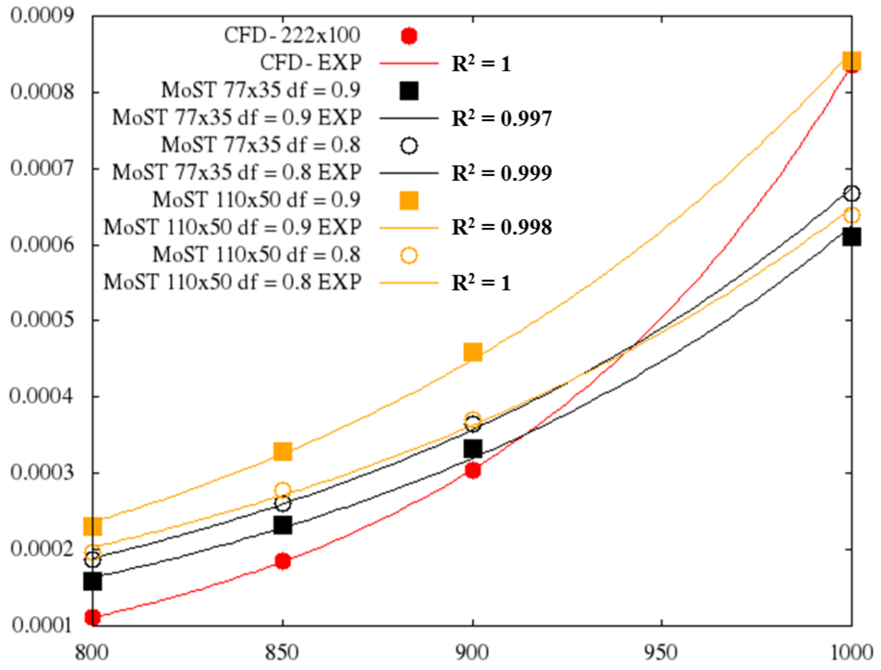


Figure 77: Comparison of the RFCFD PPDF model on the 222x100 grid against the MoST algorithm using 16 and 24 reactors on a 77x35 and 110x50 grid with mixture fraction gradient thresholds of $\partial f = 0.9$ and $\partial f = 0.8$.

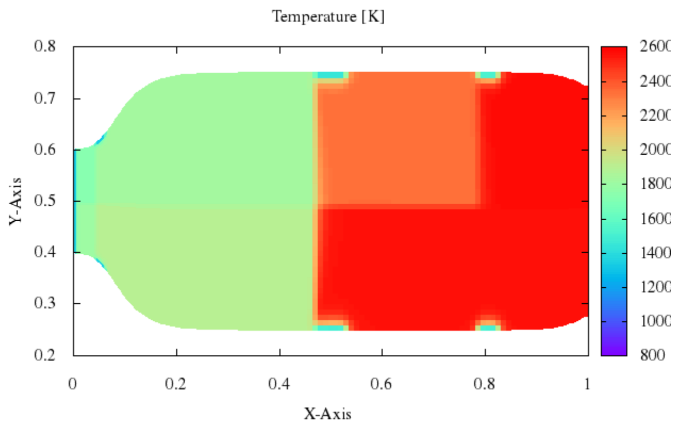
The reason for this large drop in NOx production is attributed to the temperature change attained by the individual reactors when the cells are grouped (as displayed in Figure 74). Due to the amount of fuel and air which are assigned to given reactors by the algorithm, the temperature lowers since the mixture is further from stoichiometric. The temperature contours for the final solution are shown for each mixture fraction gradient threshold in Figure 78, and shows the temperature at each point in the

CFD field (which are homogeneous for the locations within a given reactor). The temperature of each reactor as a function of time is shown in Figure 79 for different mixture fraction gradient thresholds.

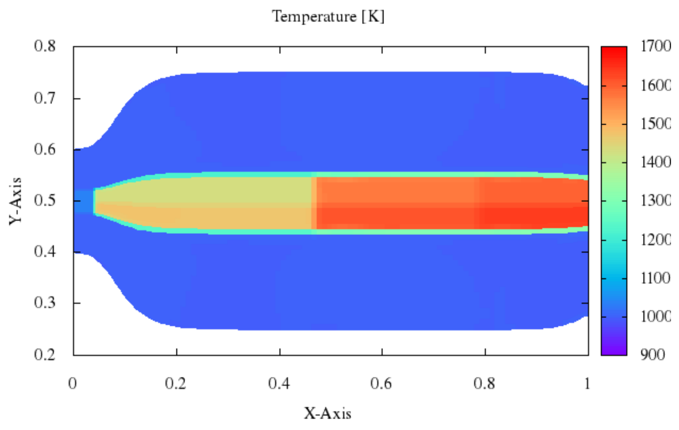
Note that the aforementioned temperature change is also seen with grid fidelity when running RFCFD: As the grid becomes finer, the fuel/air ratios become more resolved and the correct NOx obtained (Figure 70). Additionally, it is evident in Figure 79 that the flow field is adapting to temperature changes. Just after initial start up, the temperature curves rapidly change as the mass fluxes change, based on the flow field's response to the temperature changes. As the temperature in each reactor increases, the density decreases, causing the mass entrainment rates to change between the reactors. This directly affects the mass flow rates (as shown in Equation 49). This is an iterative process for each of the reactors until the flow field relaxes to its converged state.

The start up process is shown in Figure 80, which is a plot of the beginning part of the simulation in Figure 79(c). Here, the increase temperature effects the density and flow rates, which cause changes in the amount of fuel and air to each reactor. As these amounts change, the temperature varies until all differences subside and the system finds a solution. Certain reactors are pointed out, such as Reactor 0 in Sub-Network 5 (the yellow line in Figure 80). The temperature in this reactor dips low until other reactors push fuel and air through this reactor in the correct proportions. As iterations occur, this proportion gets closer to stoichiomentric at around $t = 5ms$. After more iterations occur, fuel stops being sent to this reactor, and the temperature drops since the reactor contains mainly air. The same thing happens in Reactor 0 in Sub-Network 4 (purple line in Figure 80). Even the reactors which always have both fuel and air at proportions conducive to burning still oscillate in temperature until the steady-state values are achieved. These temperature changes occur for the same reasons as above, as different amounts of fuel, air, and products enter these reactors as the mass flow rates change. This explains the fluctuations in the blue, red, and black lines (Reactor 1 in Sub-Network 3, Reactor 2 in Sub-Network 3, and Reactor 3 in Sub-Network 3, respectively).

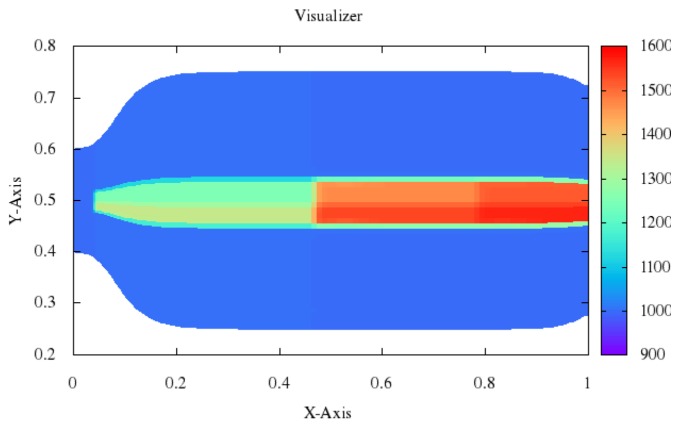
The run with $\partial f = 1$ (Figure 79(a)) converges the fastest, as it only uses eight reactors. When the reactor count is doubled (sixteen reactors when $\partial f = 0.9$ in Figure 79(b)), the physical time to convergence is a bit over twice as long. Adding eight more reactors (tewnty-four reactors when $\partial f = 0.9$ in Figure 79(c)) again increases the time until convergence.



(a) $\partial f = 1$

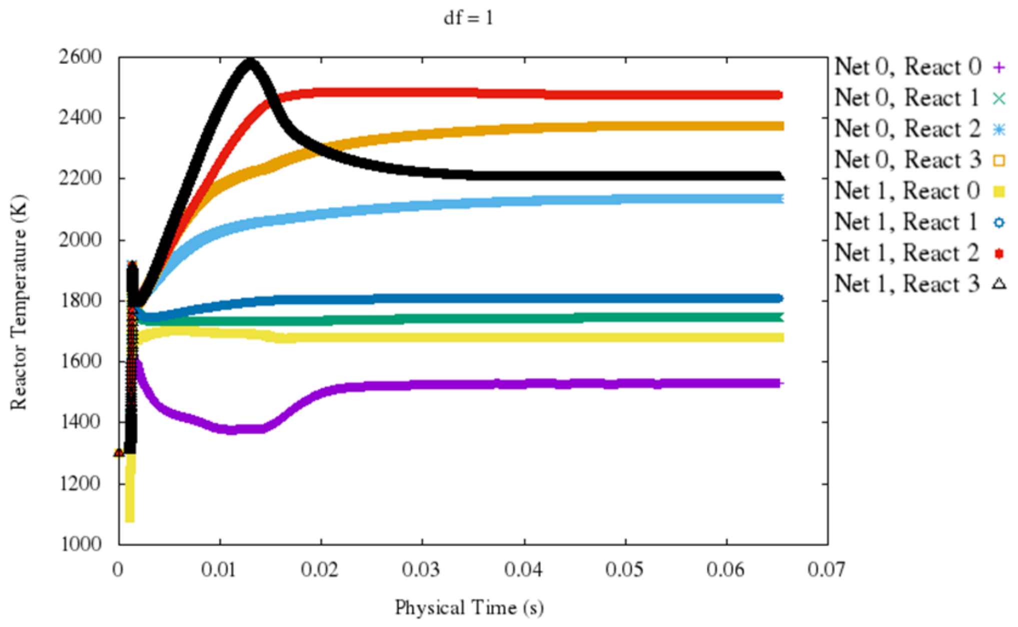


(b) $\partial f = 0.9$

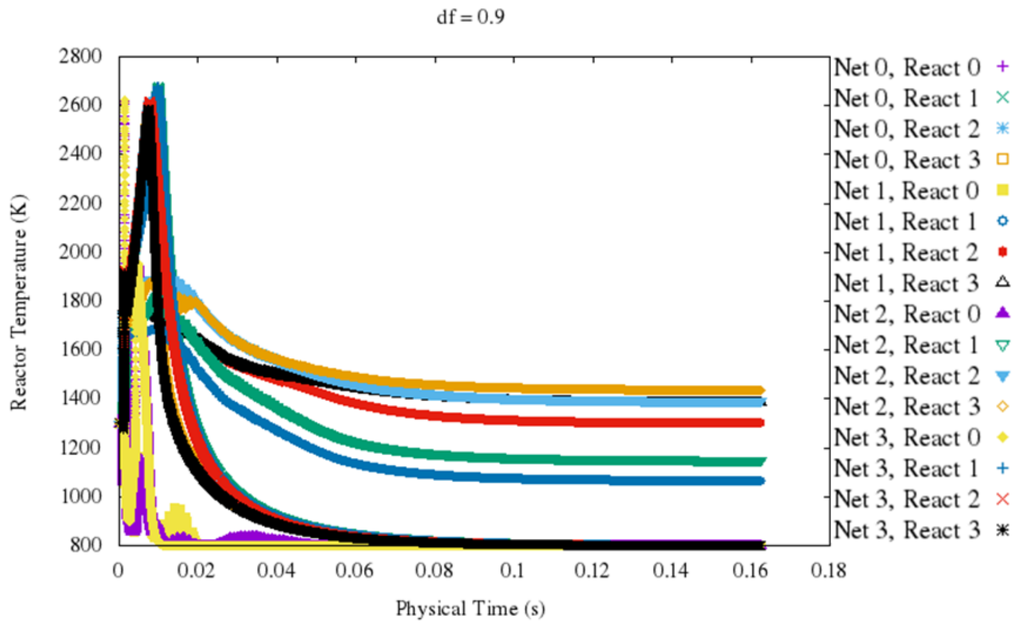


(c) $\partial f = 0.8$

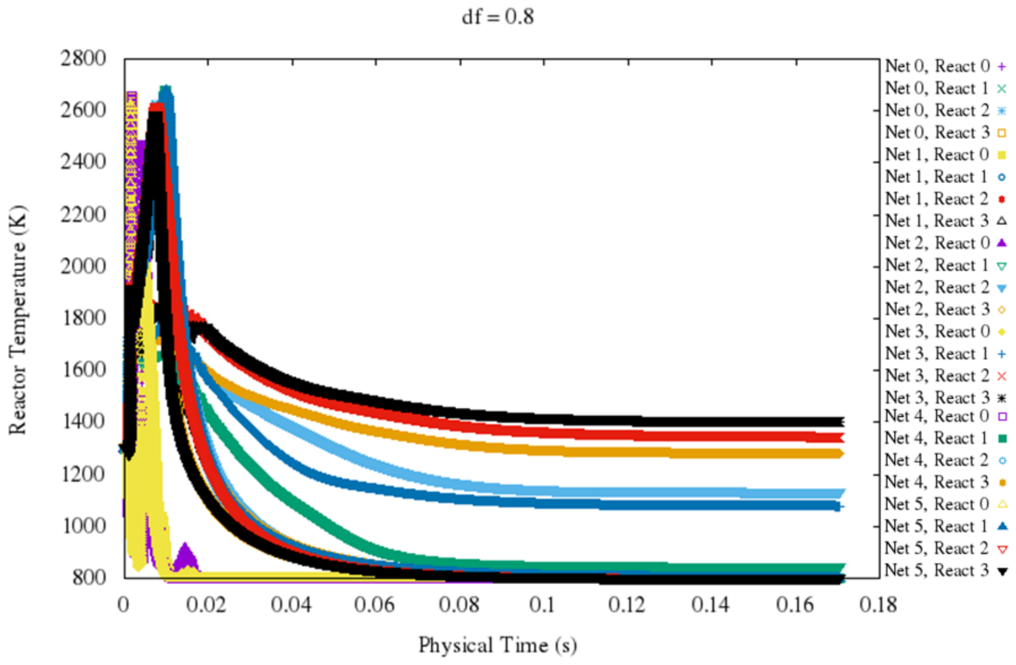
Figure 78: Converged temperature contours using the MoST algorithm with varying mixture fraction thresholds. In these simulations, $T_3 = 1000K$ and the grid size is 110x50 nodes. As the mixture fraction gradient threshold changes, the equivalence ratio for each kinetics cell also changes, which causes the reactors to burn at different temperatures.



(a) $\partial f = 1$



(b) $\partial f = 0.9$



(c) $\partial f = 0.8$

Figure 79: Reactor temperatures as a function of physical time for $T_3 = 800K$ simulations and varying mixture fraction gradient thresholds. The grid size is 110x50 nodes. The reactor numbers are not labeled due to the large number of reactors.

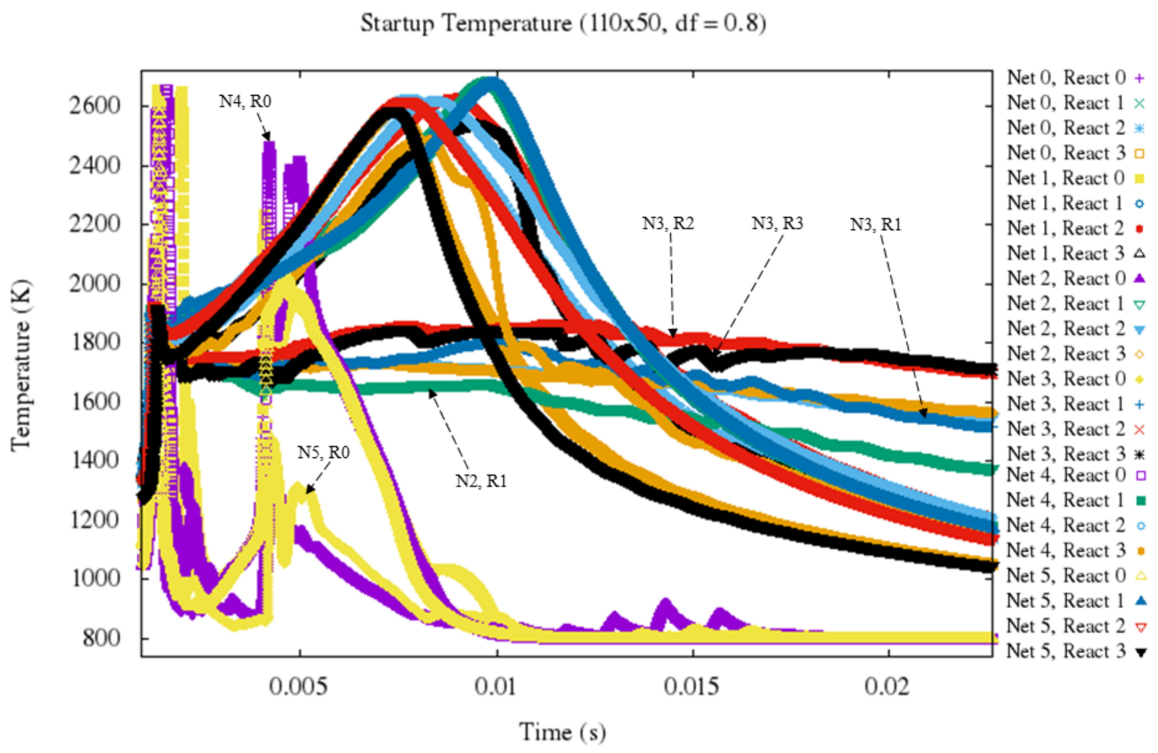


Figure 80: Start up reactor temperature profiles for the 110x50 grid node run with $T_3 = 800K$ and $\partial f = 0.8$.

5.2.2.1 CFD and CKN Mesh Fidelity Study

The topic of accuracy vs. CFD cell density and kinetics cell density was mentioned in previous chapters. It was hypothesized (Research Hypothesis II in Section 3.3) that a lower fidelity grid with a higher order NOx approximation predicts NOx better than the combination of a high fidelity grid with a low order NOx approximation. The lowest fidelity grid (37x17 nodes) was neglected in previous sections due to the fact that the RFCFD solution predicts NOx which is orders of magnitudes higher than the other meshes (which are more dense). The 37x17 grid is a relatively low fidelity grid, so this may be used to perform tests with a high fidelity chemical mechanism. Specifically, the 37x17 grid is used along with the GRI 3.0 mechanism to predict NOx with varying mixture fraction gradient thresholds. The results are shown in Figure 81.

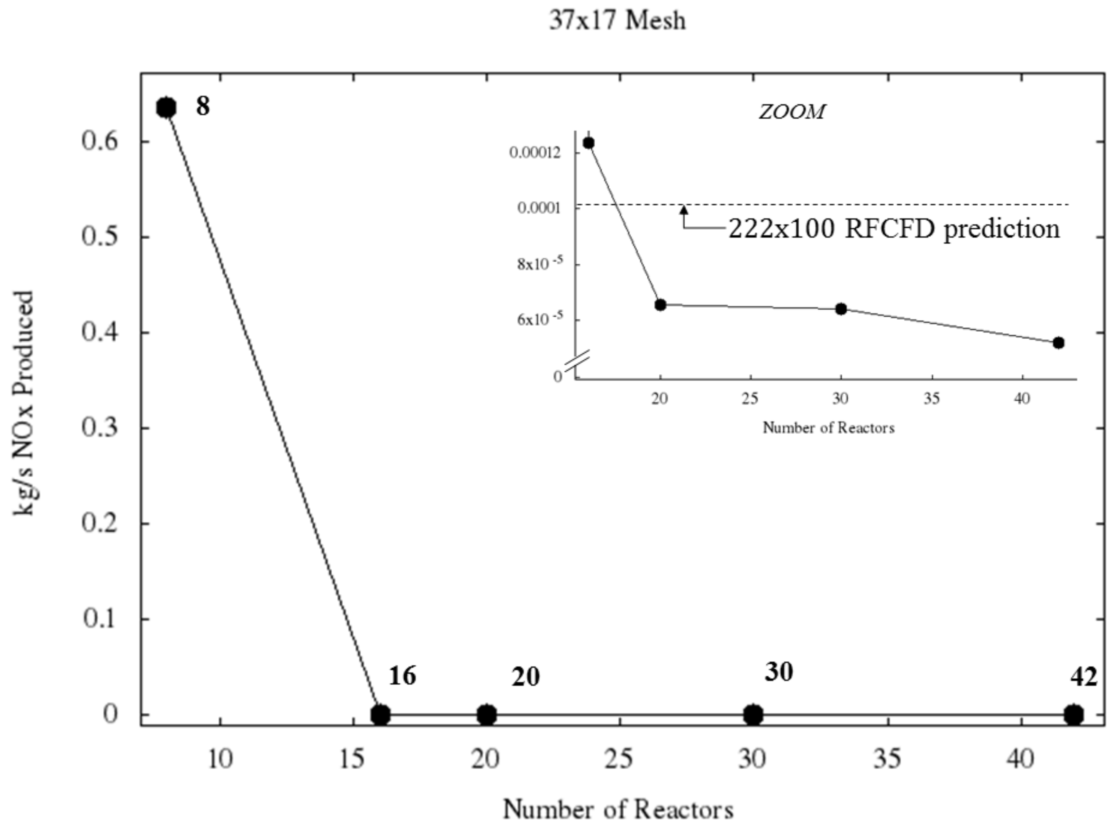


Figure 81: Comparison of the NOx results for a RFCFD PPDF model using a partial equilibrium assumption on the 222x100 grid against the MoST algorithm using varying number of reactors on a low fidelity (37x17 node) grid. Both the RFCFD and MoST runs use an inlet temperature of $T_3 = 800K$. The GRI 3.0 mechanism is used in the chemical reactors.

The results are actually quite good for such a coarse grid, and perform better than the RFCFD solution at the same CFD grid fidelity. The mixture fraction gradients ($\partial f = 0.99999$ down to $\partial f = 0.15$, which created 16 to 42 reactors) give much closer NOx values to the high fidelity RFCFD solution (222x100 node grid) than the low fidelity RFCFD solution (37x17 node grid), the results of which are two orders of magnitude away. This also gives an indication of NOx sensitivity to number of kinetics zones in this regime. It is difficult to make general statements about how the flow field behaves with different mixture fraction gradient thresholds (since it is purposefully kept high for computational speed). As stated above, the steady state temperature between each reactor might increase again if cells are divided into kinetics zones with fuel/air ratios conducive to higher temperatures due to their proximity to stoichiometric equivalence ratios.

5.3 Summary

It is evident that the CFD mesh fidelity plays a large role in the flow field determination and NOx emissions prediction using the MoST. The resolution of the flow field (especially mixture fraction) is a key parameter when determining where to place reactors. However, the biggest impact is brought in with the specification of the mixture fraction gradient threshold, as this directly specifies which cells belong to which reactors. It is important to note that these two are very closely connected. If the CFD field is not able to resolve the mixture fraction well, then the gradients in the flow field will not be sharply defined, and therefore will not be split into the correct reactors.

Note that the kinetics zone temperatures are lower than the maximum temperature shown for the RFCFD solution, so the NOx production rate in any given reactor is lower than that in the high temperature RFCFD cells. However, the flow rate out of a given reactor is higher than out of any given RFCFD cell. The idea here is that these two effects average out, and a decent approximation for NOx is obtained by averaging kinetics properties across cells. As a matter of fact, very close NOx values are obtained compared to those simulations which are run with RFCFD, with one quarter (77x35 nodes) to one half (110x50 nodes) the amount of CFD cells (Figure 77), and without directly coupling all of the reacting flow equations together. More kinetics cells may be necessary in order to correctly capture flame temperature (depending on the architecture), but the MoST algorithm gives comparatively good NOx results.

CHAPTER VI

CONCLUSION

Chapter 1 introduces conventional methods for combustor design using correlations. Here, the theoretical concepts behind NOx emissions formation are presented, as well as the connection between different NOx formation mechanisms in different aerothermodynamic regimes. Chapter 2 discusses previous approaches which have been developed to rapidly approximate emissions. As stated, these techniques either rely on data, involve a calibration step, or are too costly and time consuming to run in DOEs for rapid trades and assessments in PD. Chapter 3 confers the goals of this work; to develop a technique which gives the combustor designer a starting point for novel architectures, where correlations have not yet been obtained. As hardware tests are time consuming and expensive, a computational approach is desired. Additionally, the manufacturer may not have the luxury of relying on standard emissions formation mechanisms (i.e. rely solely on predicting thermal NOx), and standard reacting flow CFD is unable to efficiently handle large chemical mechanisms. Chapter 4 describes the approach for the development of a new computational method which indirectly couples the kinetics and flow equations. This description includes an examination of attempts and other methods which do not meet the performance criteria. Most importantly, this chapter gives the exact details pertaining to the final form of the MoST algorithm. The simplicity involved in extending this method to other gaseous emissions is also displayed, in addition to the flexibility of the method to handle steady and unsteady flows, and the difference between the proposed algorithm and other methods.

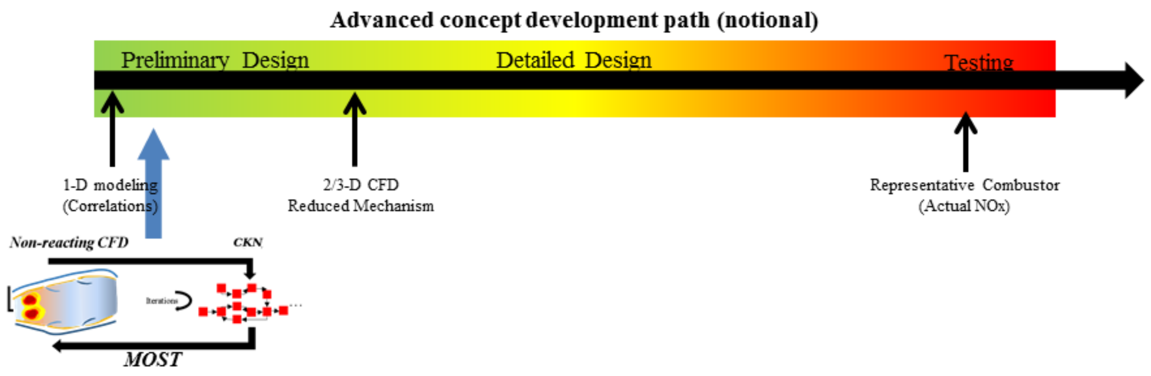
Chapter 5 shows the results of the work. It displays the method's ability to correctly match trends and NOx emissions over a range of CFD mesh sizes and number of kinetics zones. Research Hypothesis I is verified using the results of Research Experiments I (Section 3.3), since an algorithm which applies flow field filters to determine where reactions occur was developed and tested. It was successfully used to predict the correct trends with high accuracy (based on the R^2 values). It continuously updates the reactors based on the flow field so that changes to the flow field caused by reactions are taken into consideration. Though preliminary sizing correlations are not known for the fictitious combustion chamber used during experimentation, the NOx emissions are on the same order as a moderately high fidelity RFCFD run using a PPDF combustion model and partial equilibrium approximation for thermal NOx. It is difficult to determine what size mechanism is necessary for accurate emissions approximations, since the GRI 3.0 mechanism was the only one used, as this is an extremely popular and trusted mechanism. Therefore, the second part of Research Hypothesis I remains inconclusive due to a lack of data. However, this experiment can very well be carried out with the addition of another NOx mechanism with fewer or more species, and sensitivities can be obtained. Section 6.1 addresses computational speed.

Research Hypothesis II was verified in the particular cases which were tested: The low fidelity grid worked very well when determining NOx emissions, as long as the CKN is not overly sparse (i.e. more than eight reactors are used). Therefore, Research Hypothesis II is verified based on the data at hand. Additionally, mixture fraction is the only flow field parameter necessary to use as a filter, as constant pressure PSRs perform well to predict emissions based solely on differences in equivalence ratio. The density changes within each reactor due to temperature changes, and this is reflected within the flow field, as well. Additionally, incompressible flow is a popular assumption, and the density change here *only* occurs due to temperature changes. To conclude, the CFD mesh does have an impact on the ability to predict emissions. A higher fidelity mesh allows one to capture mixture fraction better, and ought to be used when available. However, it is important to use a high fidelity mesh when a low mixture fraction gradient is used, otherwise resources are wasted when splitting the CFD grid into numerous zones since the mixture fraction is not resolved enough.

Research Hypothesis III is accepted based on the obtained data. Though only the results from a combustor can with jets were presented here, this method was tested with other geometries (e.g. flame tubes, cans with no jets, etc.), and performed well. The combustor can presented in this work was tested over numerous inlet conditions, and gave results which agreed with theoretical and RFCFD predictions. The conclusions deduced from Experiment III are tightly coupled to those in Experiment II, since the number of reactors used should be a function of how well the mixture fraction is resolved in order to not waste resources.

The main purpose of this work is to complement CFD and testing when correlations are not available for the design of advanced concept combustors. Conventional design of new combustors involves using correlations, however, Figure 64 shows they perform poorly when used for different concepts. To reduce the amount of loop iterations (Figure 12), the MoST can be used in place of and to augment correlations and test data, as shown in Figure 82. This allows one to reduce the amount of loop iterations during design, saving time and money. It provides an alternative to relying solely on the incorrect correlations (or expensive test data) for a very different combustor architecture.

Additionally, this method proves particularly useful in many other situations. For example, the PPDF and FGM models involve a lookup table for thermodynamic properties during simulation. These data may not exist for non-basic fuels such as alternative fuel combustors or futuristic catalytic combustors where the fuel and air interact with other species that are not normally present in the atmosphere. These reaction schemes may be very involved and could be unrealistic to run using reacting flow CFD. However, this can be done using the MoST method as long as the chemical mechanism is known. The MoST allows one to run large mechanisms by indirectly coupling kinetics equations. It takes into account flow field changes due to temperature, in addition to minor species influence on the flow field.



Advanced concept development path (notional)
Preliminary Design
Detailed Design
Testing
1-D modeling (Correlations)
2/3-D CFD Reduced Mechanism
Representative Combustor (Actual NOx)
Non-reacting CFD
CKN
Iterations
MOST

Figure 82: The MoST algorithm is presented in a notional design path. It can be used to augment correlations when they are not available for designs which are much different.

This is the first iteration in the development of the MoST algorithm, and therefore, there are many ways to improve upon this method. This work focused on feasibility of the method and proof of concept, and many advances can be made to make the method mature. The main target for improvement involves saving even more computational time while losing minimal accuracy (See Section 6.2).

6.1 Remarks on Timing

A major portion of this work involves the development of a speedy method for PD by helping expedite the design process of combustion chambers and determining emissions. An important question regarding timing remains, since the point is to save time and money in the preliminary design phases of a combustor concept. RFCFD methods commonly (if not always) involve implicit (and typically steady) algorithms to obtain solutions. The CFD code developed for this work is a strictly explicit unsteady method, making timing comparisons very difficult. As discussed in Section 6.2, the runtime is highly dependent on the CFD grid, number of reactors, valve coefficients, etc. With this being said, the MoST technique in its current implementation for this work runs on the order of tens of minutes to hours for the 110x50 cell grid using an *explicit* CFD time stepping method an tens of reactors[1]. Note that neither this flow field nor reactor network are preconditioned at all, which greatly expedites the solution process. Depending on the aforementioned parameters, the runs may take anywhere from under a minute to days (under a minute for sparse grids, sparse reactor networks, and small chemical mechanisms, to days for extremely dense grids, many hundreds to thousands of reactors, and large kinetics mechanisms). Though is difficult to say how much an implicit CFD solver method may accelerate the MoST algorithm, implicit methods allow one to take much larger time steps while maintaining stability (See Section 6.2).

With this being said, it is very difficult to compare the solution time for the STAR-CCM+ RFCFD solution with that for the MoST algorithm, as STAR-CCM+ does not offer an explicit solver along with the other models of interest. As a matter of fact, running similar algorithms in STAR-CCM+ (non-reacting flow using an AUSM type explicit time stepping method on the same grid) runs much slower (in some cases, orders of magnitudes slower) than the code developed in Section 4.1.2. This is common, since the code in this work was developed to do one very specific task very quickly, and industrial solvers calculate solutions to a wide variety of problems using general techniques. This makes the subject of timing a very difficult topic to discuss, and future works hould be carried out before conclusions can be made on temporal comparisons.

[1]This is using the GRI 3.0 mechanism while writing out reactor properties and species concentrations at each solver iteration.

6.2 Recommendations for Further Development

Many algorithms besides those which made it into the final method in Section 4.3 were developed as part of this work. This is a direct result of the fact that these types of filtering methods for this purpose do not exist, so experimentation was performed (Research Experiment I) to devise a scheme which accomplished the goal. Though not all of these algorithms proved useful for the MoST methodology as it is presented above, there are many reasons why they are still important. First, it is important to know what works and what does not work, as this research may be continued in the future. Second, the algorithms developed as part of this work may be useful in other branches of mathematics, science, and engineering. Lastly and most importantly, new ideas may be explored using these techniques; especially when looking to create the next generation of the MoST algorithm. Ideas for how to improve different aspects and extend the MoST algorithm are presented here, in addition to some supplemental ideas and preliminary work on these advancements which are presented in the Appendix, Chapter E.

This work primarily focuses on proof of concept of the MoST algorithm using numerical experiments. The methodology was tested using a notional combustor can (which incorporates a varying shape and dilution jets) to compare the MoST algorithm against CFD predictions. Though this is a decent approach, there are ways to improve the analysis of the results, specifically comparison with test data obtained from physical hardware tests. These comparisons were not carried out since test data are typically proprietary. Although this is the case, certain openly available test data do exist, however these geometries tend to be quite simple (e.g. a standard flame tube). To avoid comparisons with only the simplest cases (and to show the utility of the method with intricate geometries and flow features), this notional can is used for comparison. Future work must include comparison with experimental data by meshing the geometry of a test can and making direct comparisons.

As stated previously, running this method involves using a kinetics solver, of which any may be selected. The Cantera solver may act very stiff if the correct coefficients are not specified for the valves which sit between reactors, as these terms may have vastly different orders of magnitude compared to species terms. Though these valves do not add any extra equations, they do add terms to the ODE system which increases the time for a solution. Specifically, having these valve coefficients too low may lead to unwarranted flow through the valves, yet having them too high increases the chance of stiff equations. Numerical experiments should be performed to determine good valve coefficients and rules should be developed for particular systems. This may be unnecessary if using a different kinetics solver (since valves may not exist), however different kinetics solvers may behave similarly so caution must be used. On the other side of this is the CFD solver. The CFD code developed here (Section 4.1) is a very simple one indeed, since it is only an explicit unsteady solver. Implicit solvers tend to converge must faster and save a lot of time during simulation. The same argument can be made for a steady state reactor network. Details about these future steady state solvers are given in Chapter E.

Additionally, it should be noted that only methane was used in this work, and the optimum valve coefficients may change if different species are present. When looking at aircraft gas turbine combustors, Jet-A should be used during simulation. The next extension of this work is to obtain a good kinetics mechanism for Jet-A and perform tests using the MoST. Other advancements may be made to the code, including extension to three dimensions (Section E.2), optimizing the iteration scheme between the non-reacting CFD solution and the CKN (Section E.3), and extending the MoST algorithm to utilize an unstructured grid (Section E.5). The use of an unstructured grid is important, as using one may save time and may also decrease the number of reactors even more. An unstructured grid may require the addition of more algorithms, such as vortex tracking and sizing algorithms in order to determine how to connect reactors. Preliminary work on these methods are shown in Section E.6.

APPENDIX A

GRID GENERATION TECHNIQUES

The governing equations for grid relaxation using the Laplace method can be written as [38]

$$A^n_{1_{i,j}} X^{n+1}_{\xi\xi_{i,j}} - 2A^n_{2_{i,j}} X^{n+1}_{\xi\eta_{i,j}} + A^n_{3_{i,j}} X^{n+1}_{\eta\eta_{i,j}} = 0 \tag{55}$$

and

$$A^n_{1_{i,j}} Y^{n+1}_{\xi\xi_{i,j}} - 2A^n_{2_{i,j}} Y^{n+1}_{\xi\eta_{i,j}} + A^n_{3_{i,j}} Y^{n+1}_{\eta\eta_{i,j}} = 0. \tag{56}$$

The value of the As are

$$A_1 = x_\eta^2 + y_\eta^2, \tag{57}$$

$$A_2 = x_\xi x_\eta + y_\xi y_\eta \tag{58}$$

and

$$A_3 = x_\xi^2 + y_\xi^2. \tag{59}$$

The governing equations for grid relaxation using Poission's equation can be written as [38]

$$A^n_{1_{i,j}}(X^{n+1}_{\xi\xi_{i,j}} + \phi^n_{i,j}X^{n+1}_{\xi_{i,j}}) - 2A^n_{2_{i,j}}X^{n+1}_{\xi\eta_{i,j}} + A^n_{3_{i,j}}(X^{n+1}_{\eta\eta_{i,j}} + \psi^n_{i,j}X^{n+1}_{\eta_{i,j}}) = 0 \tag{60}$$

and

$$A^n_{1_{i,j}}(Y^{n+1}_{\xi\xi_{i,j}} + \phi^n_{i,j}Y^{n+1}_{\xi_{i,j}}) - 2A^n_{2_{i,j}}Y^{n+1}_{\xi\eta_{i,j}} + A^n_{3_{i,j}}(Y^{n+1}_{\eta\eta_{i,j}} + \psi^n_{i,j}Y^{n+1}_{\eta_{i,j}}) = 0. \tag{61}$$

Here, $\phi(i,j)$ and $\psi(i,j)$ are computed on the boundaries and found on the interior points by linear interpolation.

Equations 60 and 61 can be discritized and rearranged into a favorable form for a line Gauss-Seidel iterative method to be used in order to solve the equations, just as it was in Laplace's equations. Making more substitutions,

$$b^n_{i,j}X^{n+1}_{i-1,j} + d^n_{i,j}X^{n+1}_{i,j} + a^n_{i,j}X^{n+1}_{i+1,j} = c \tag{62}$$

and

$$\beta^n_{i,j}Y^{n+1}_{i-1,j} + \delta^n_{i,j}Y^{n+1}_{i,j} + \alpha^n_{i,j}Y^{n+1}_{i+1,j} = \gamma \tag{63}$$

With coefficients b, d, a, c, which are given by

$$b^n_{i,j} = \frac{A^n_{1_{i,j}}}{\Delta\xi}(\frac{1}{\Delta\xi} - \frac{\phi^n_{i,j}}{2}) \tag{64}$$

$$d^n_{i,j} = -\frac{2}{\Delta\xi^2}(A^n_{1_{i,j}} + A^n_{3_{i,j}}) \tag{65}$$

$$a^n_{i,j} = \frac{A^n_{1_{i,j}}}{\Delta\xi}(\frac{1}{\Delta\xi} + \frac{\phi^n_{i,j}}{2}) \tag{66}$$

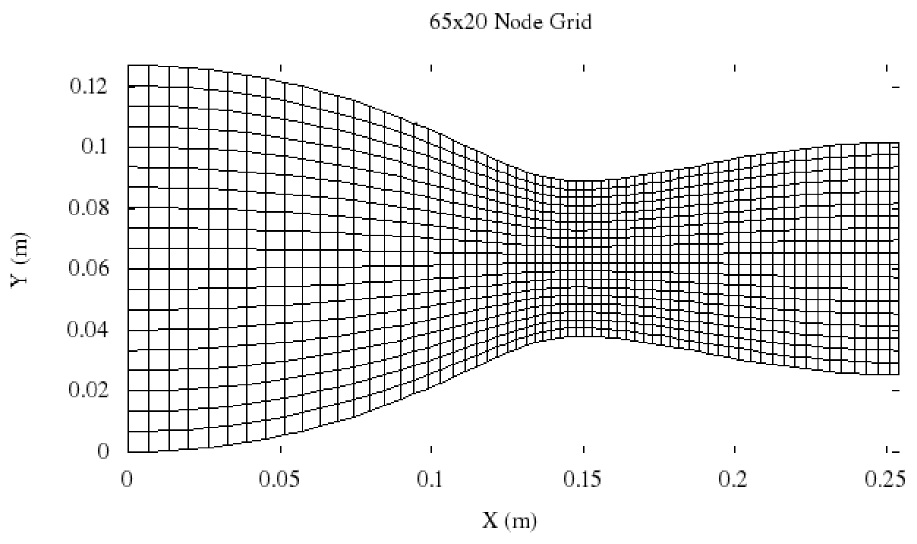
$$\begin{aligned} c \;\; = \;\; & -[(\frac{A^n_{2_{i,j}}}{2\Delta\xi\Delta\eta})(-X^n_{i+1,j+1} + X^{n+1}_{i+1,j-1} + X^n_{i-1,j+1} - X^{n+1}_{i-1,j-1}) \\ & +[\frac{A^n_{3_{i,j}}}{\Delta\xi^2} + \psi^n_{i,j}(\frac{A^n_{3_{i,j}}}{2\Delta\xi})]X^n_{i,j+1} + [\frac{A^n_{3_{i,j}}}{\Delta\xi^2} - \psi^n_{i,j}(\frac{A^n_{3_{i,j}}}{2\Delta\xi})]X^n_{i,j-1}] \end{aligned} \tag{67}$$

The coefficients β, δ, α, and γ have the same expressions, except the Xs are replaced with Ys. As can be seen, there exist $n+1$ terms on the right hand side, however they are all specified at grid point $j-1$ (hence the need for the Gauss-Seidel method). As with Laplace's equation, Equations 62 and 63 only contain three unknown terms each, which gives rise to a tridiagonal matrix.

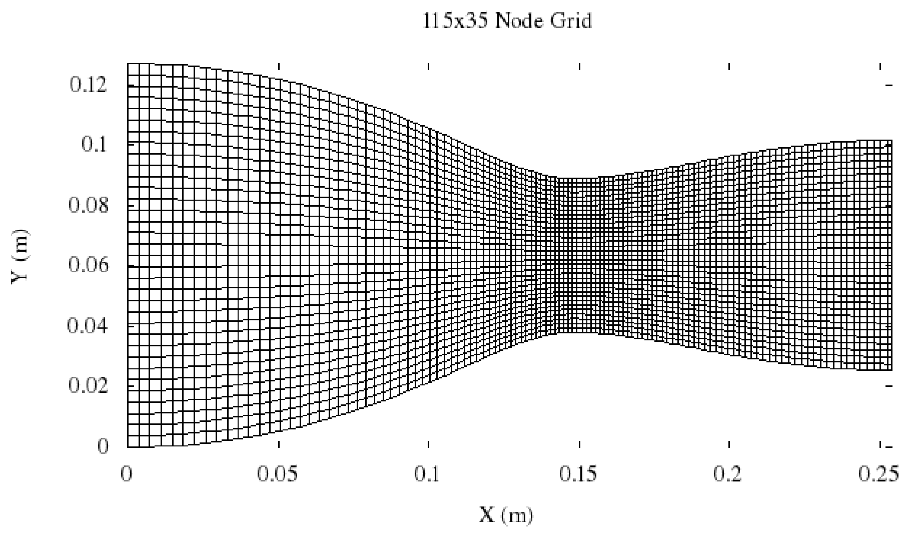
APPENDIX B

CFD VALIDATION CASES

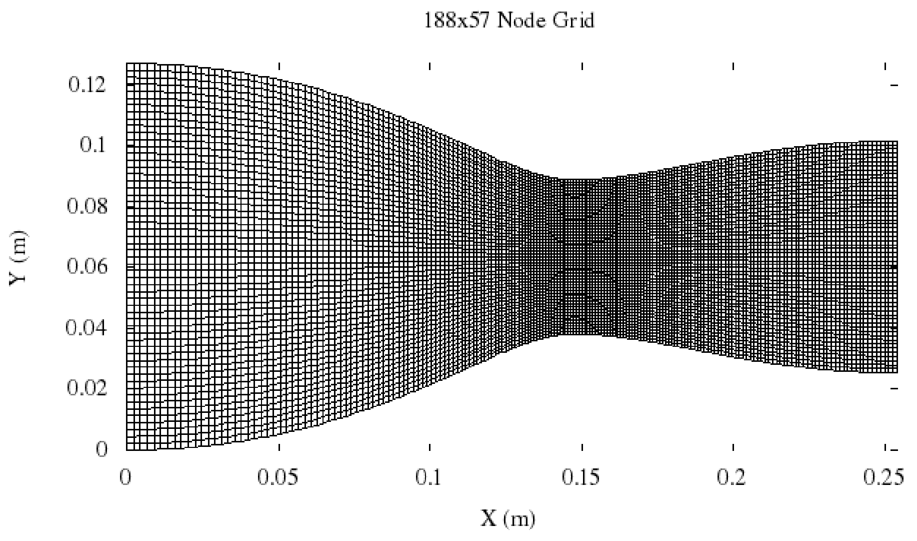
Figure 83 shows a selection of the Converging-Diverging nozzle Validation (CDV) grids. Each one represents the same CDV nozzle, but vary in fidelity from just over one thousand cells to over eight hundred thousand cells. These grids are used in Section 4.1.2.1 to compare analytic solutions to those obtained from the CFD code described in Section 4.1.2, and to perform the mass conservation study presented in the same section.



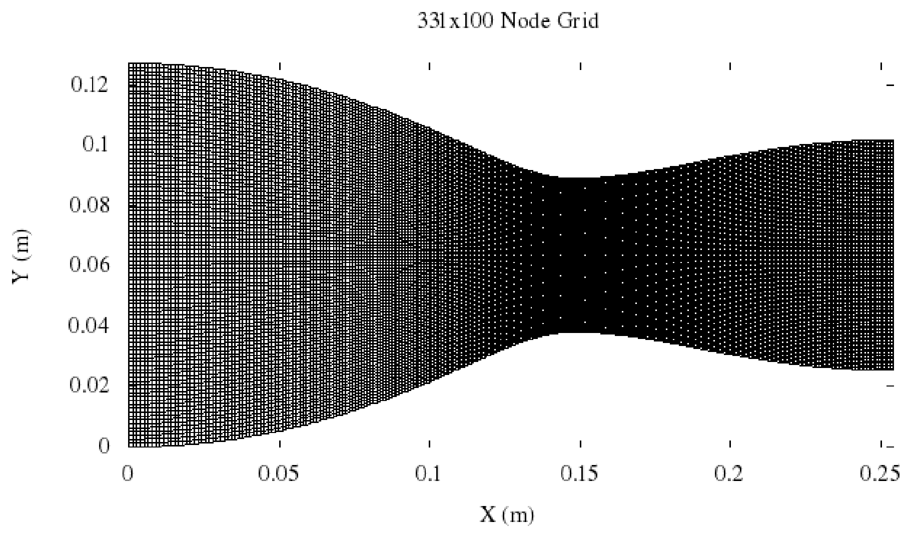
(a) Mach number contours



(b) Pressure contours



(c) Density contours



(d) Temperature contours

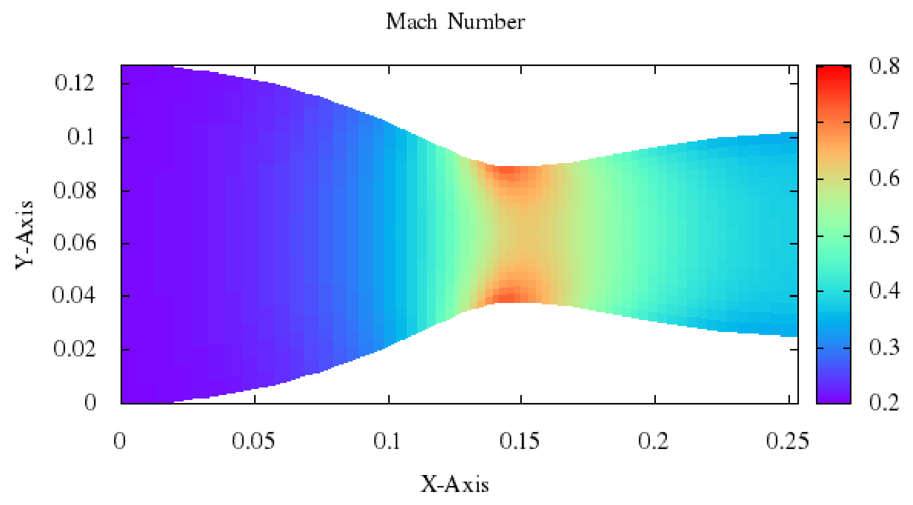
Figure 83: Grids used to run the converging-diverging nozzle validation case. Seven grids were used in total, though only four are shown since the other grids are so dense and the figures appear as a filled object. The missing grids are 668x201 (133,400 cells), 998x300 (298,103 cells), and 1665x500 (830,336 cells).

Section 4.1.2.1 displays property contours for the CDV Validation grid with 32,760 cells. However, the conservation of mass study in Figure 32 was run with seven grids. Figures 84 through 101 show contour plots of the solutions for these six other grids (from sparse to more dense, respectively) at each pressure ratio. Figures 102 through 107 show the RMS error for each of the grids at each pressure ratio, as the solution converges.

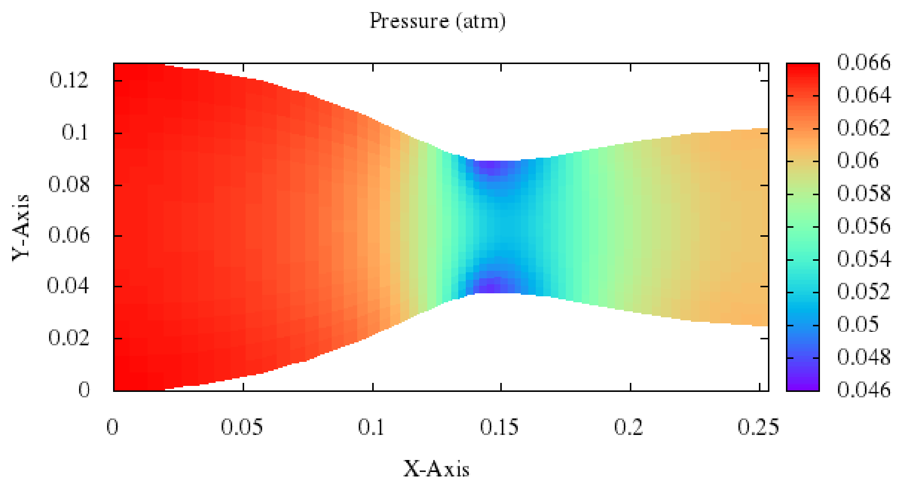
Figure 84 show the property contours for the lowest fidelity solution, where the flow is completely subsonic. Mass is conserved here to about 0.35%. The contours are not very well resolved, but this is expected for such a sparse mesh. Figure 85 show the property contours for the lowest fidelity solution with a shock. Mass is conserved here to about 0.35%, also. As with the subsonic case, the contours are not very well resolved. With the flux splitting method described in Section 4.1.2, one expects to see a very sharp shock, since both flow characteristics are taken into account during flux computations. This is not the case in Figure 85, as sparse grids do not allow the capture of sharp features very accurately. The completely supersonic case in Figure 86 has the same problems. Theoretically, the flow should become supersonic exactly at the throat. Since the grid only has about one thousand cells, the resolution suffers, and the Mach one contour is not well resolved. This contour is clearly somewhere around the minimum area, but is unable to be completely discerned.

As the resolution of the grid is changed (more cells are added to the computational mesh), the features mentioned above become better resolved. This is shown in Figures 99 through 101. The figures in between (Figures 87 through 98) are included to show how the solutions change with grid fidelity, and also so that all data and results are available. The percent difference between inflow and outflow for the highest fidelity (approximately 830,000 cells) CDV grid with subsonic conditions is approximately one thousandth of a percent (Figure 32), which is the result most important to this work since very subsonic flows are of interest. The highest fidelity case with a shock

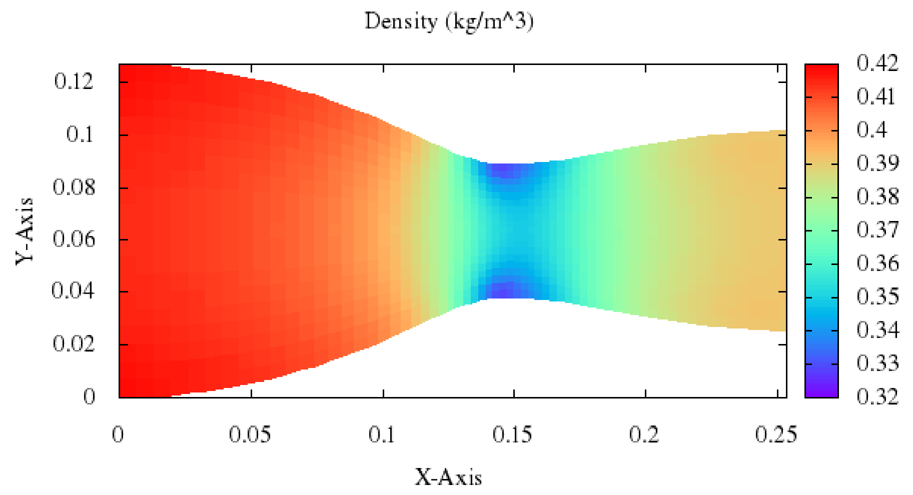
(Figure 100) is very well resolved, as a single line can be seen when the solution shocks down from supersonic to subsonic flow. Here, the mass is conserved to within a couple hundredths of a percent. The supersonic case (Figure 100) is also resolved very well. Here, the mass is conserved to within a few hundredths of a percent. Each of the plots shown below display the Mach number, density, pressure, and temperature contours for each of the six grids for all three pressure ratios.



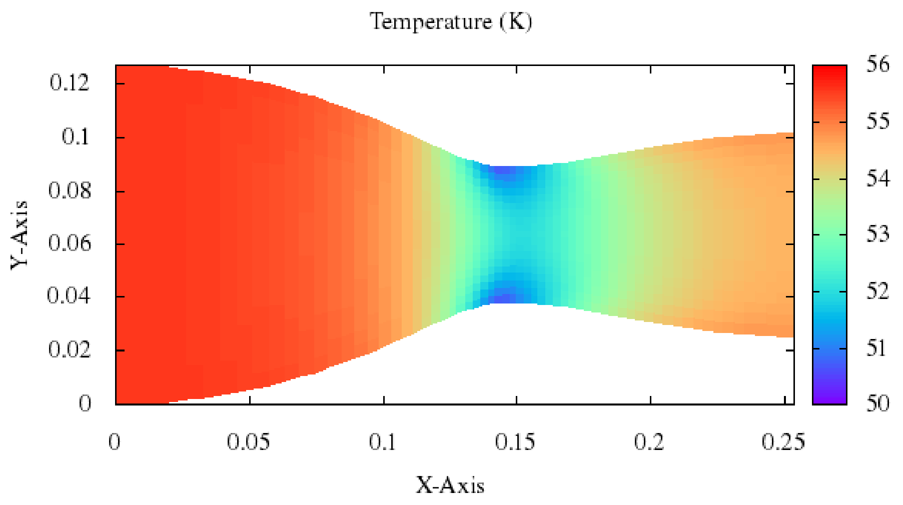
(a) 1,216 cells



(b) 3,876 cells

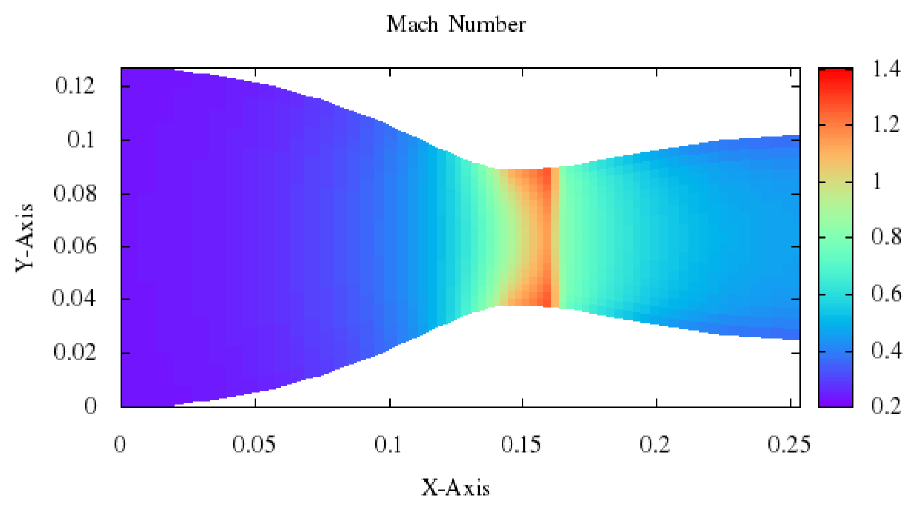


(c) 10,472 cells

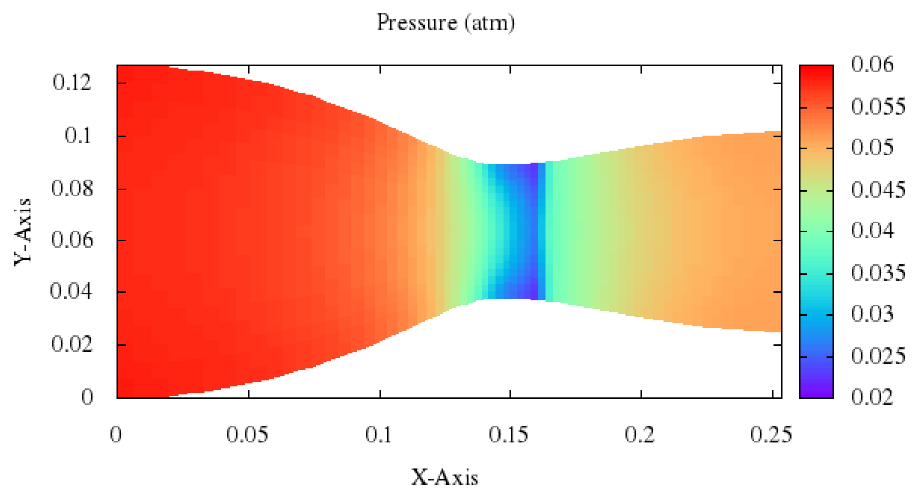


(d) 32,670 cells

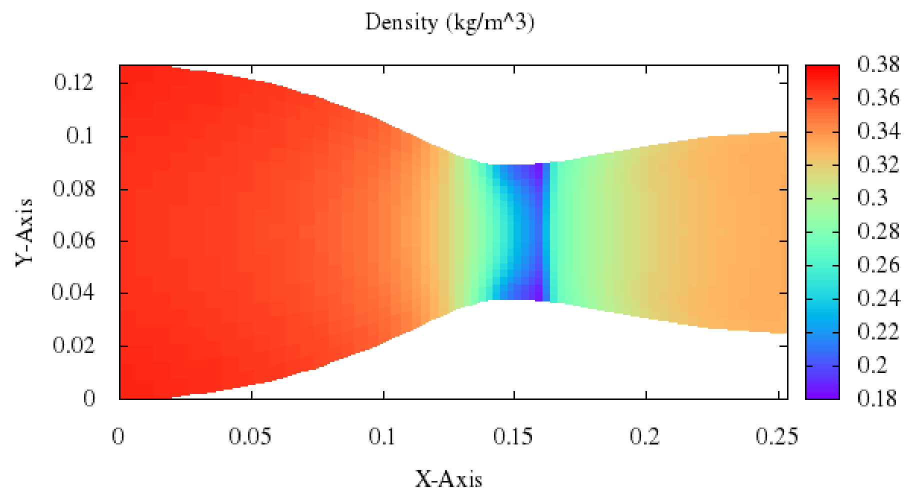
Figure 84: Property contours for converged solution with 1,216 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



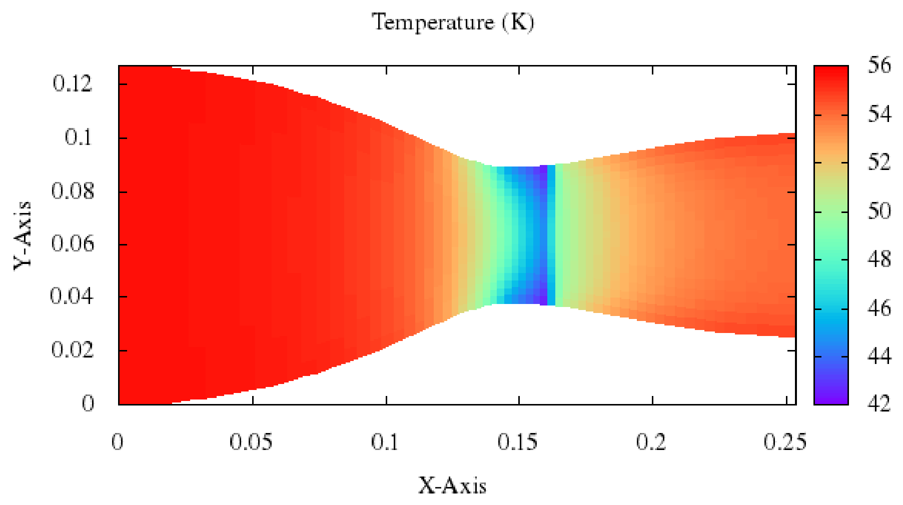
(a) Mach number contours



(b) Pressure contours

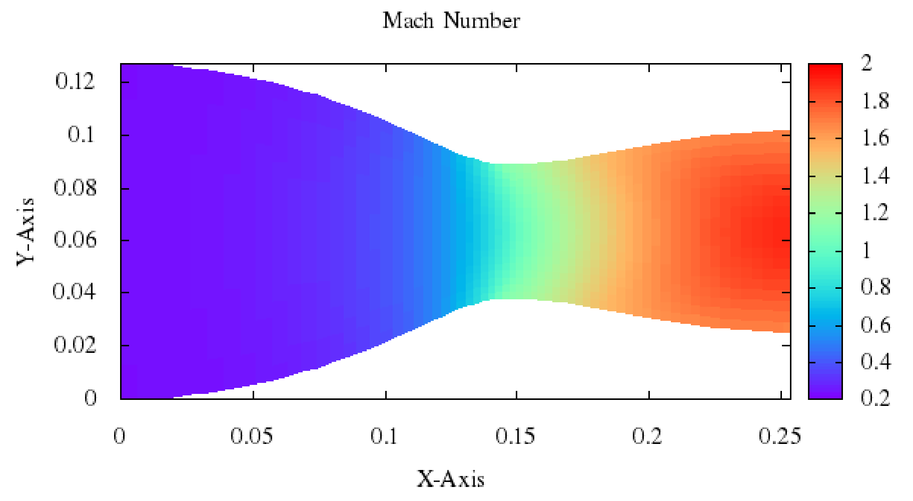


(c) Density contours

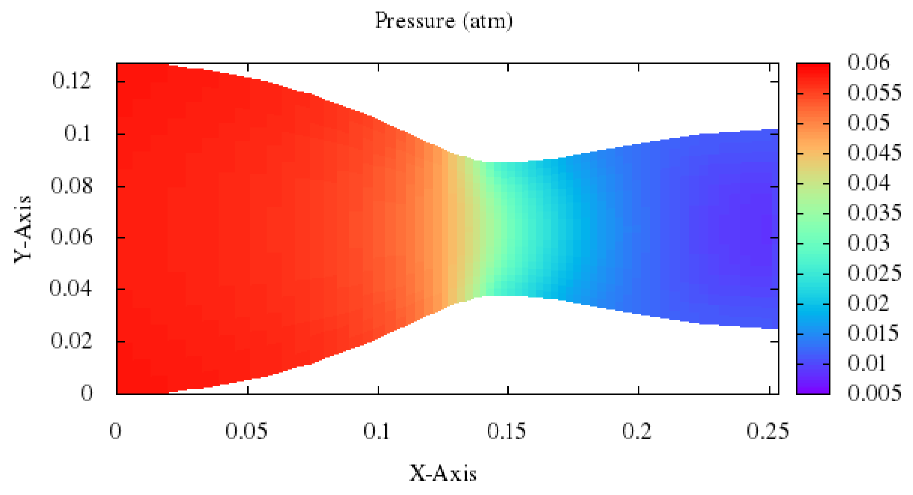


(d) Temperature contours

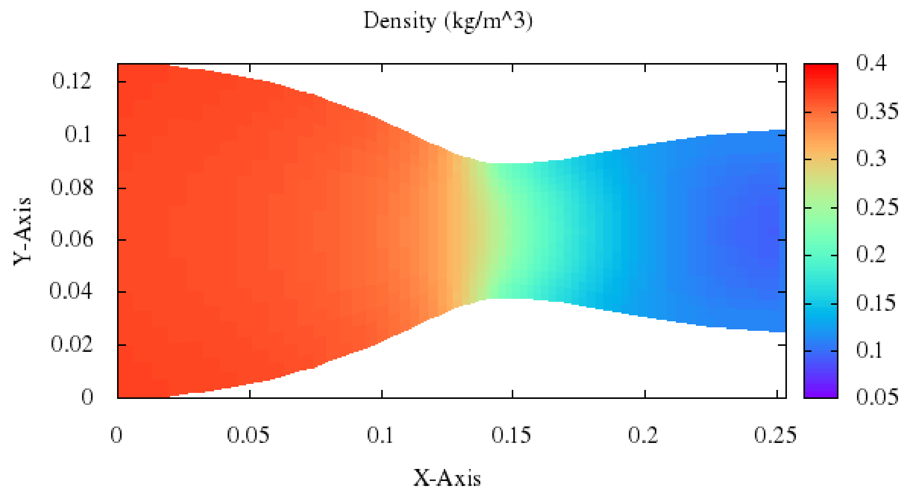
Figure 85: Property contours for converged solution with 1,216 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



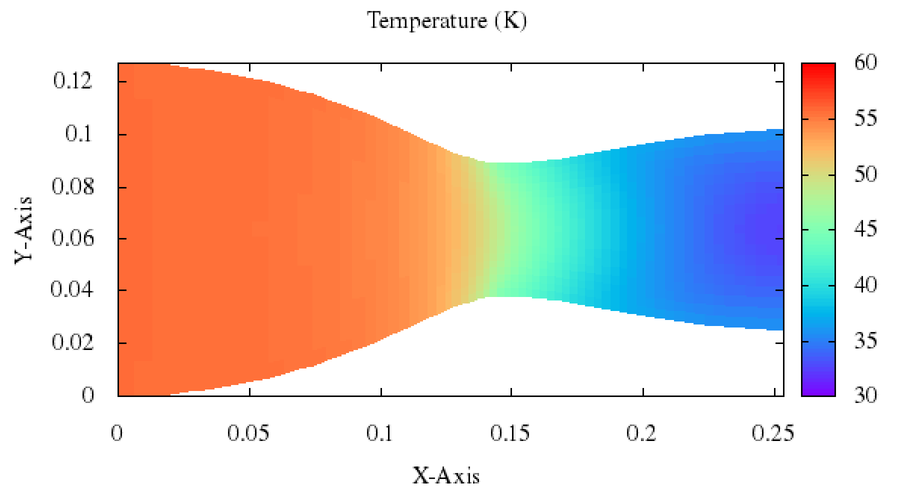
(a) Mach number contours



(b) Pressure contours

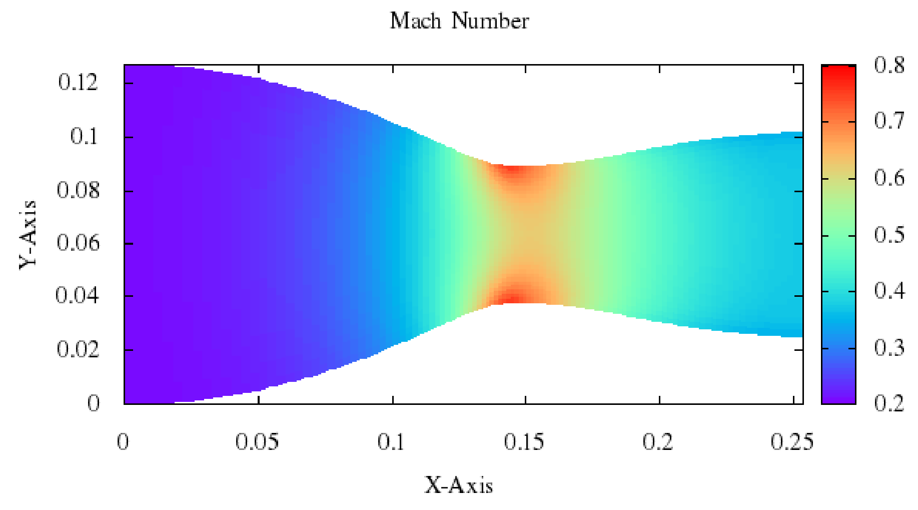


(c) Density contours

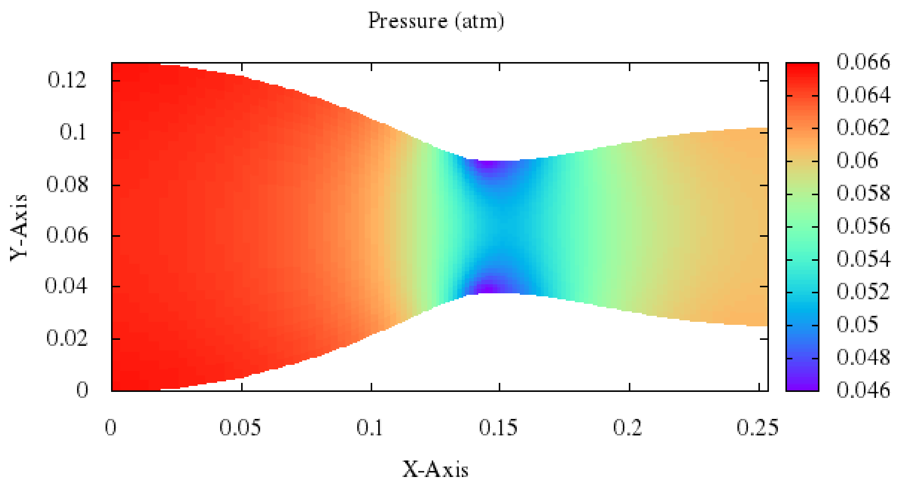


(d) Temperature contours

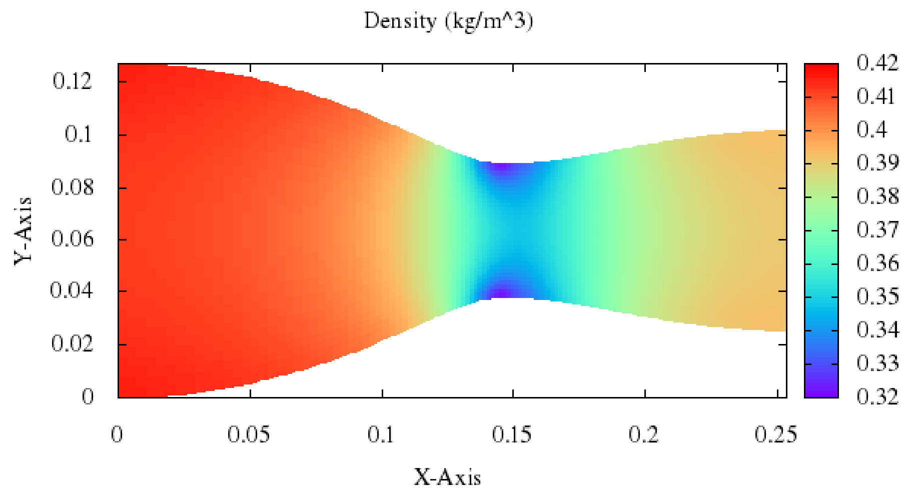
Figure 86: Property contours for converged solution with 1,216 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.



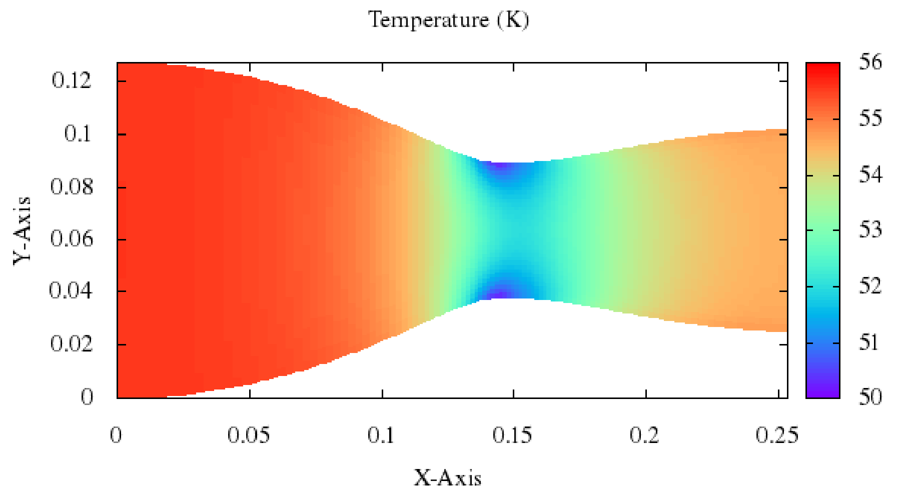
(a) Mach number contours



(b) Pressure contours

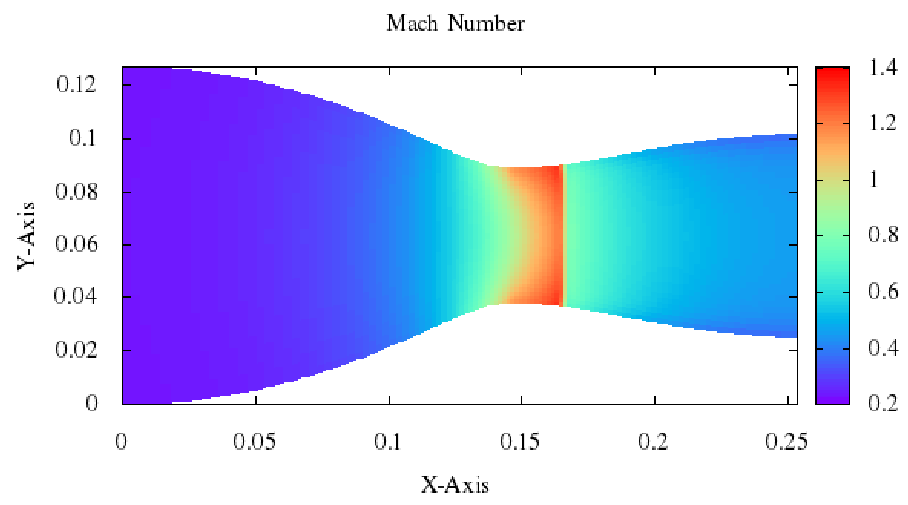


(c) Density contours

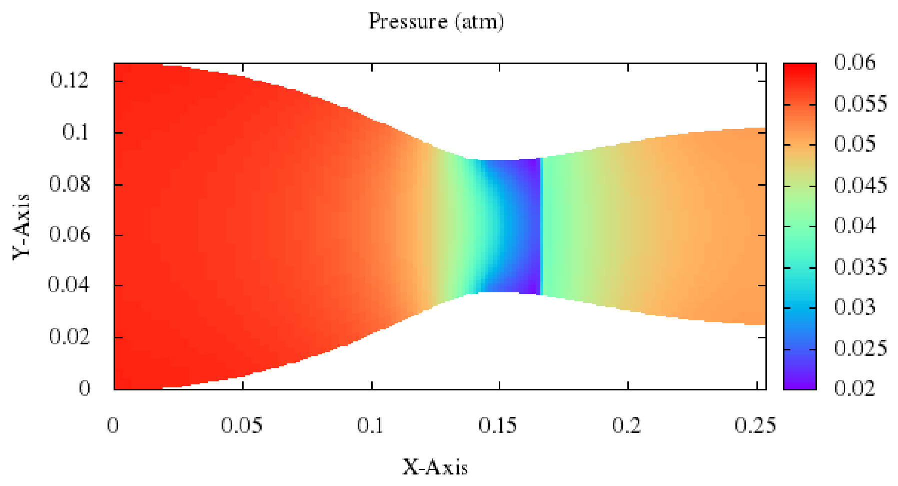


(d) Temperature contours

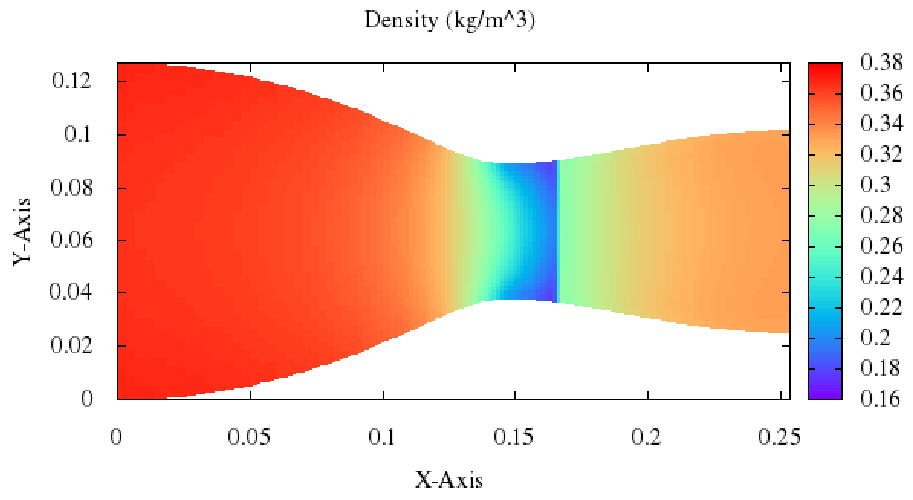
Figure 87: Property contours for converged solution with 3,876 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



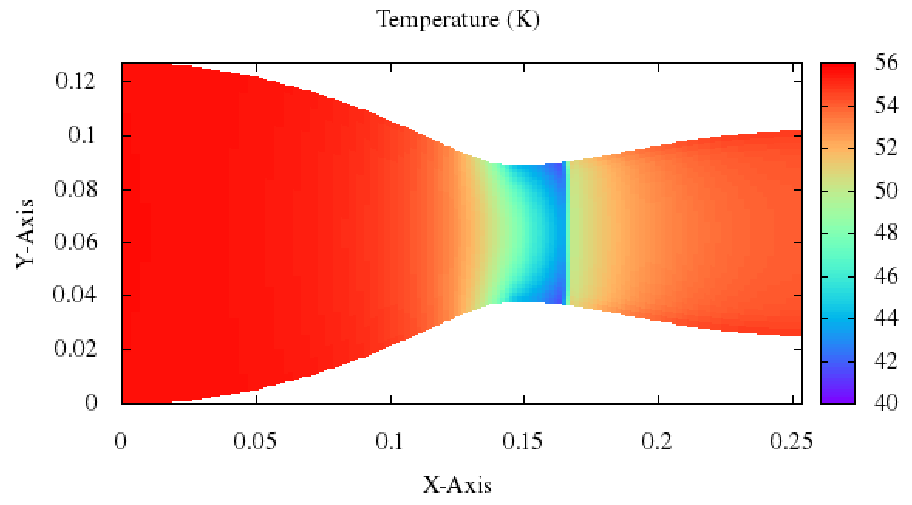
(a) Mach number contours



(b) Pressure contours

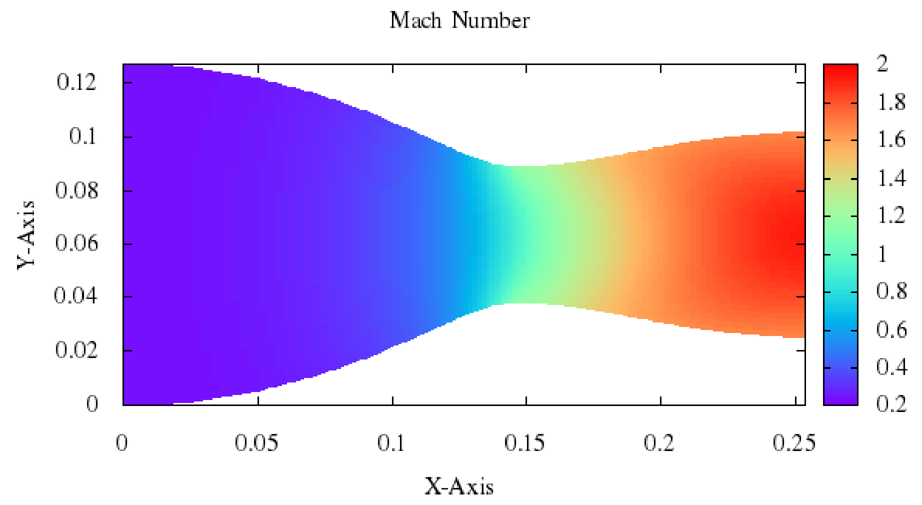


(c) Density contours

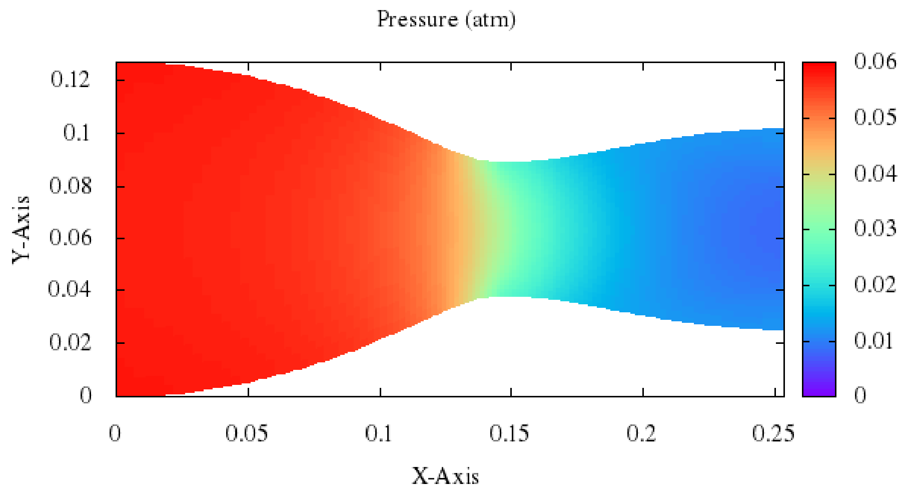


(d) Temperature contours

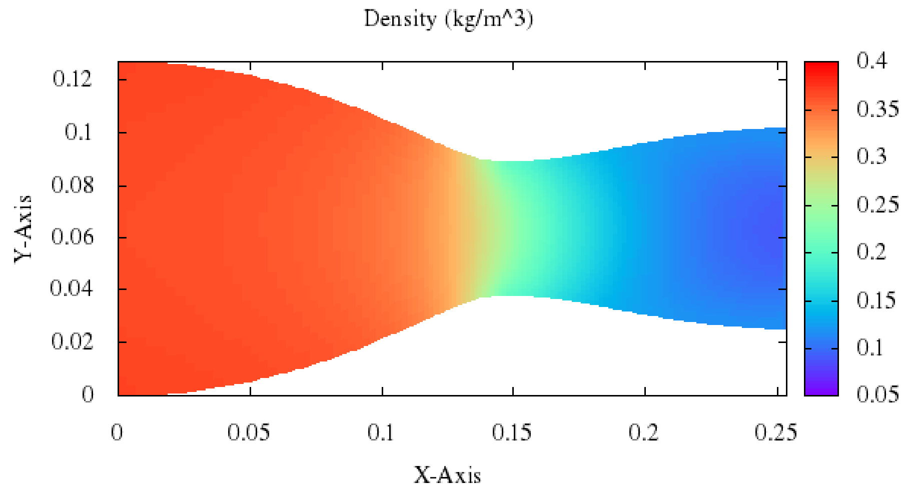
Figure 88: Property contours for converged solution with 3,876 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



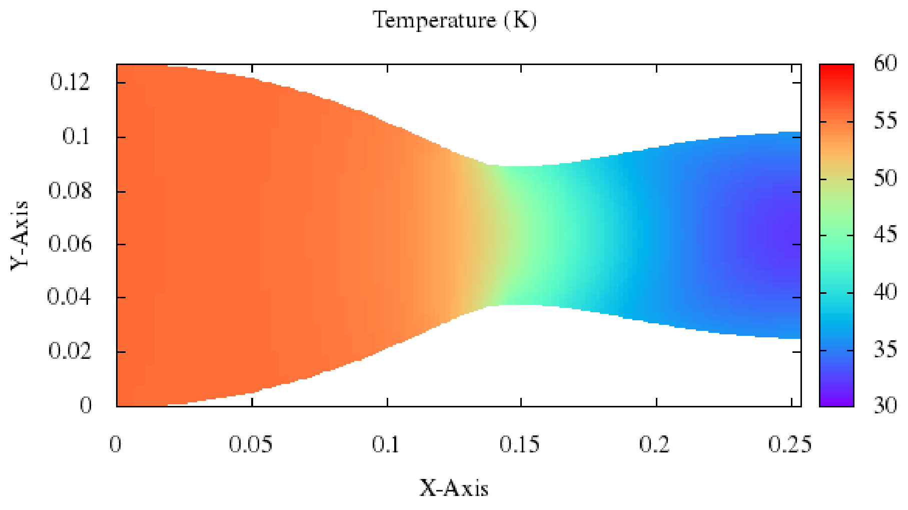
(a) Mach number contours



(b) Pressure contours

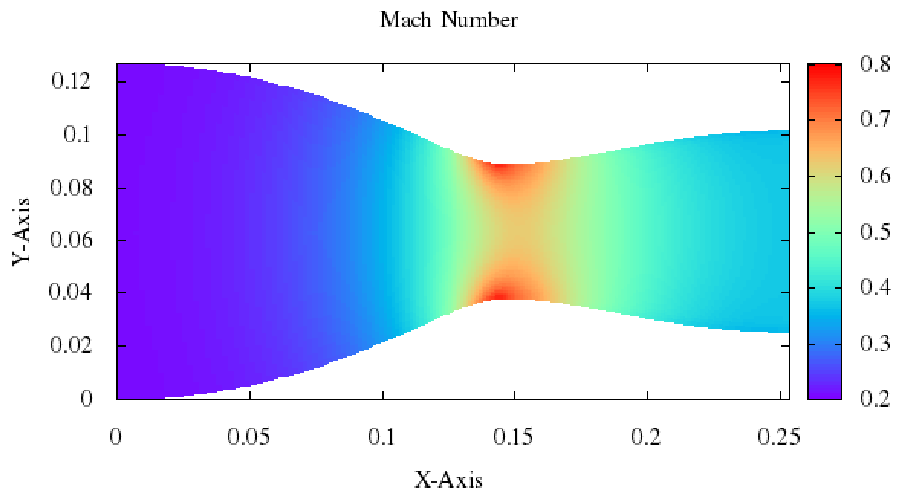


(c) Density contours

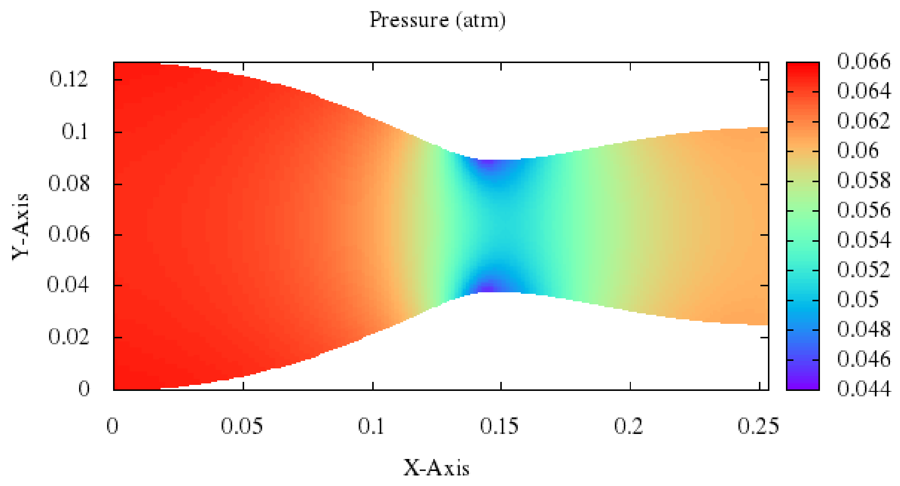


(d) Temperature contours

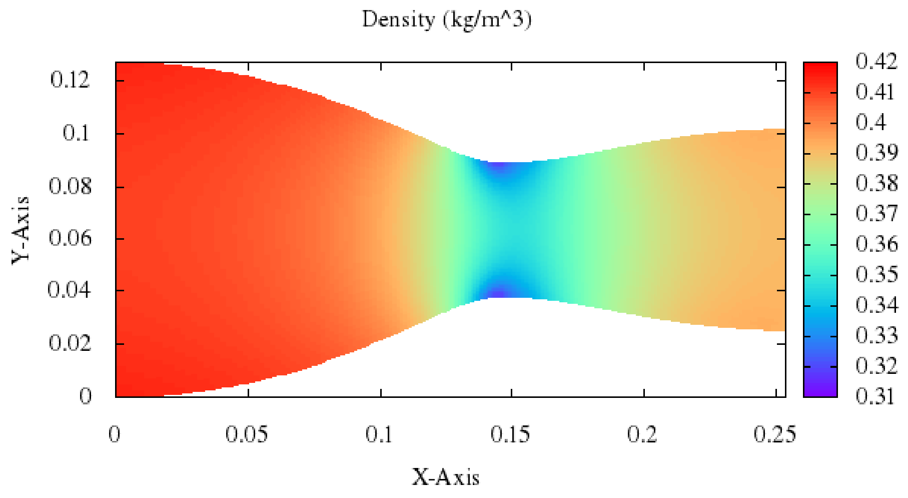
Figure 89: Property contours for converged solution with 3,876 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.



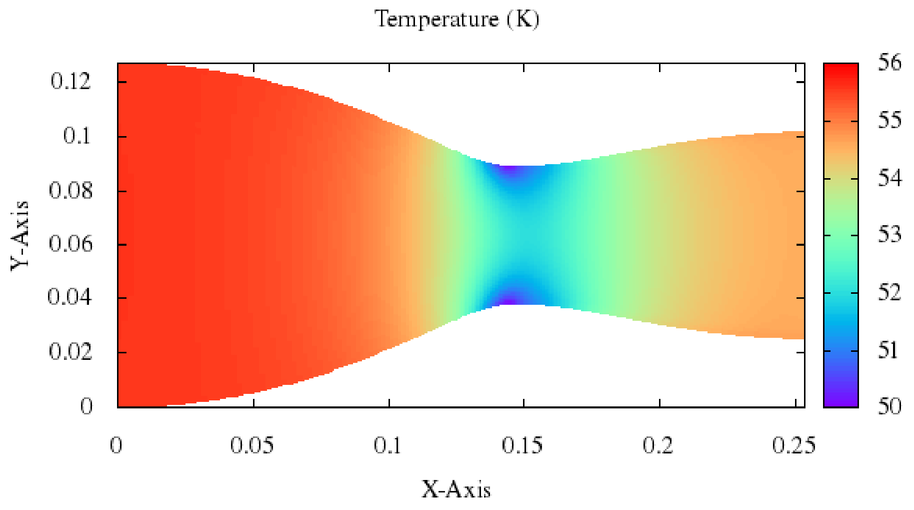
(a) Mach number contours



(b) Pressure contours

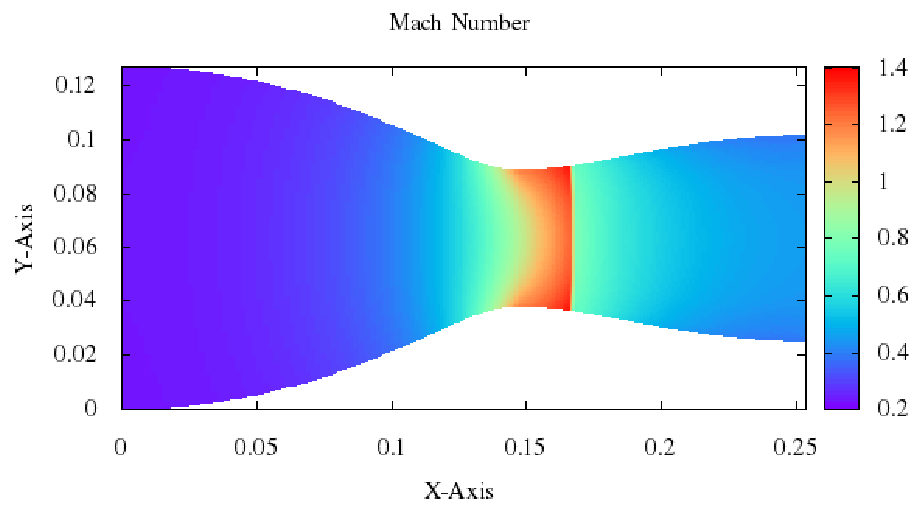


(c) Density contours

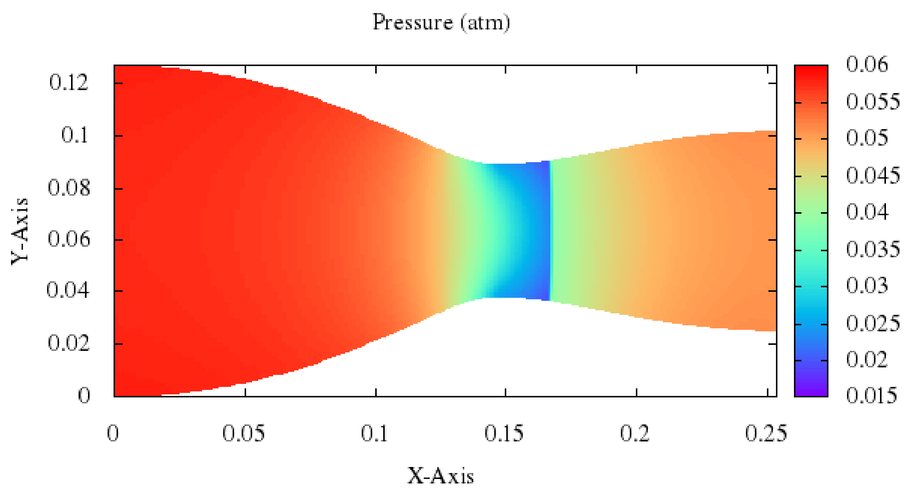


(d) Temperature contours

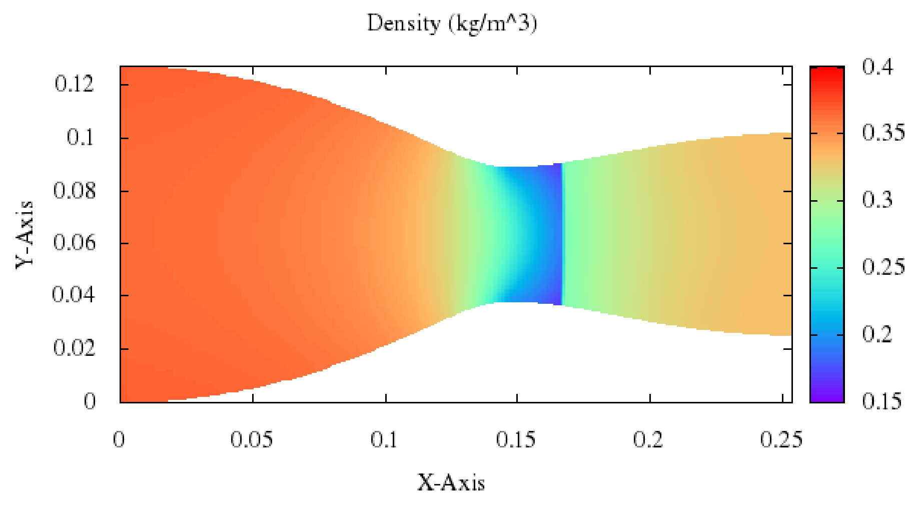
Figure 90: Property contours for converged solution with 10,472 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



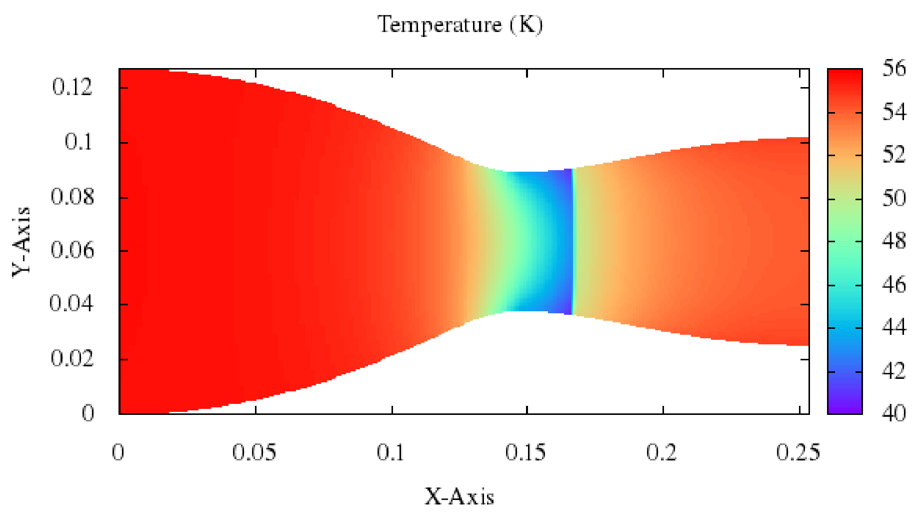
(a) Mach number contours



(b) Pressure contours

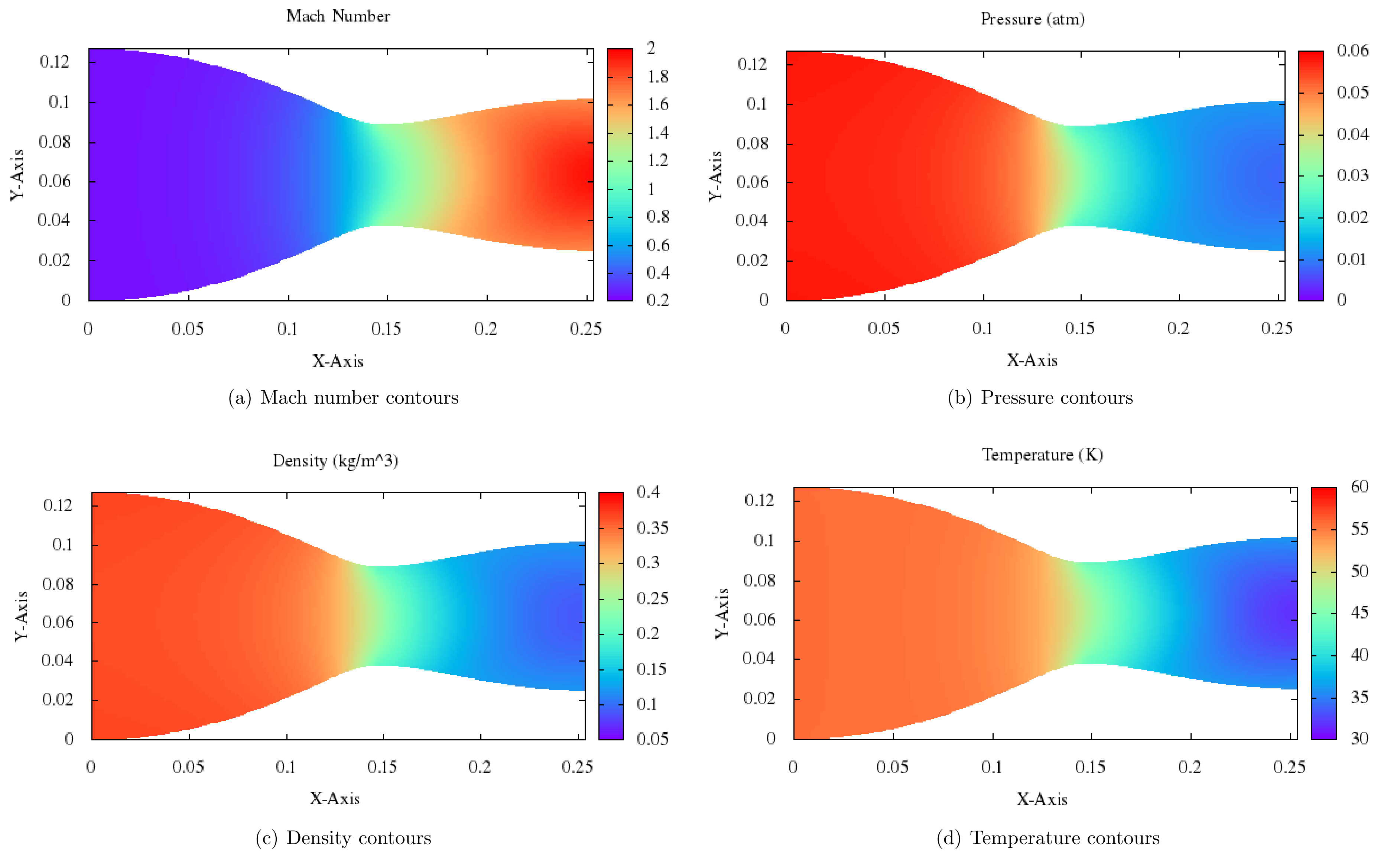


(c) Density contours



(d) Temperature contours

Figure 91: Property contours for converged solution with 10,472 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



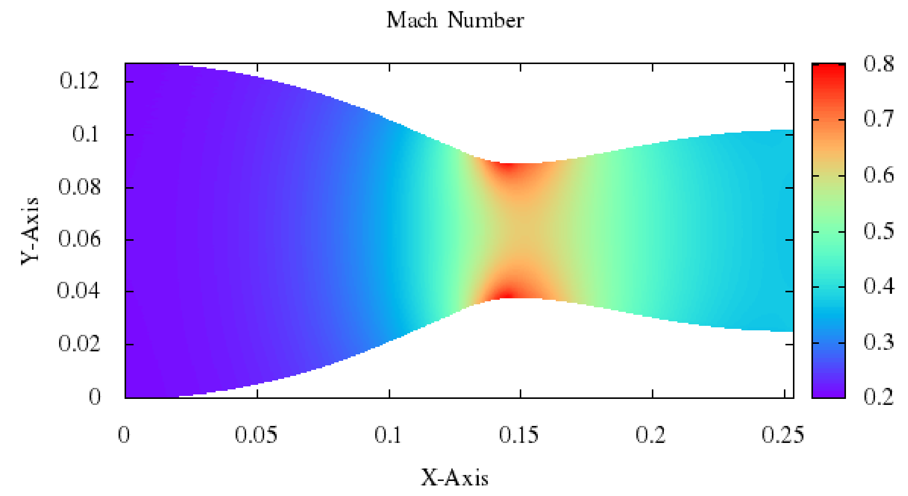
(a) Mach number contours

(b) Pressure contours

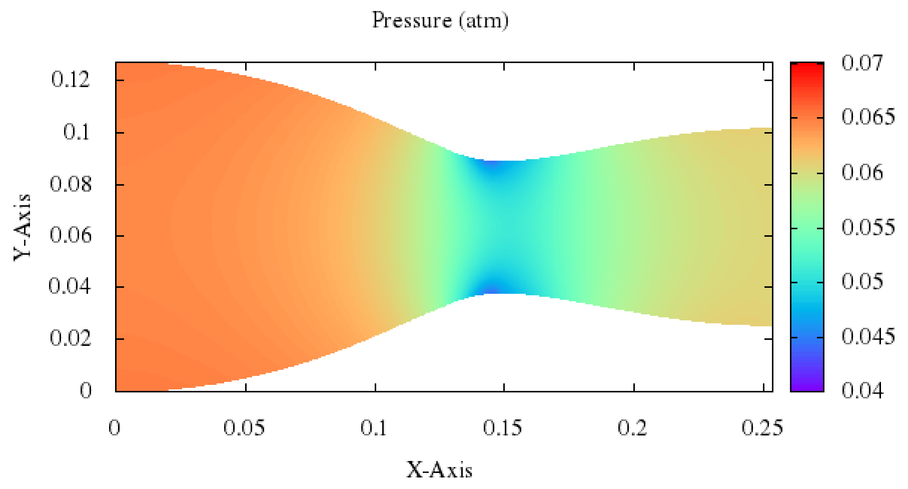
(c) Density contours

(d) Temperature contours

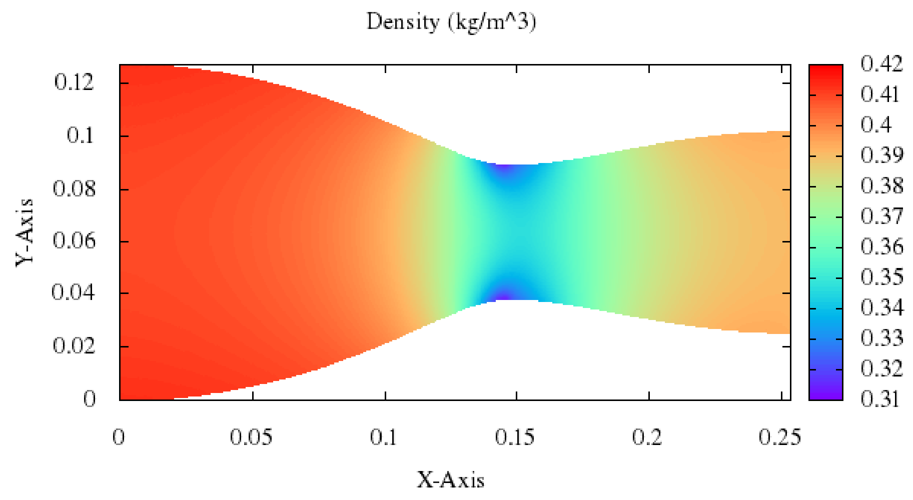
Figure 92: Property contours for converged solution with 10,472 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.



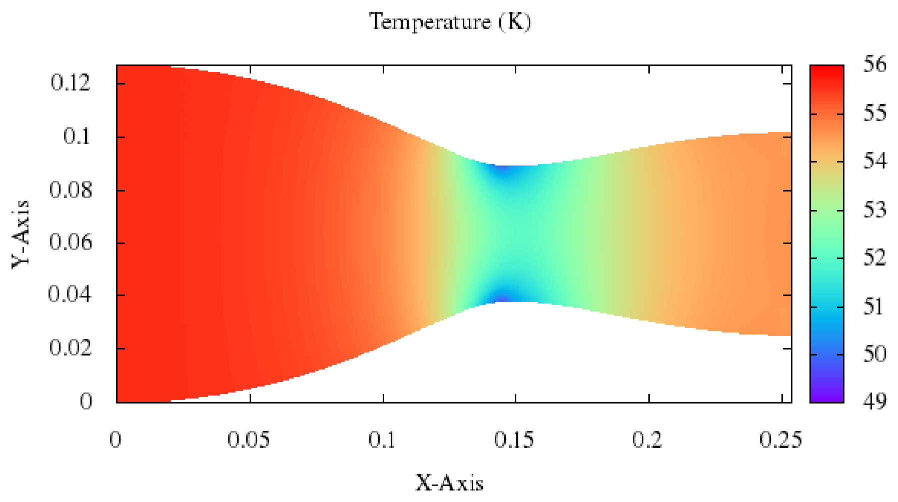
(a) Mach number contours



(b) Pressure contours

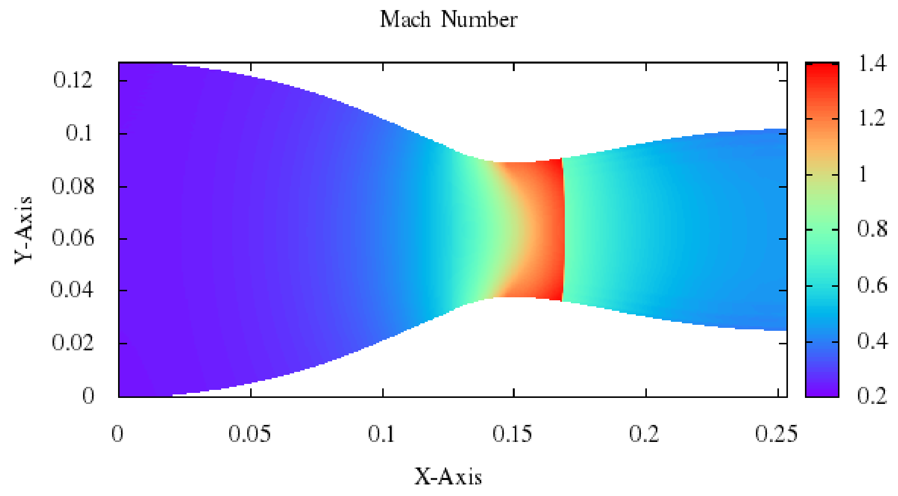


(c) Density contours

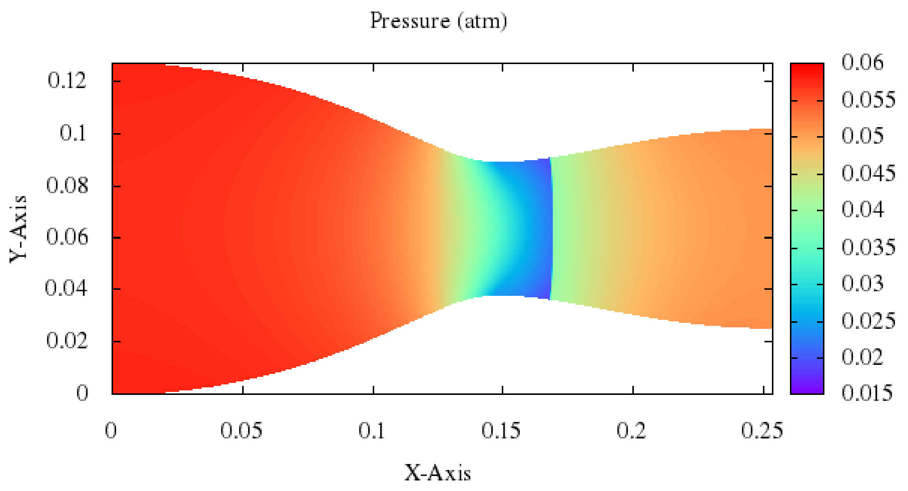


(d) Temperature contours

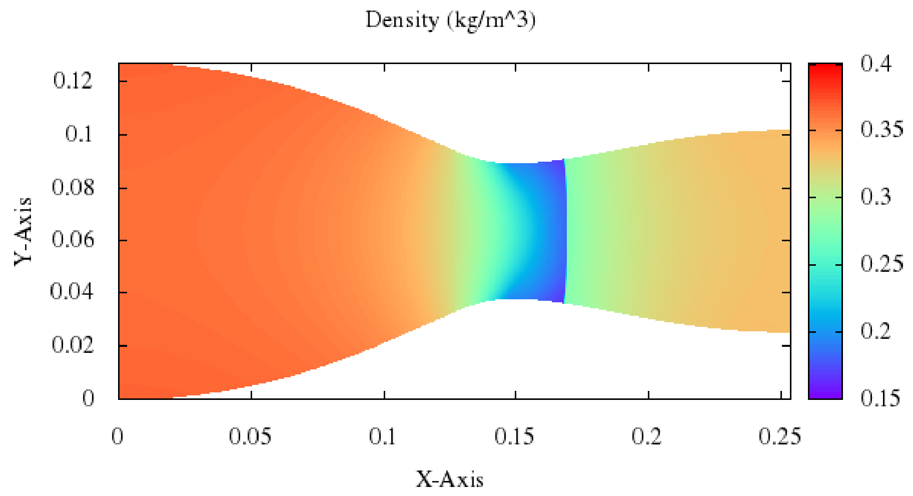
Figure 93: Property contours for converged solution with 133,400 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



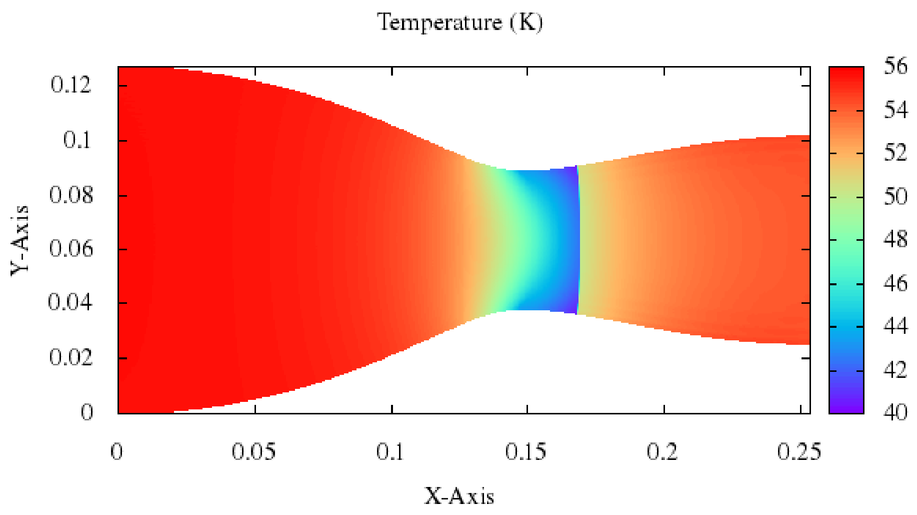
(a) Mach number contours



(b) Pressure contours

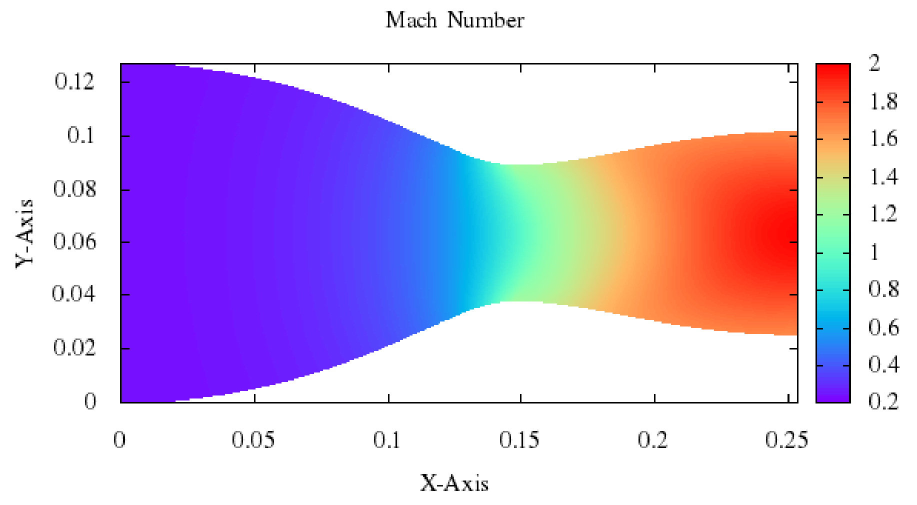


(c) Density contours

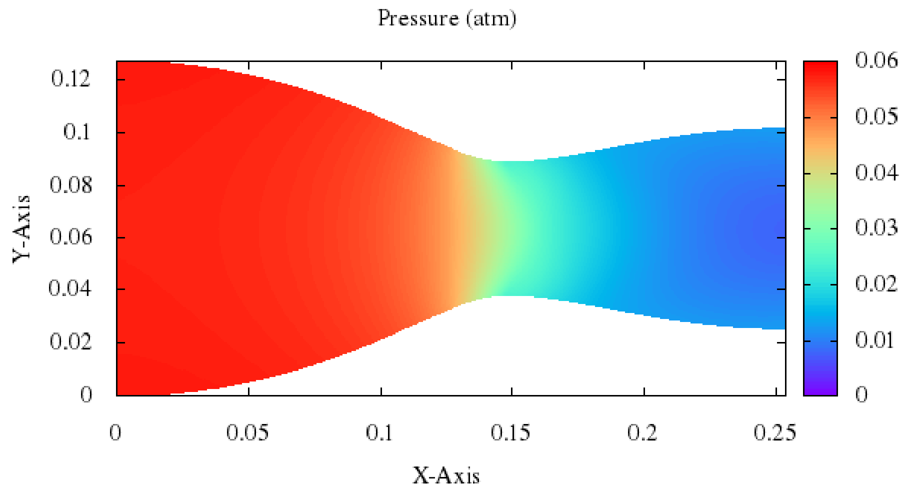


(d) Temperature contours

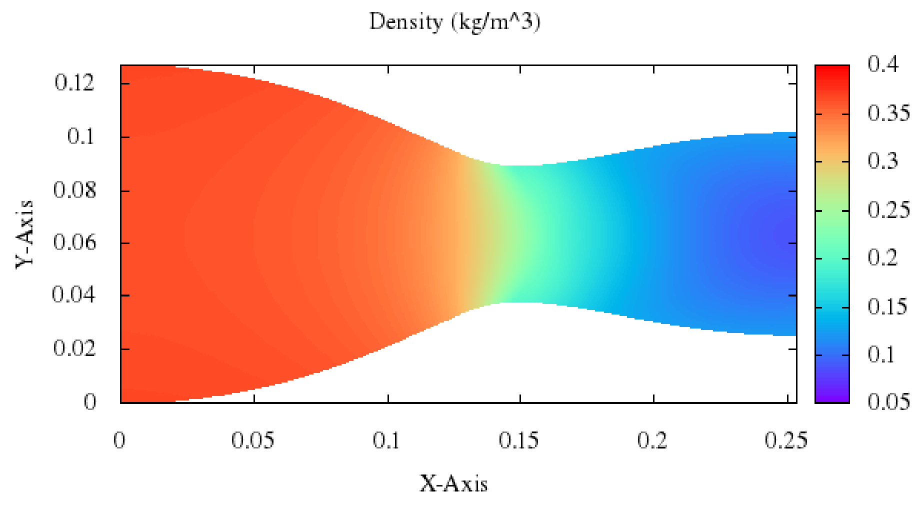
Figure 94: Property contours for converged solution with 133,400 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



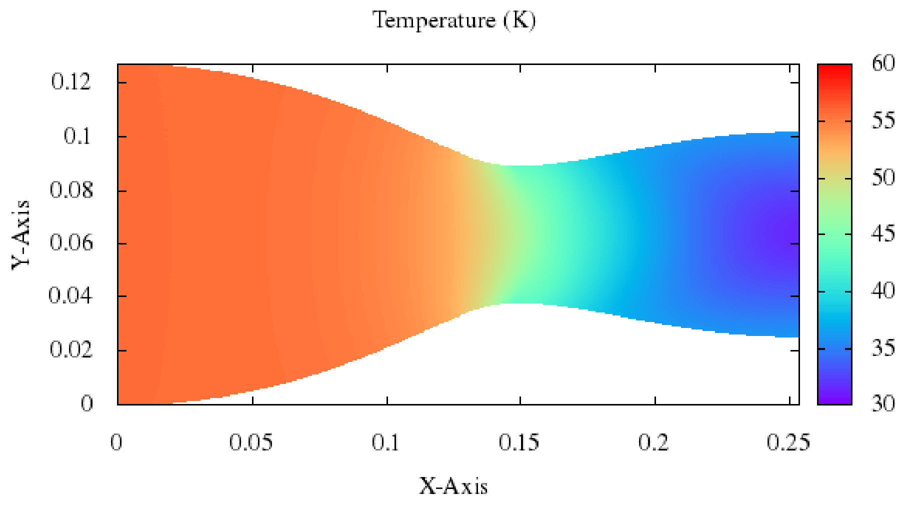
(a) Mach number contours



(b) Pressure contours

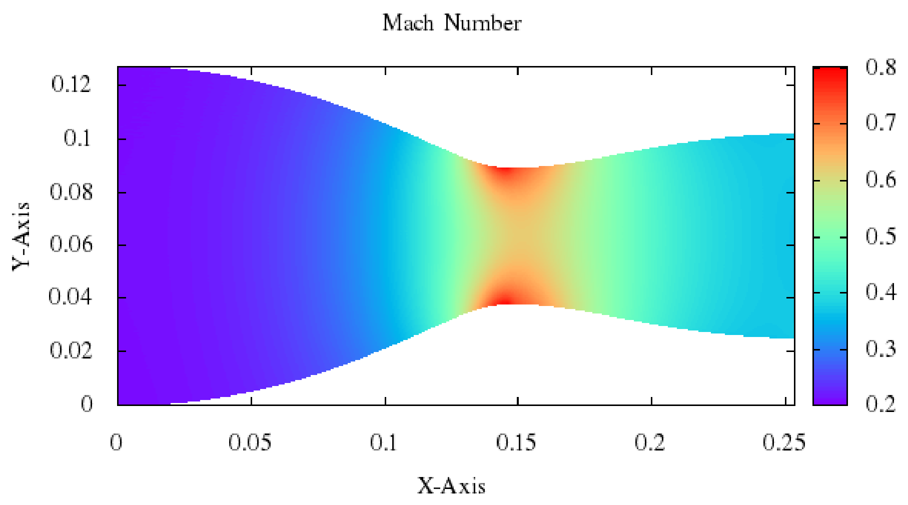


(c) Density contours

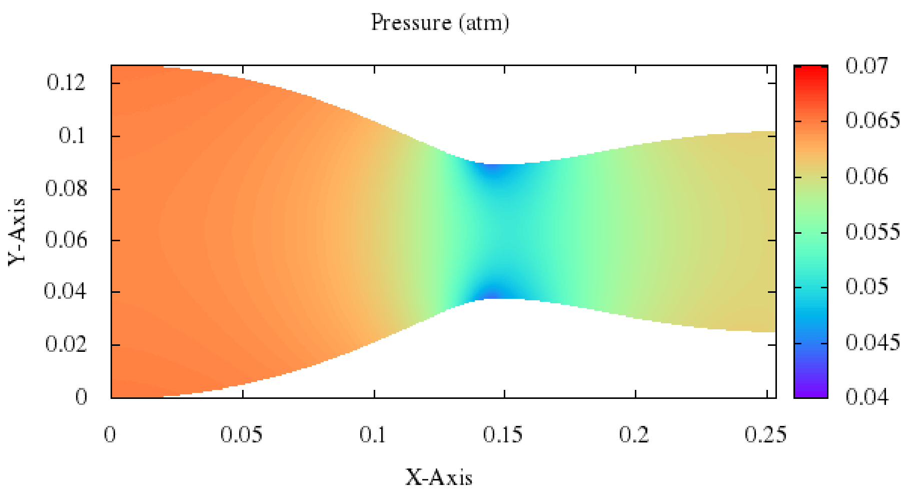


(d) Temperature contours

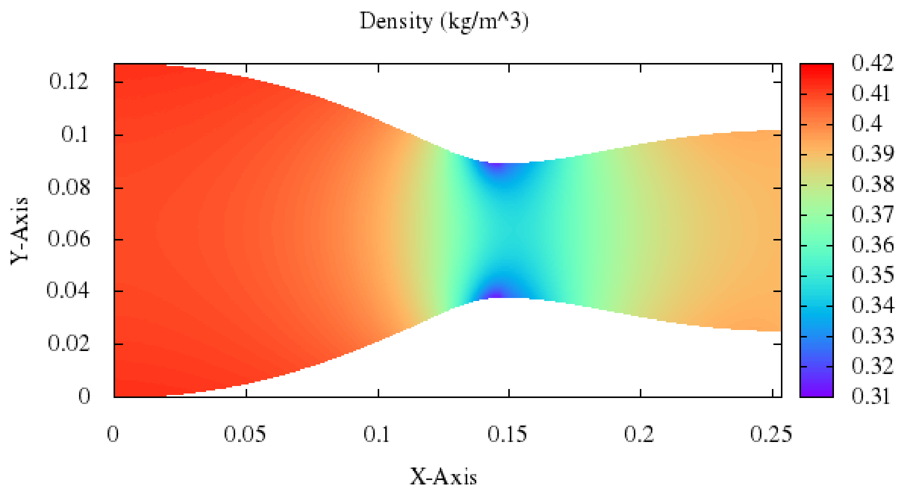
Figure 95: Property contours for converged solution with 133,400 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.



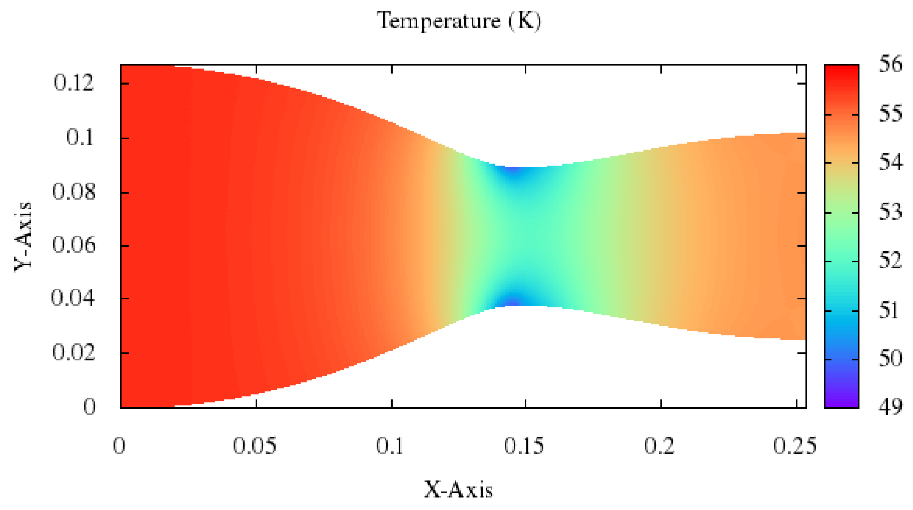
(a) Mach number contours



(b) Pressure contours

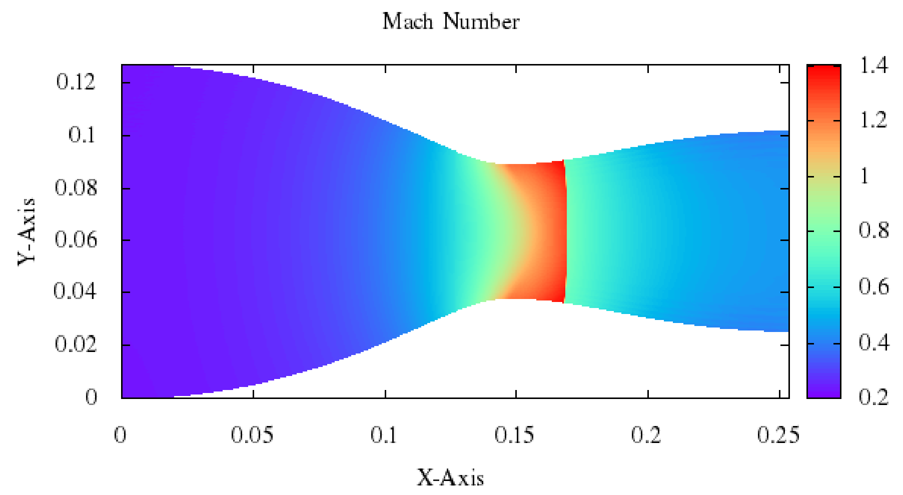


(c) Density contours

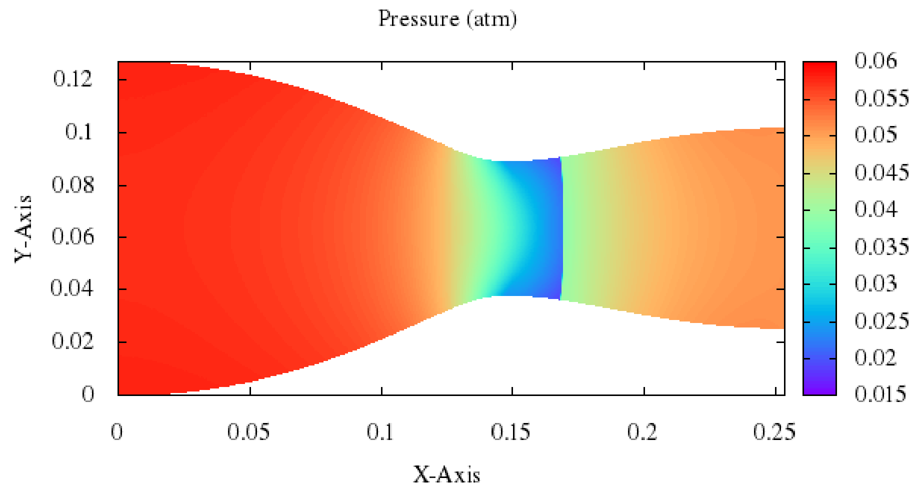


(d) Temperature contours

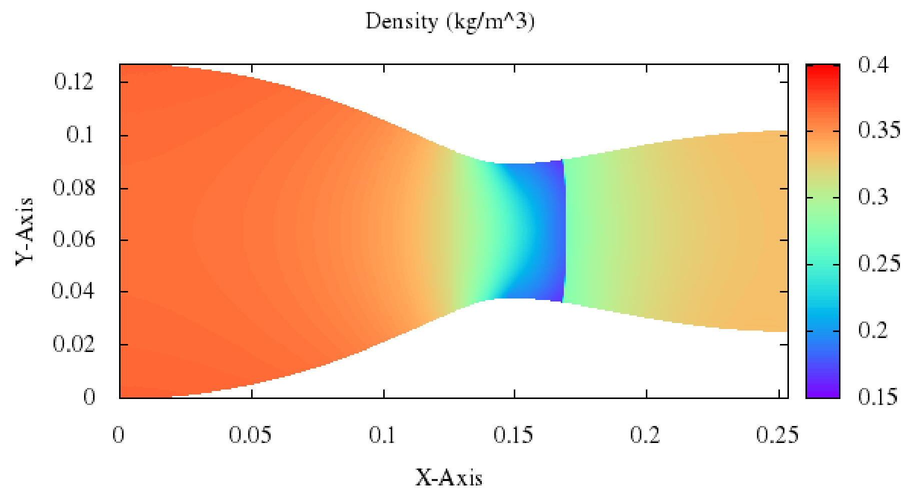
Figure 96: Property contours for converged solution with 298,103 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



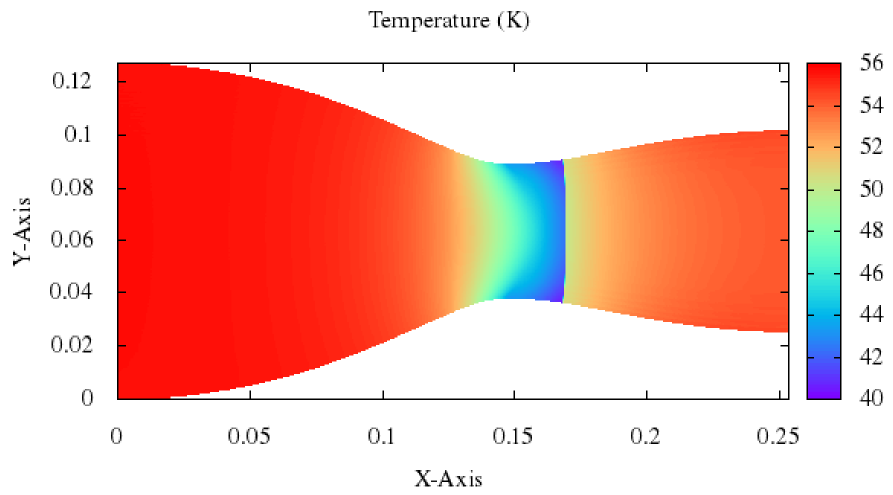
(a) Mach number contours



(b) Pressure contours

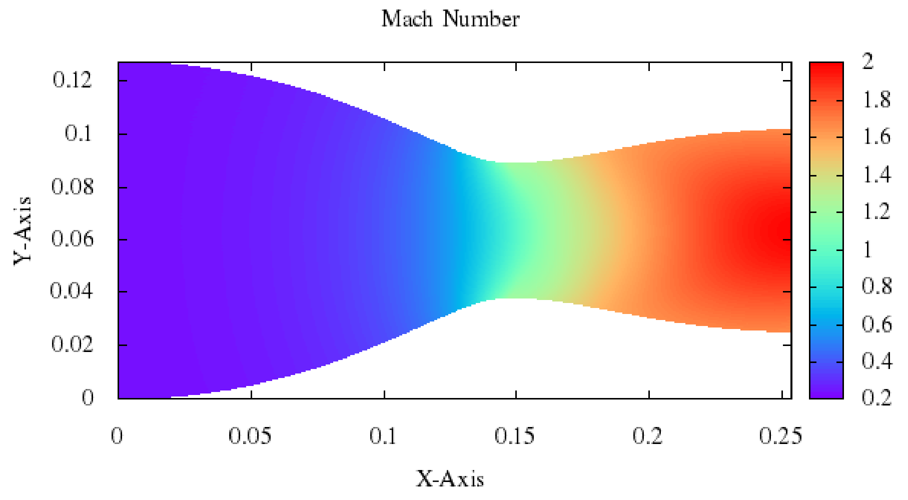


(c) Density contours

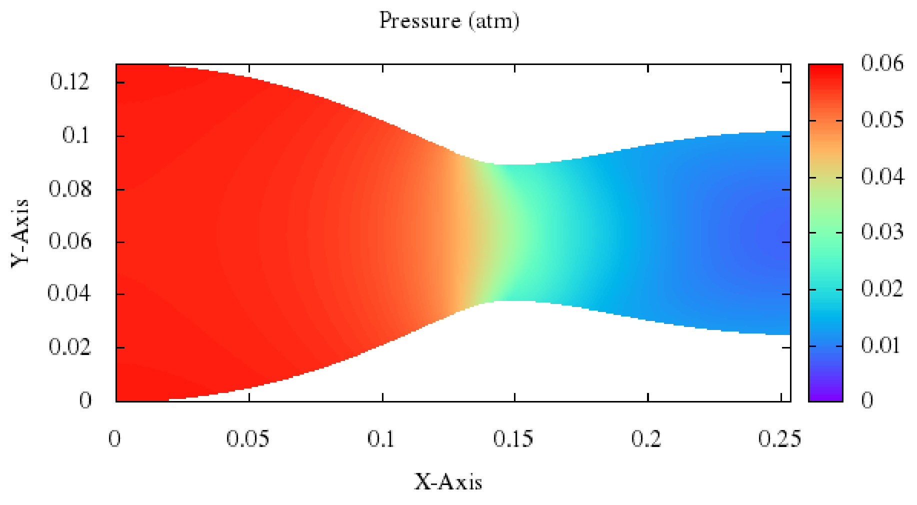


(d) Temperature contours

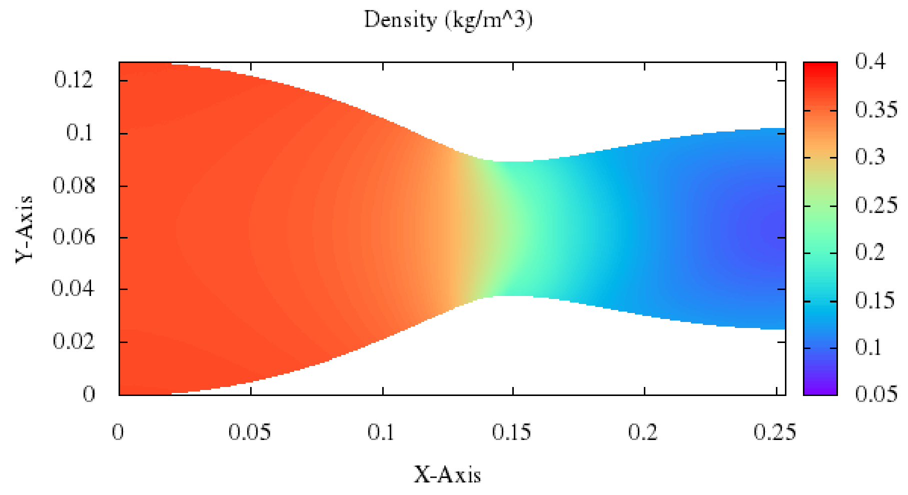
Figure 97: Property contours for converged solution with 298,103 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



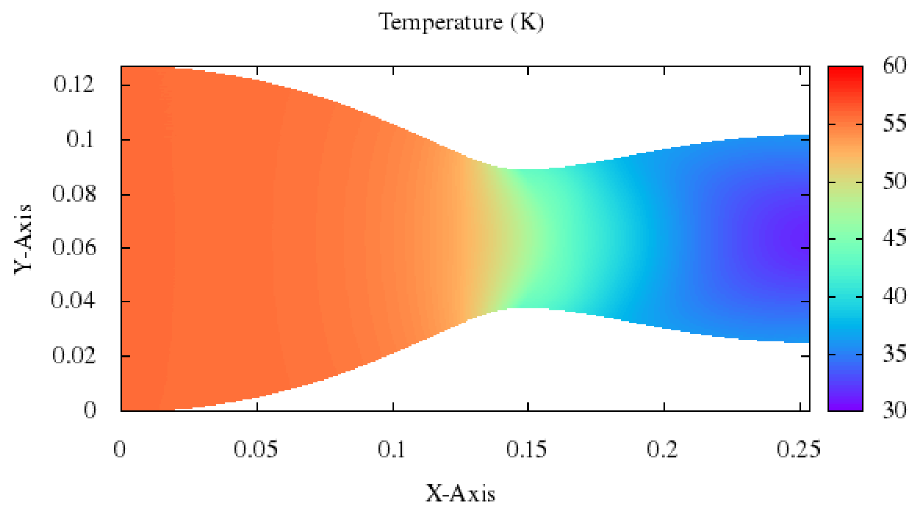
(a) Mach number contours



(b) Pressure contours

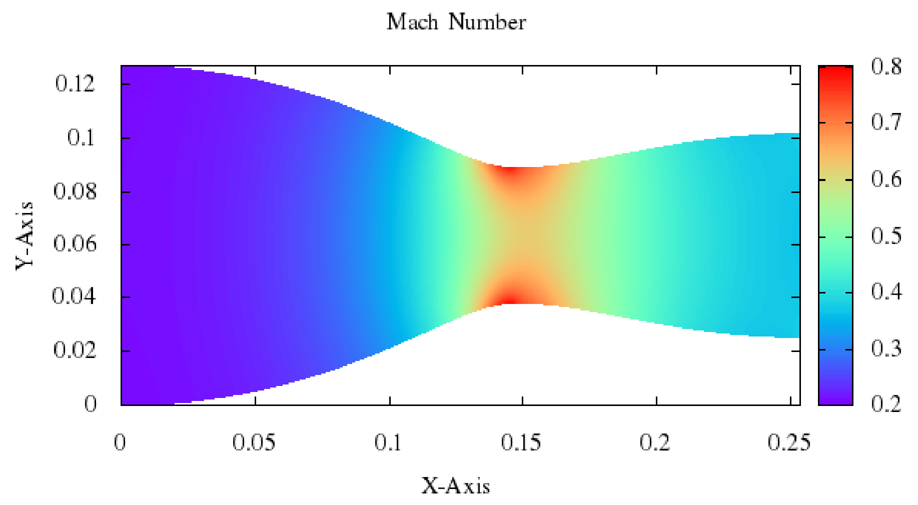


(c) Density contours

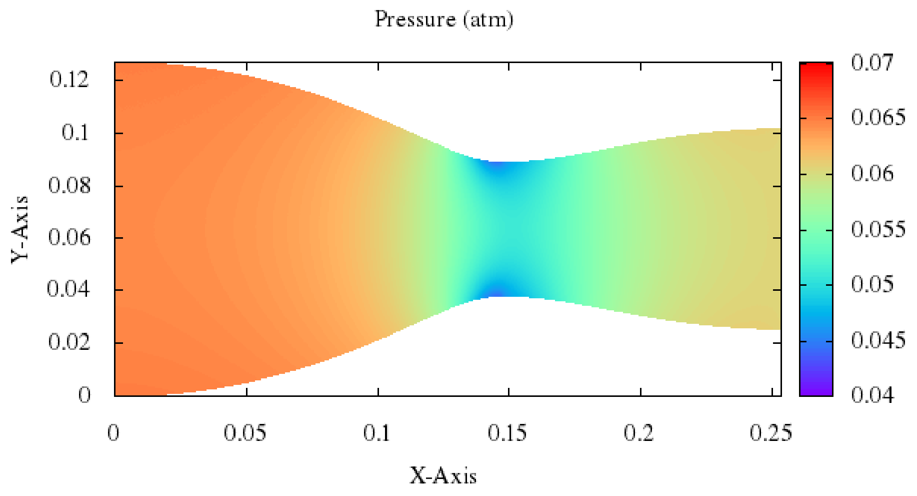


(d) Temperature contours

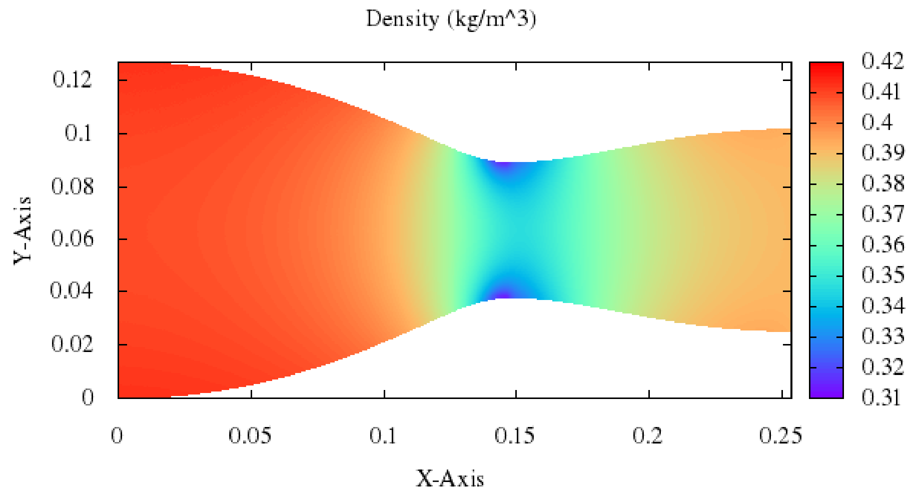
Figure 98: Property contours for converged solution with 298,103 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.



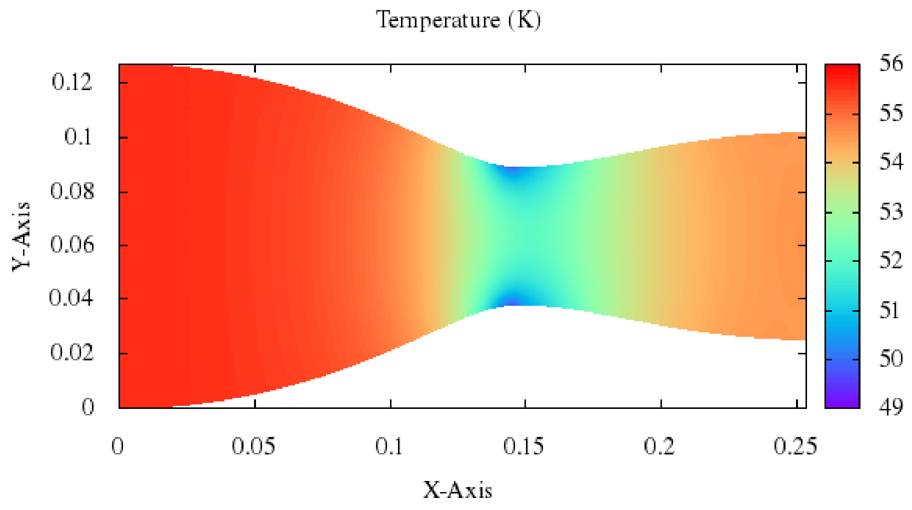
(a) Mach number contours



(b) Pressure contours

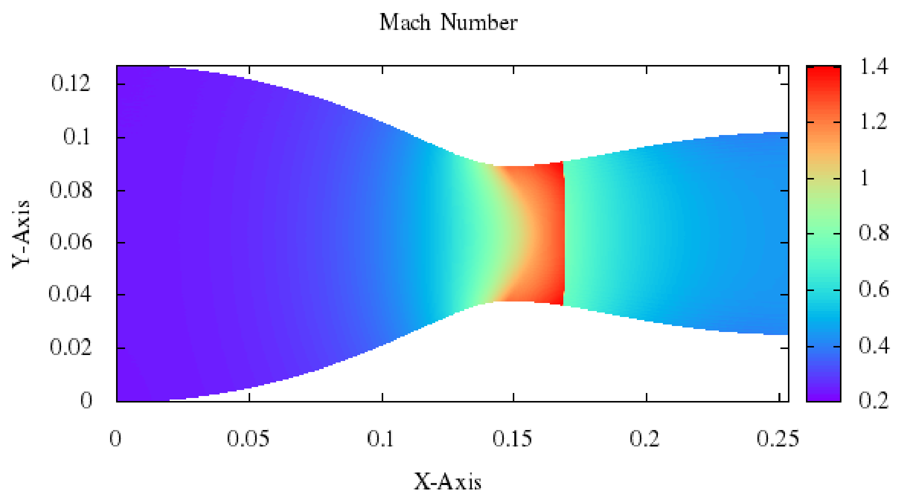


(c) Density contours

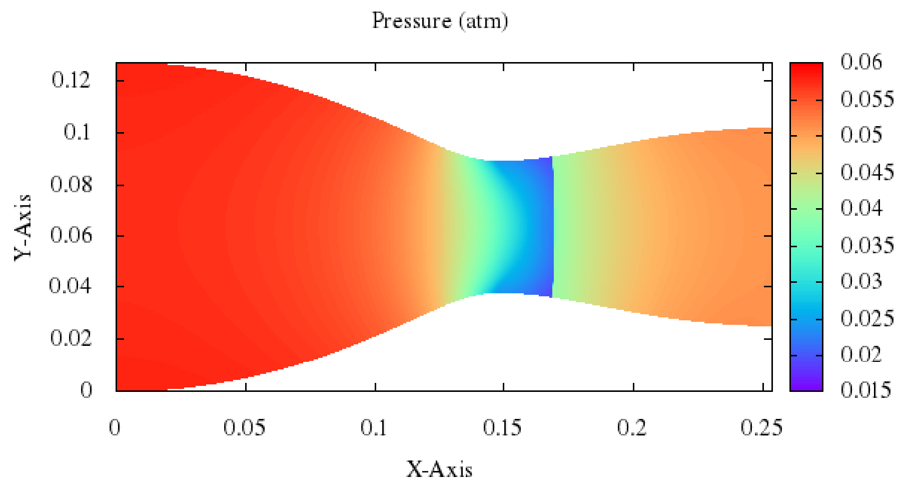


(d) Temperature contours

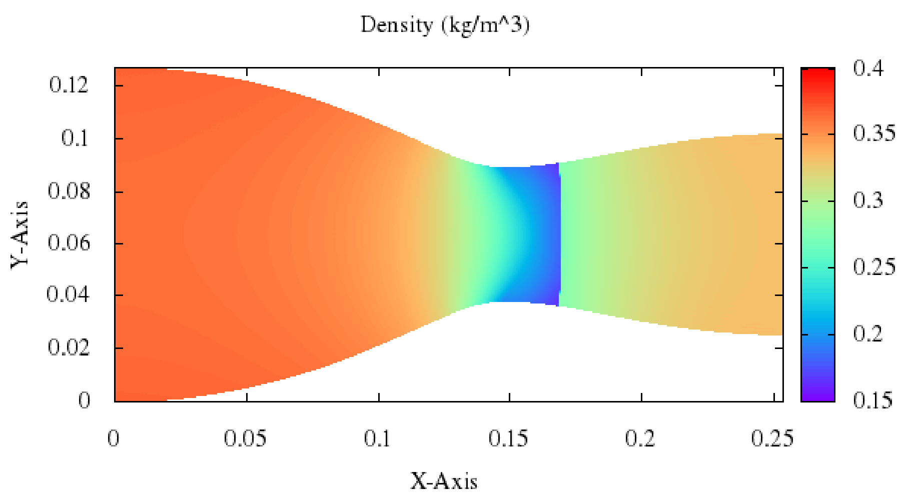
Figure 99: Property contours for converged solution with 830,336 cells and a pressure ratio of $p_{out}/p_{in} = 0.89$. Axes are in meters.



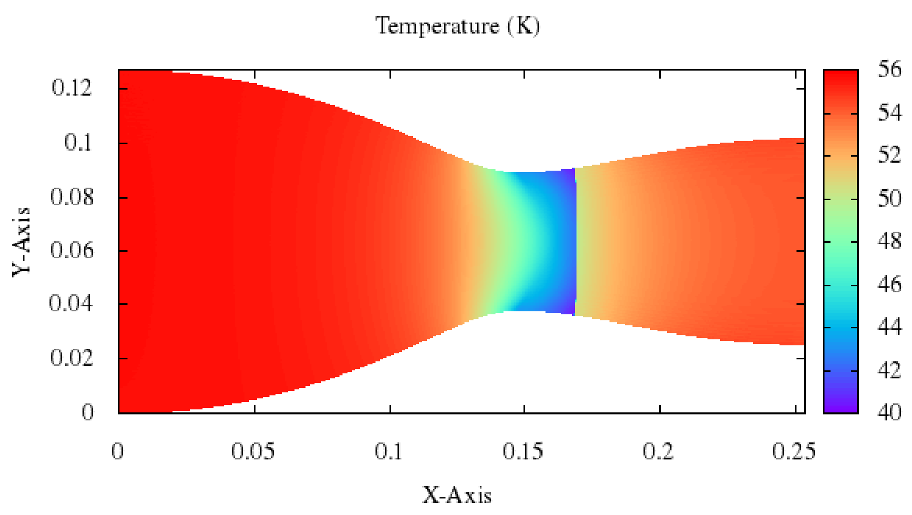
(a) Mach number contours



(b) Pressure contours

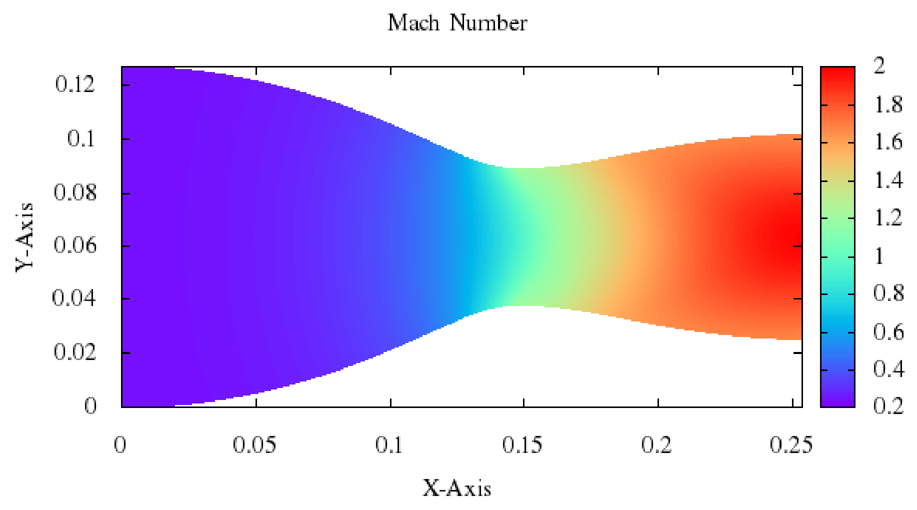


(c) Density contours

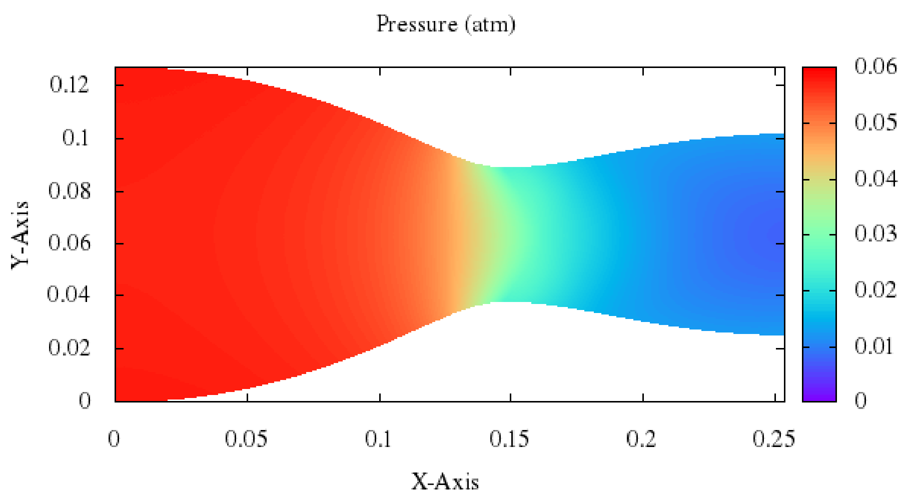


(d) Temperature contours

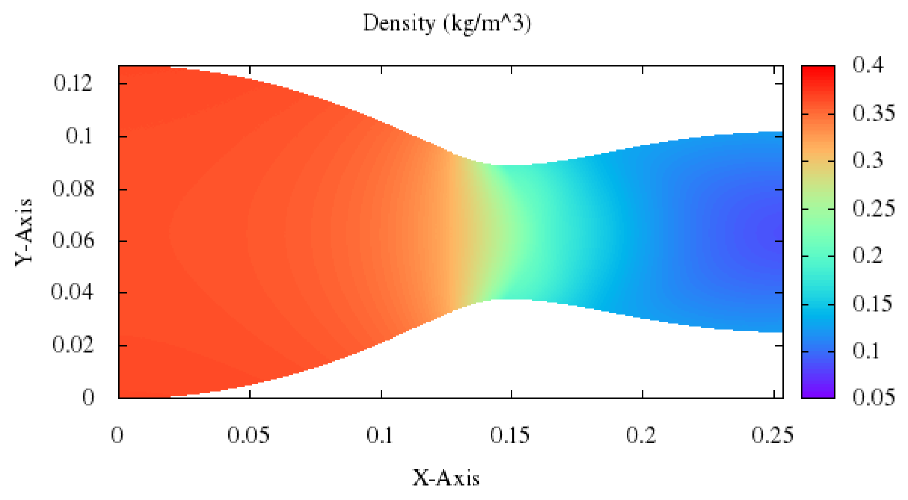
Figure 100: Property contours for converged solution with 830,336 cells and a pressure ratio of $p_{out}/p_{in} = 0.75$. Axes are in meters.



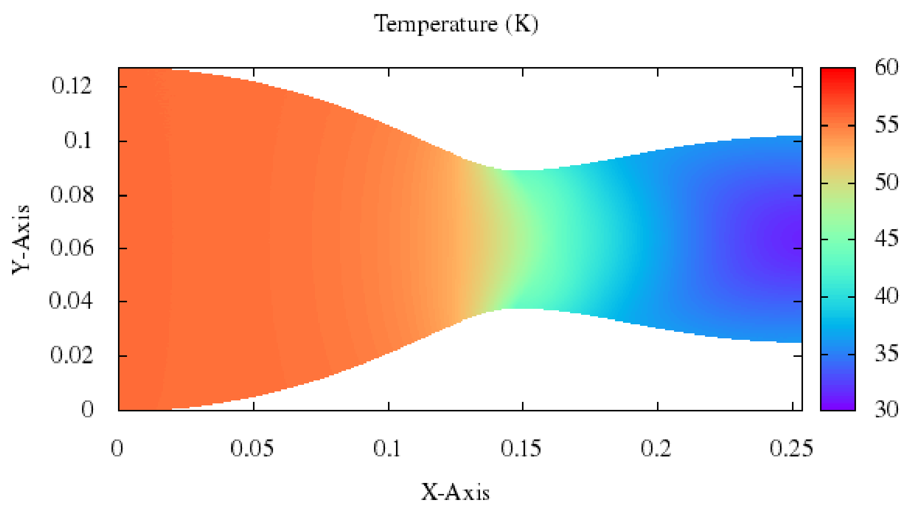
(a) Mach number contours



(b) Pressure contours



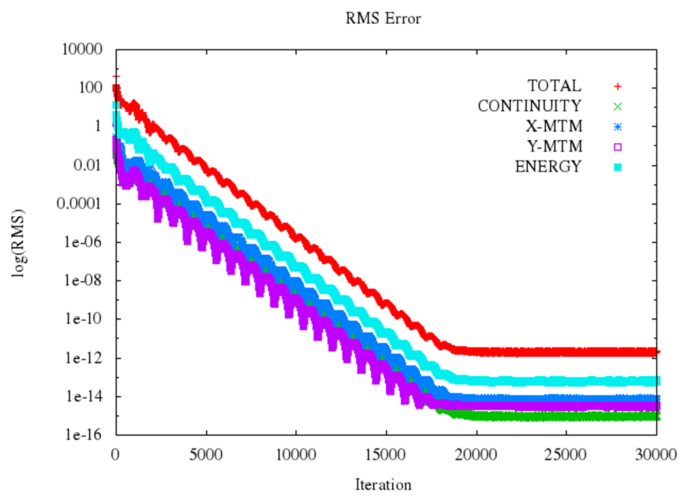
(c) Density contours



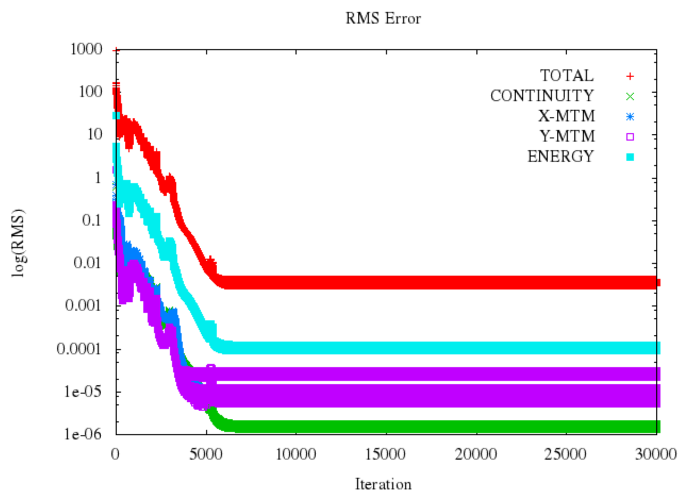
(d) Temperature contours

Figure 101: Property contours for converged solution with 830,336 cells and a pressure ratio of $p_{out}/p_{in} = 0.16$. Axes are in meters.

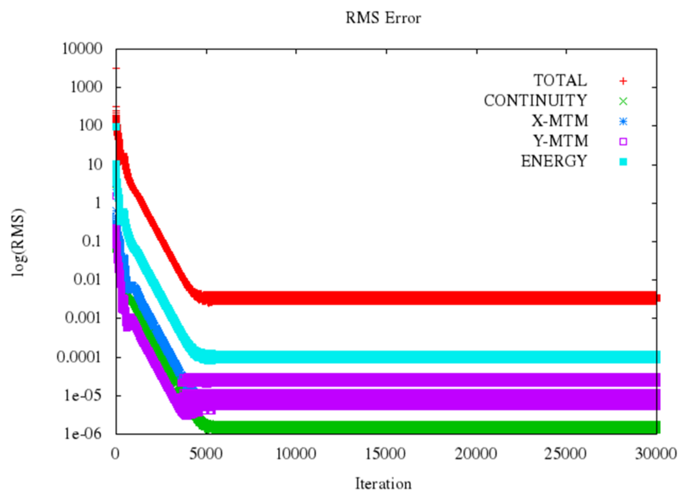
The plots in Figures 102 through 107 show the RMS error plots as a function of iteration number for each of the grids (62x20 nodes through 1665x500 nodes) for all three pressure ratios. These are included for completeness.



(a) $p_{out}/p_{in} = 0.89$

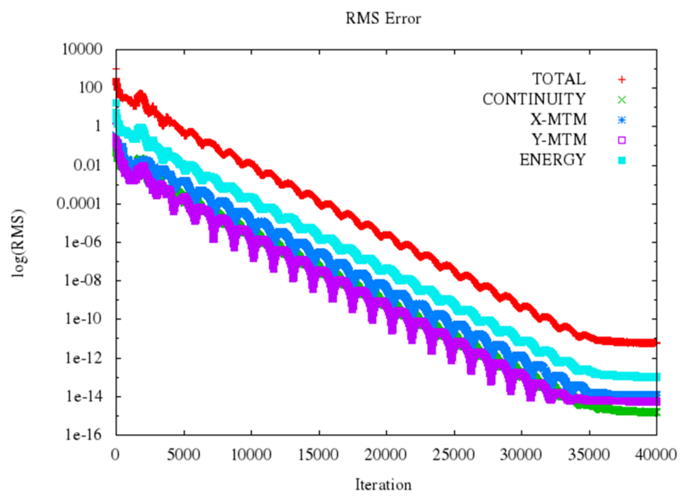


(b) $p_{out}/p_{in} = 0.75$

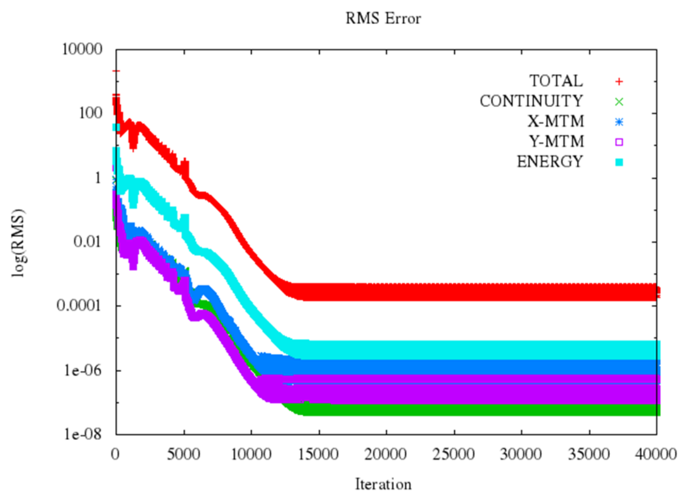


(c) $p_{out}/p_{in} = 0.16$

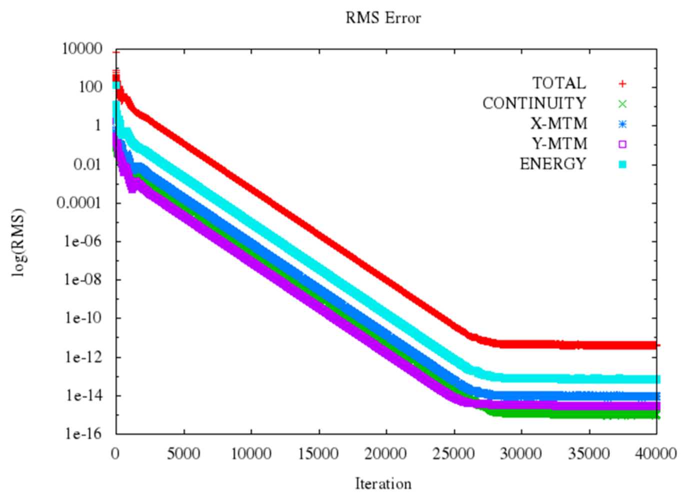
Figure 102: Absolute RMS Error for converged solution with 1,216 cells.



(a) $p_{out}/p_{in} = 0.89$

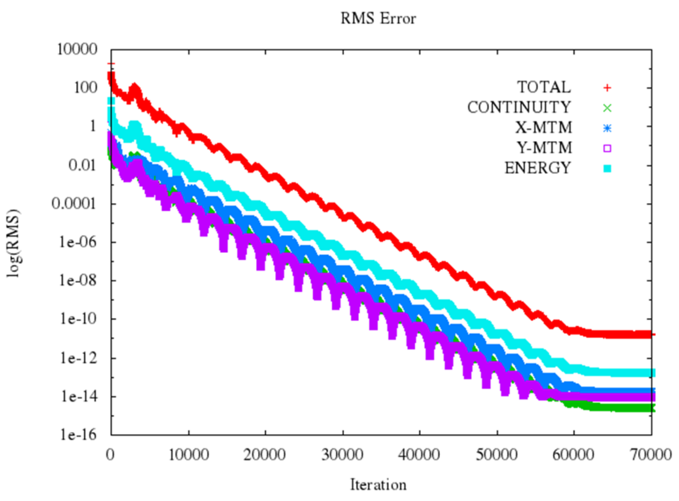


(b) $p_{out}/p_{in} = 0.75$

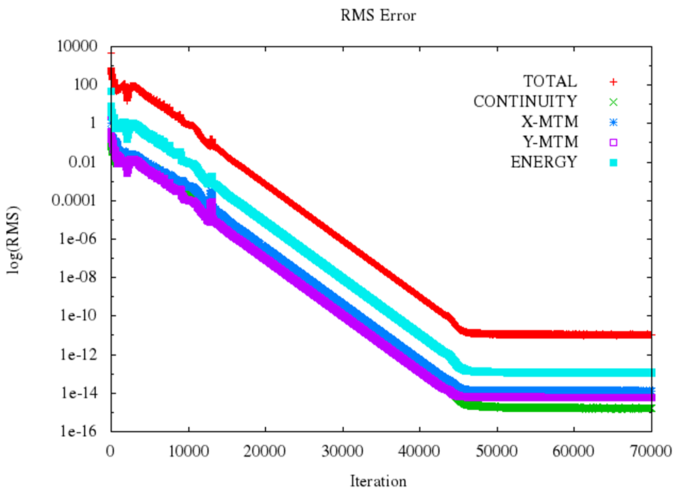


(c) $p_{out}/p_{in} = 0.16$

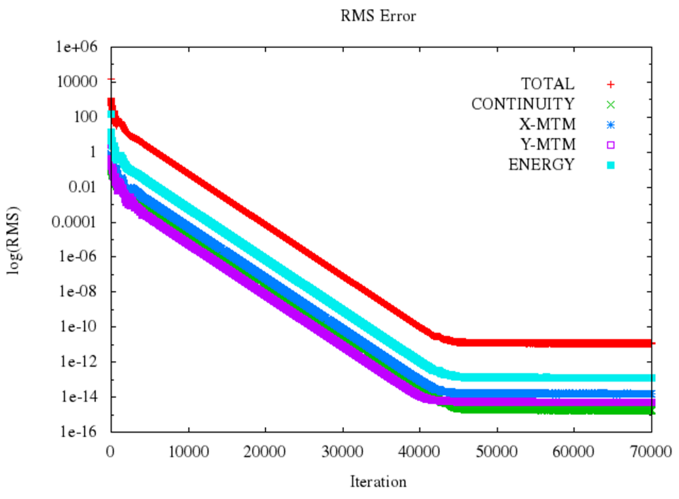
Figure 103: Absolute RMS Error for converged solution with 3,876 cells.



(a) $p_{out}/p_{in} = 0.89$

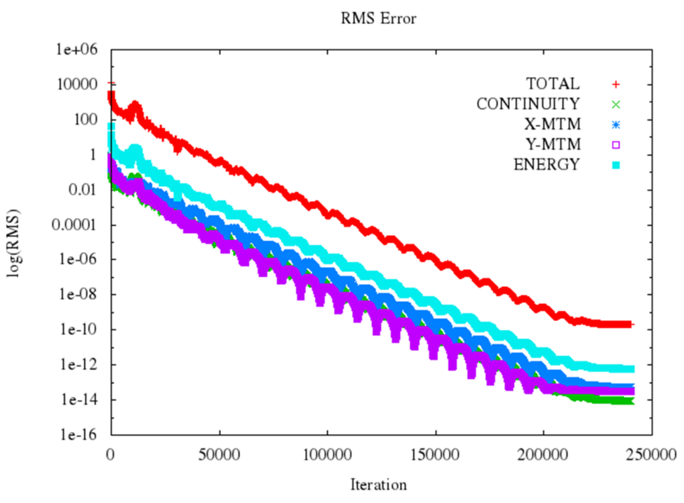


(b) $p_{out}/p_{in} = 0.75$

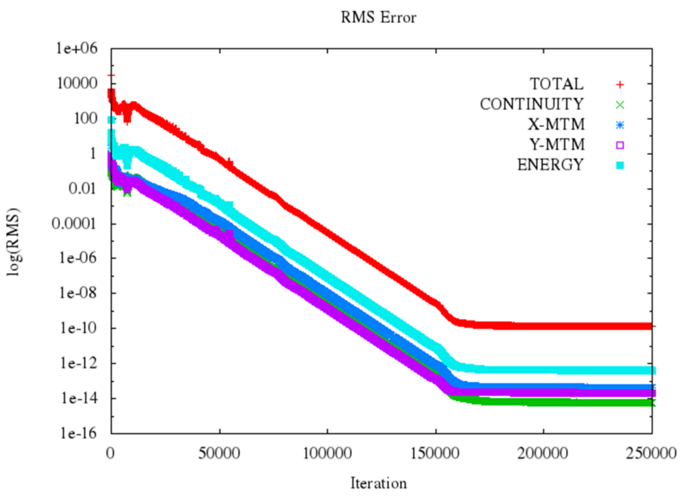


(c) $p_{out}/p_{in} = 0.16$

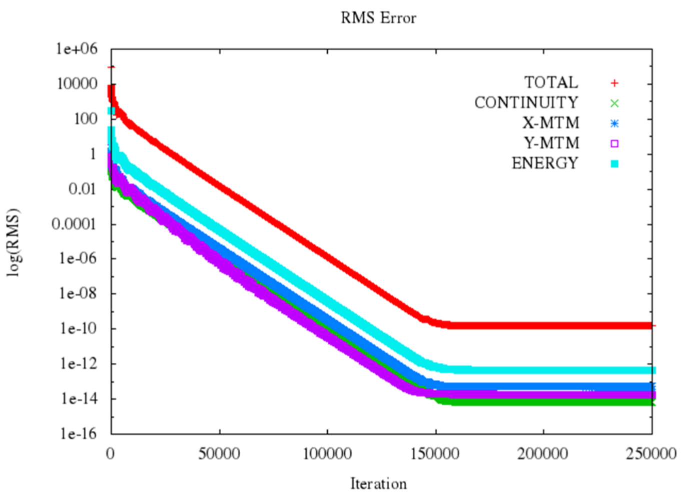
Figure 104: Absolute RMS Error for converged solution with 10,472 cells.



(a) $p_{out}/p_{in} = 0.89$

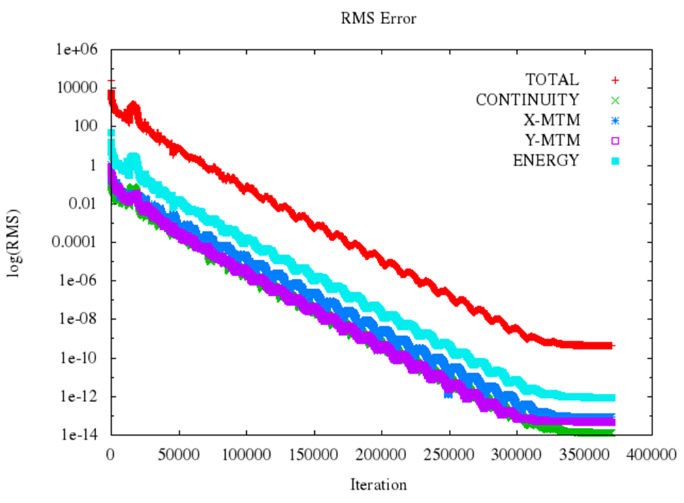


(b) $p_{out}/p_{in} = 0.75$

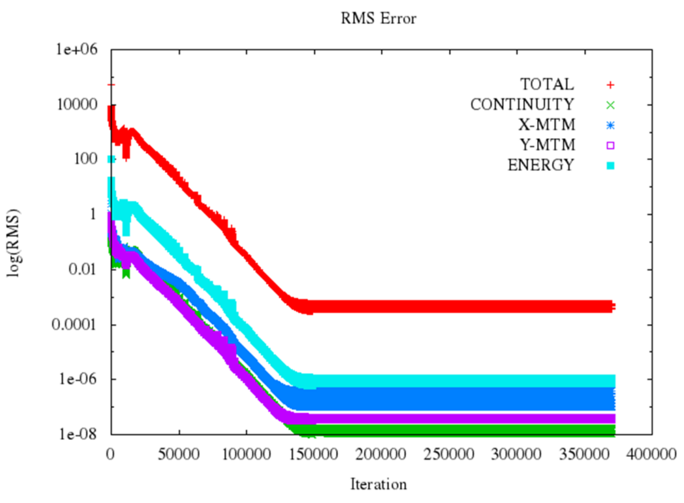


(c) $p_{out}/p_{in} = 0.16$

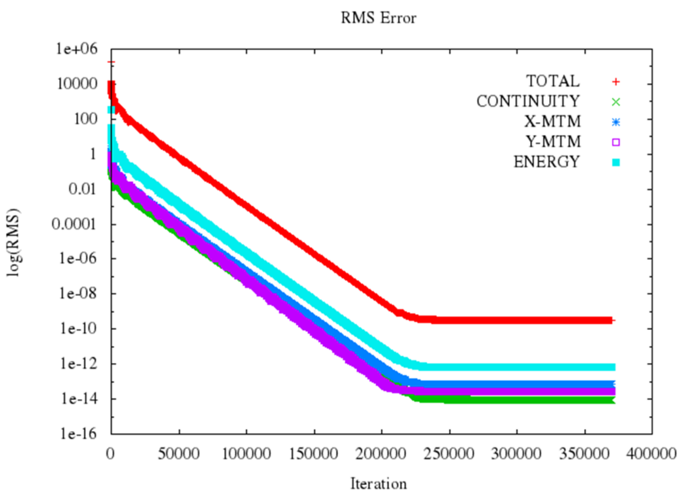
Figure 105: Absolute RMS Error for converged solution with 133,400 cells.



(a) $p_{out}/p_{in} = 0.89$

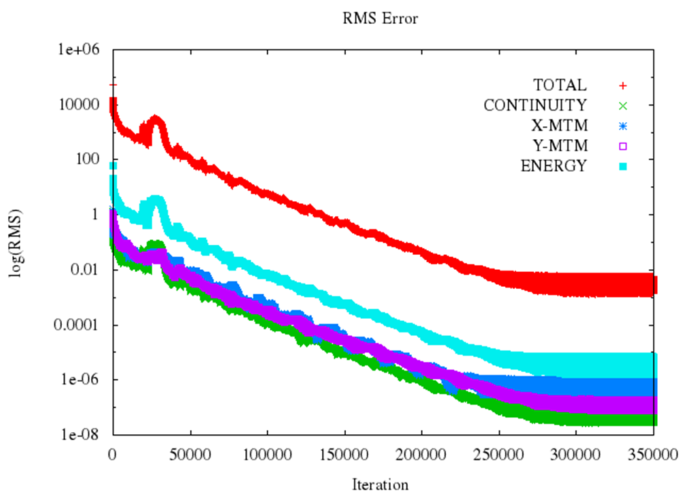


(b) $p_{out}/p_{in} = 0.75$

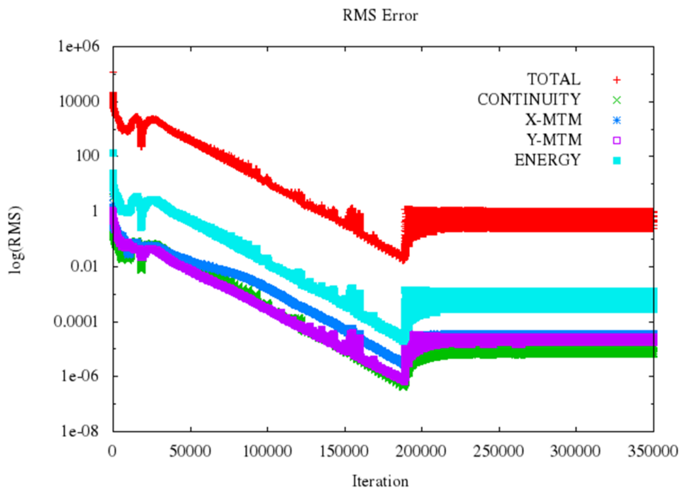


(c) $p_{out}/p_{in} = 0.16$

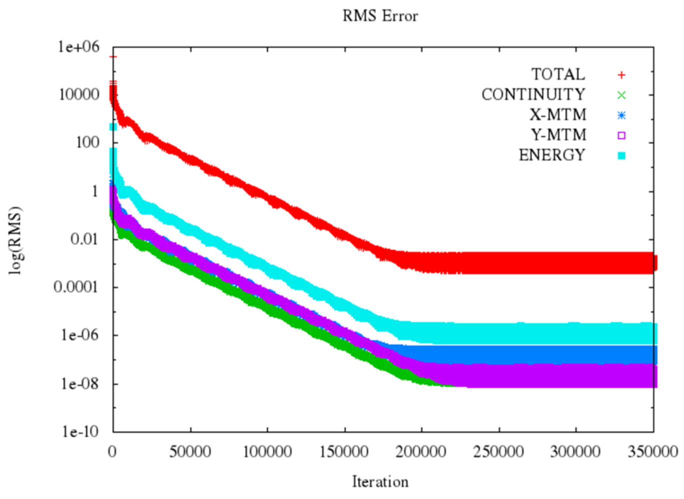
Figure 106: Absolute RMS Error for converged solution with 298,103 cells.



(a) $p_{out}/p_{in} = 0.89$



(b) $p_{out}/p_{in} = 0.75$



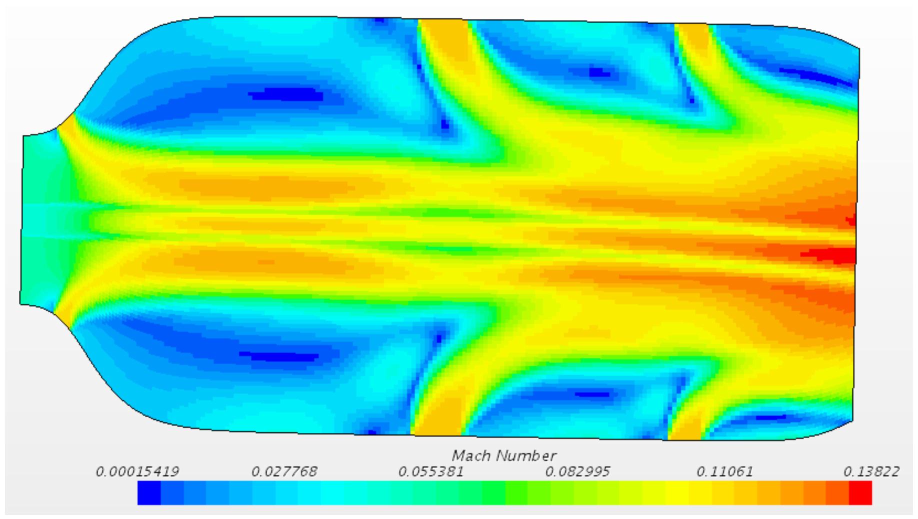
(c) $p_{out}/p_{in} = 0.16$

Figure 107: Absolute RMS Error for converged solution with 830,336 cells.

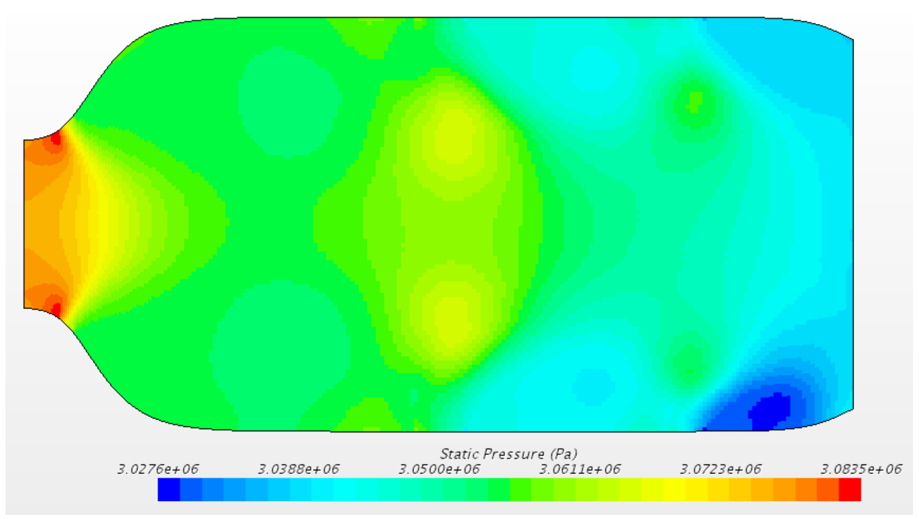
APPENDIX C

STAR-CCM+ RFCFD CONTOURS

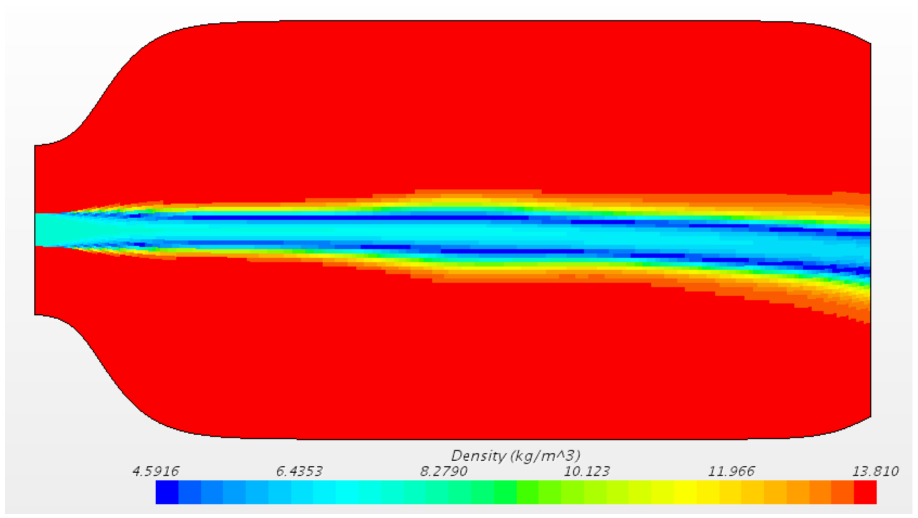
Below are the flow field contours for the standard properties (density, temperature, pressure, and Mach number), in addition RMS convergence plots for the STAR-CCM+ PPDF RFCFD runs. These are shown for the 222x100 grid at inlet Temperatures of 800 K to 1000 K, as discussed in Section 5.2.1. The contours for the other grids (37x17 through 166x75) are not included, however the temperature results are displayed in the plots in Figure 69.



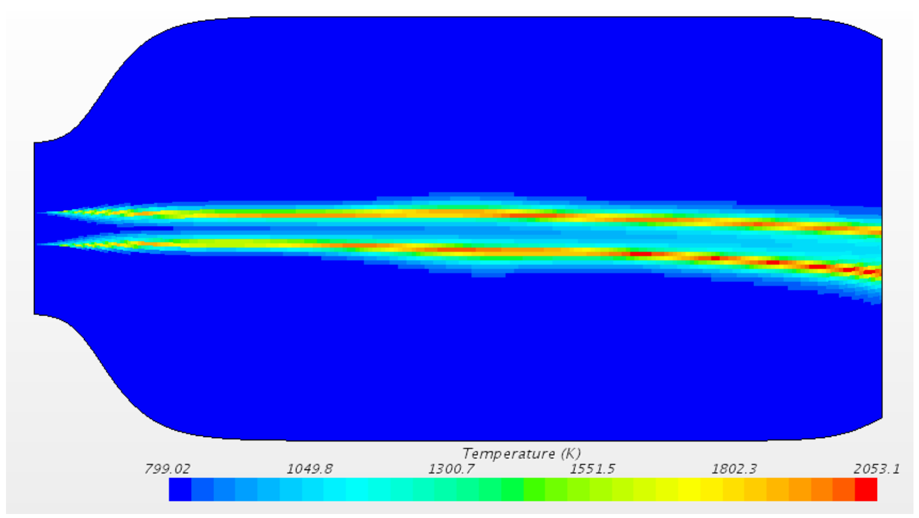
(a) Mach number contours



(b) Pressure contours

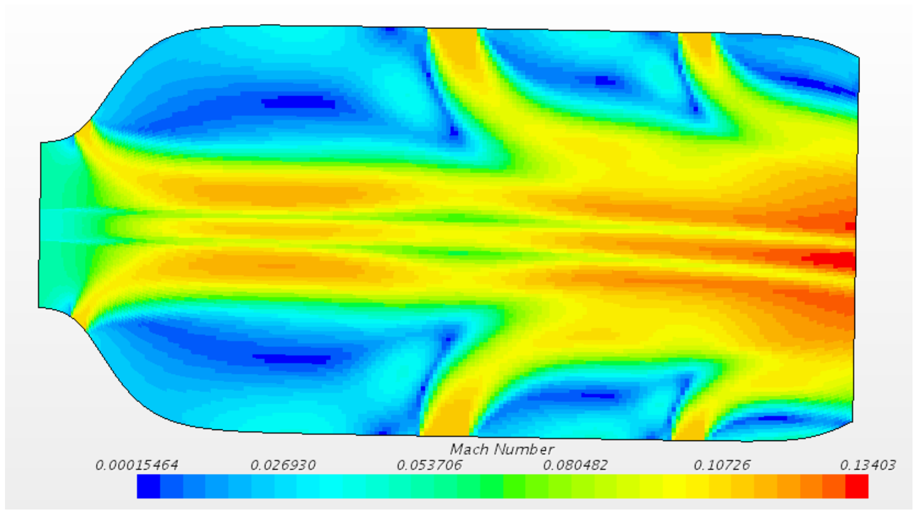


(c) Density contours

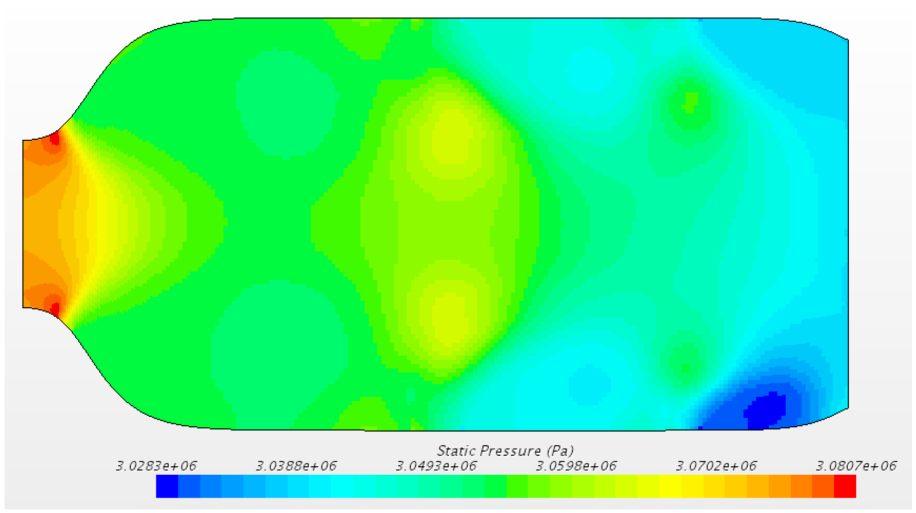


(d) Temperature contours

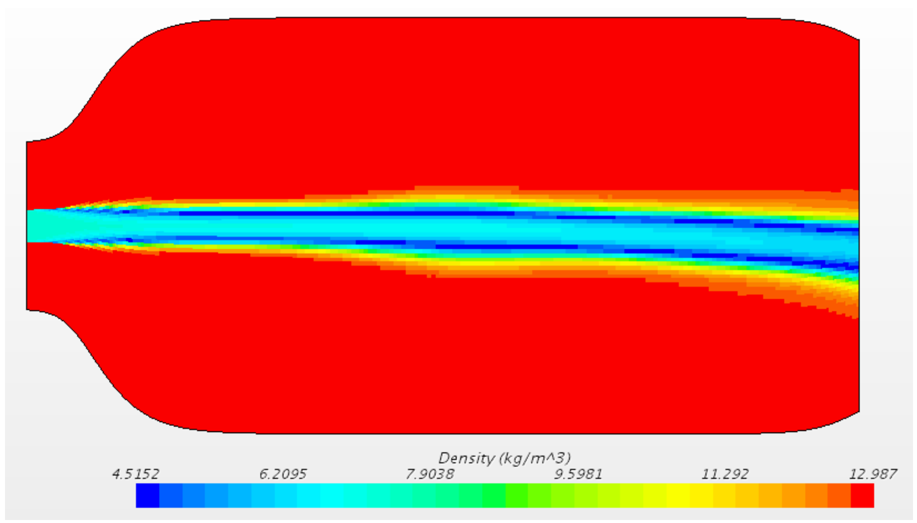
Figure 108: STAR-CCM+ PPDF RFCFD contours for the 222x100 grid at $T_3 = 800K$ and $\phi \approx 0.5$



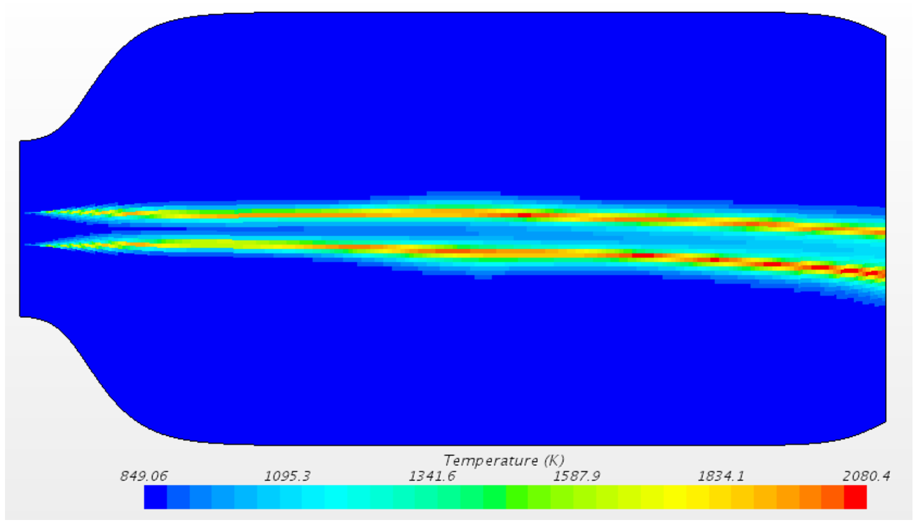
(a) Mach number contours



(b) Pressure contours

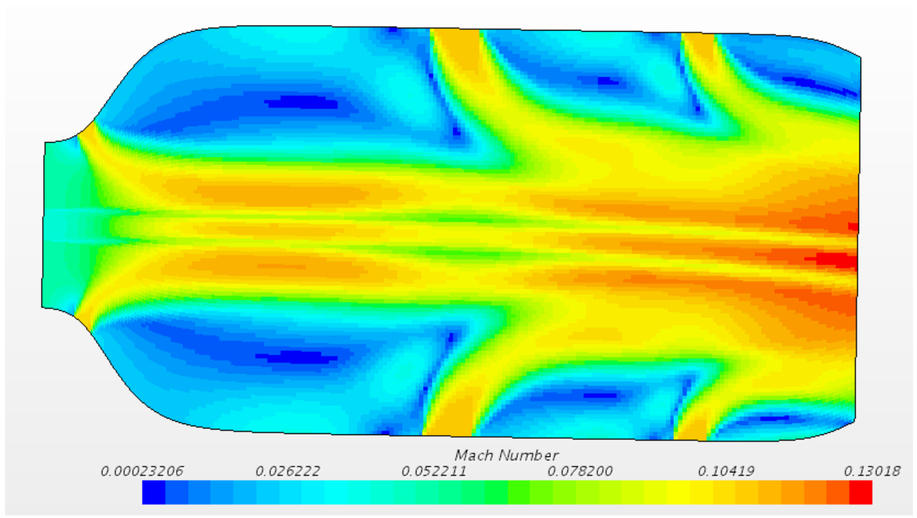


(c) Density contours

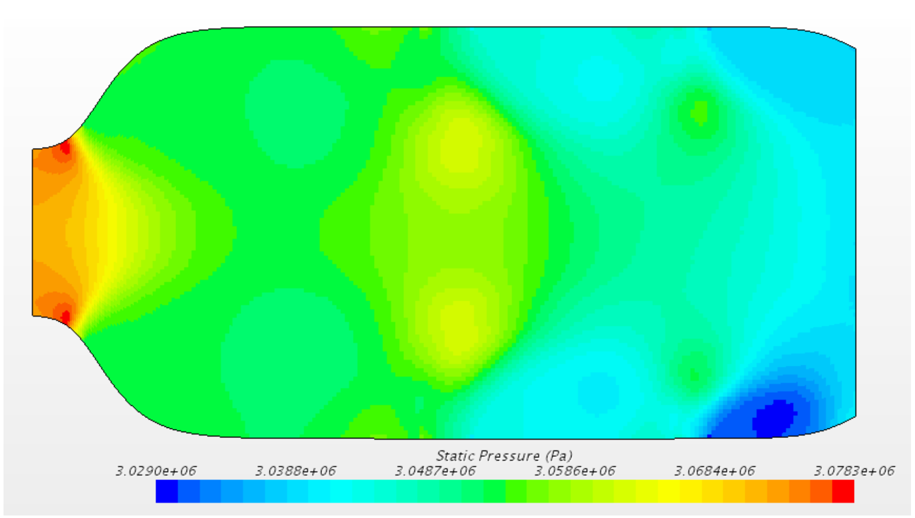


(d) Temperature contours

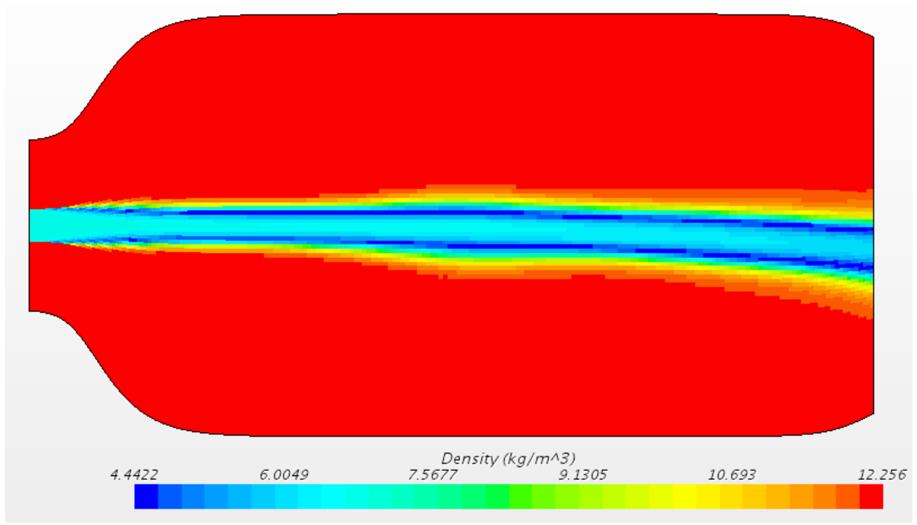
Figure 109: STAR-CCM+ PPDF RFCFD contours for the 222x100 grid at $T_3 = 850K$ and $\phi \approx 0.5$



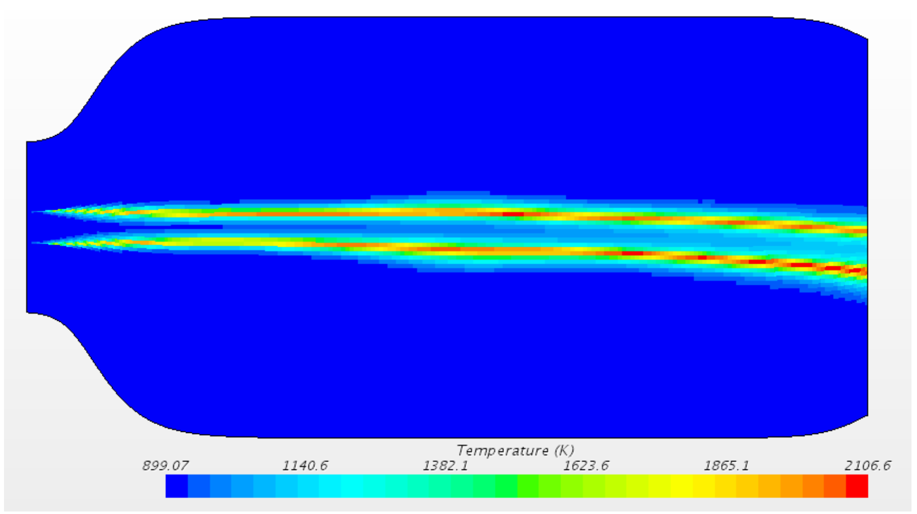
(a) Mach number contours



(b) Pressure contours

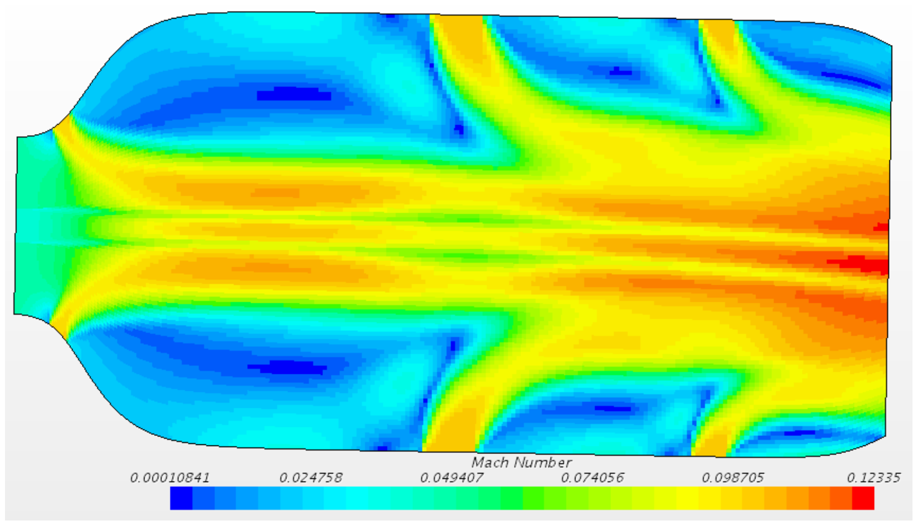


(c) Density contours

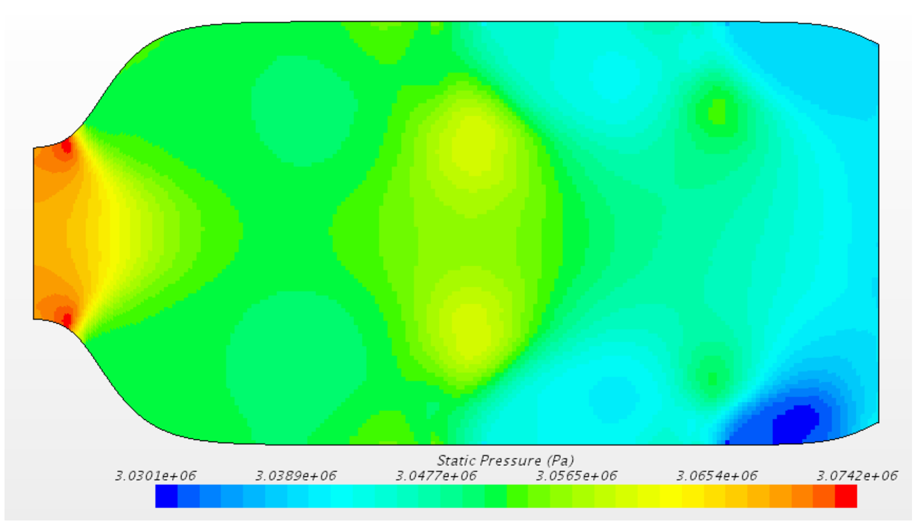


(d) Temperature contours

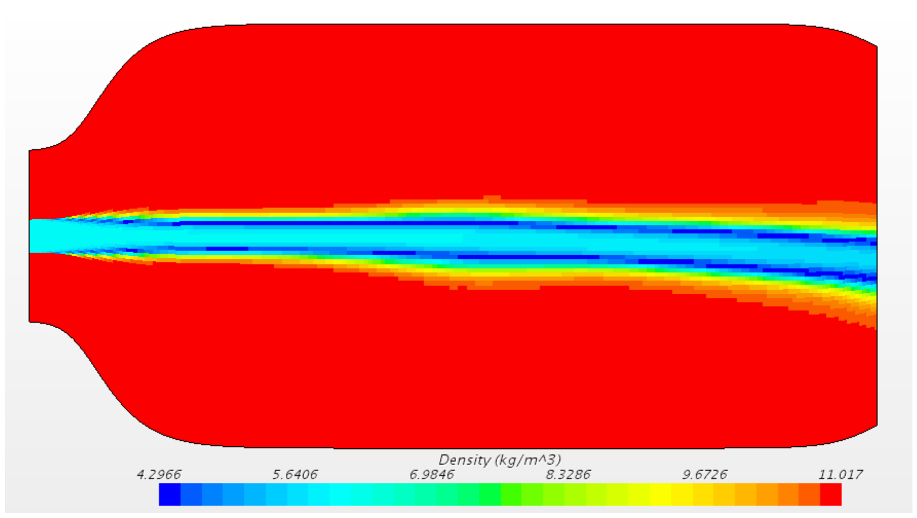
Figure 110: STAR-CCM+ PPDF RFCFD contours for the 222x100 grid at $T_3 = 900K$ and $\phi \approx 0.5$



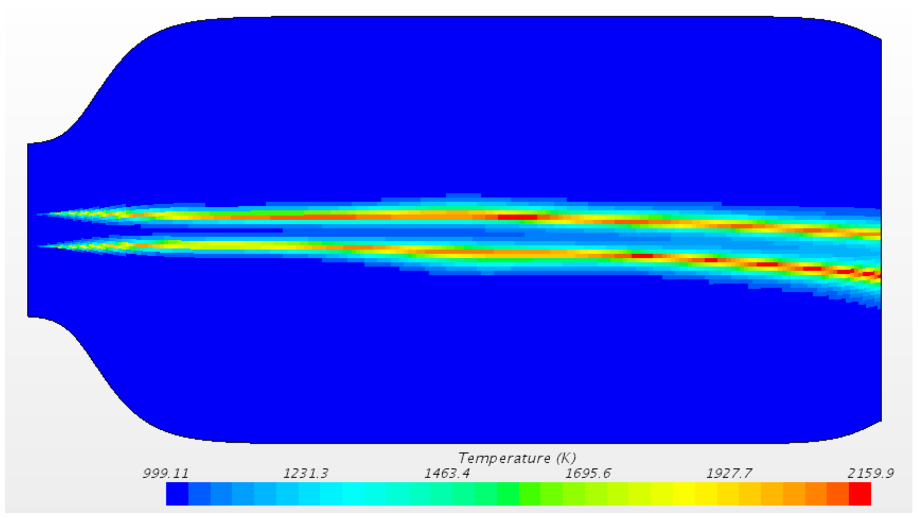
(a) Mach number contours



(b) Pressure contours

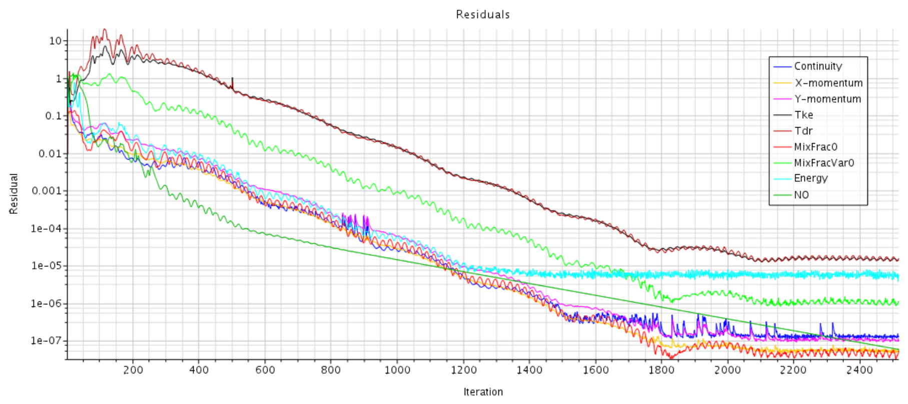


(c) Density contours

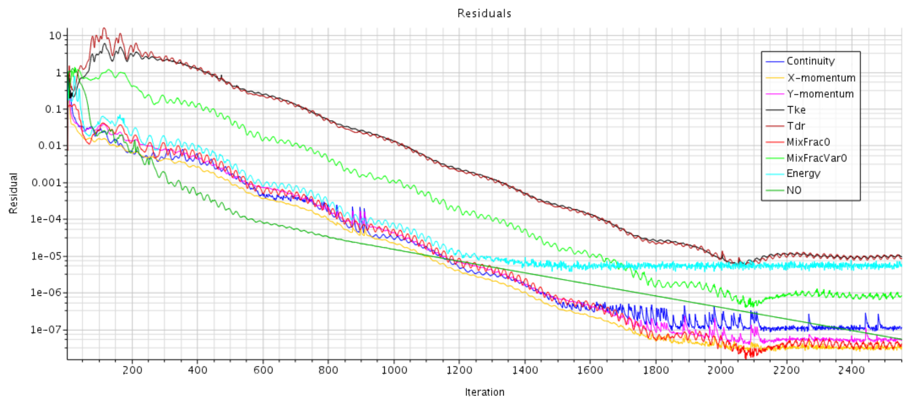


(d) Temperature contours

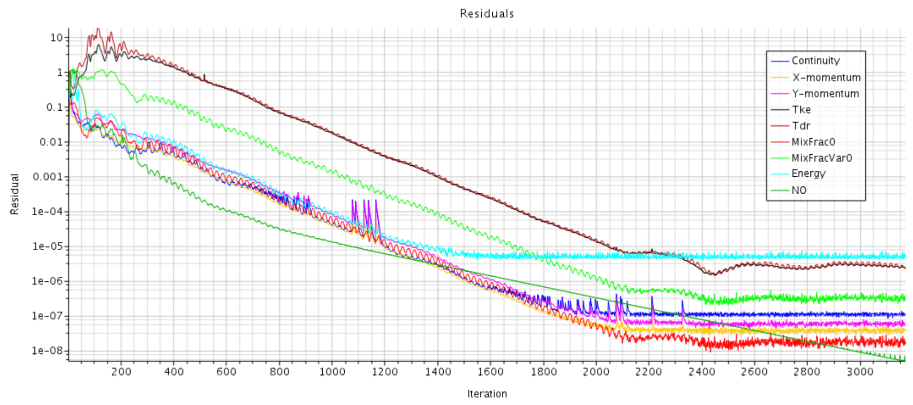
Figure 111: STAR-CCM+ PPDF RFCFD contours for the 222x100 grid at $T_3 = 1000K$ and $\phi \approx 0.5$



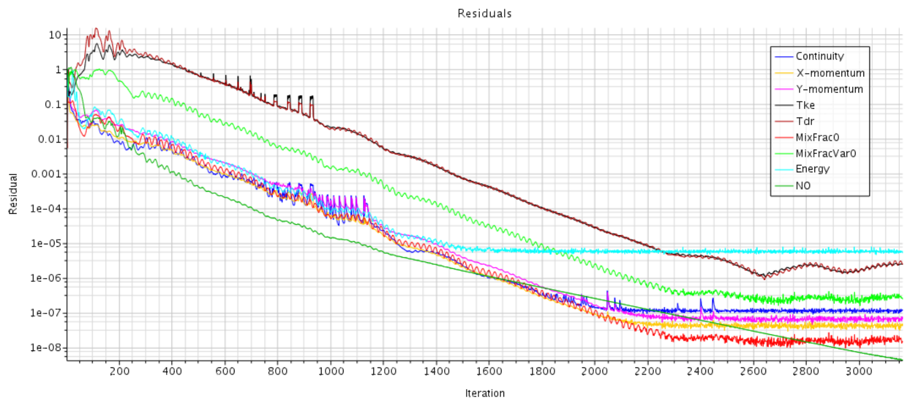
(a) $T_3 = 800K$



(b) $T_3 = 850K$



(c) $T_3 = 900K$



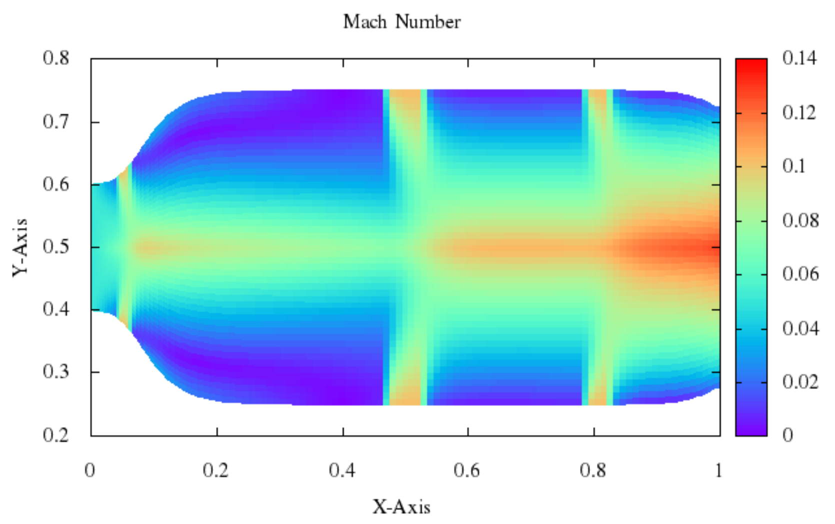
(d) $T_3 = 800K$

Figure 112: STAR-CCM+ PPDF RFCFD RMS contours for the 222x100 grid at $T_3 = 800K$ through $T_3 = 1000K$ and $\phi \approx 0.5$

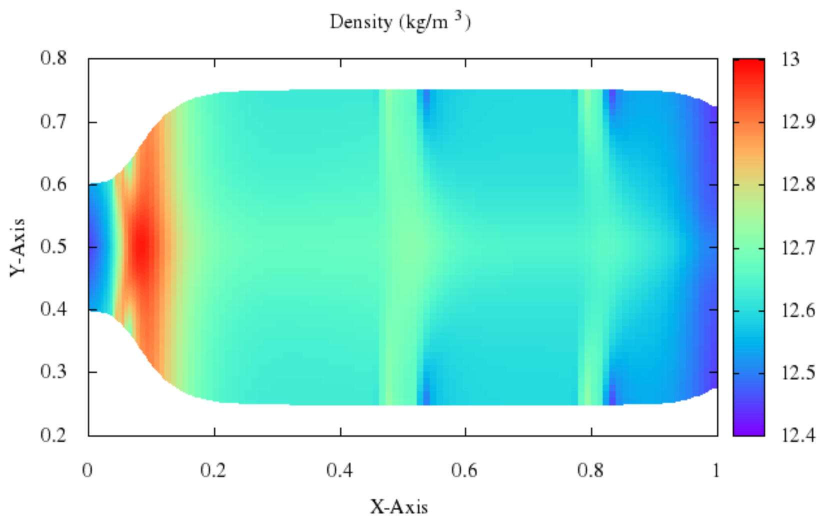
APPENDIX D

CFD COMBUSTOR CAN RESULTS

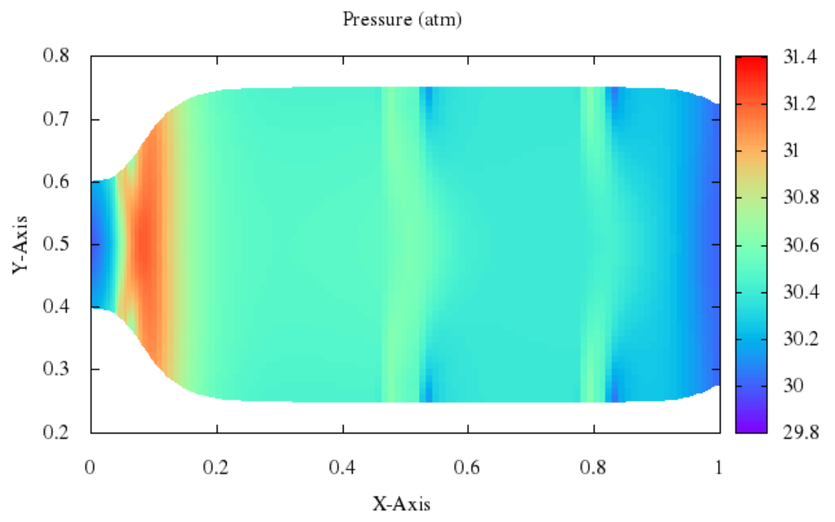
Below are the non-reacting CFD flow contour solutions for the 110x50 grid at $T_3 = 850K$, $T_3 = 900K$, and $T_3 = 1000K$. Each were run to an absolute RMS of 1.0×10^{-6}.



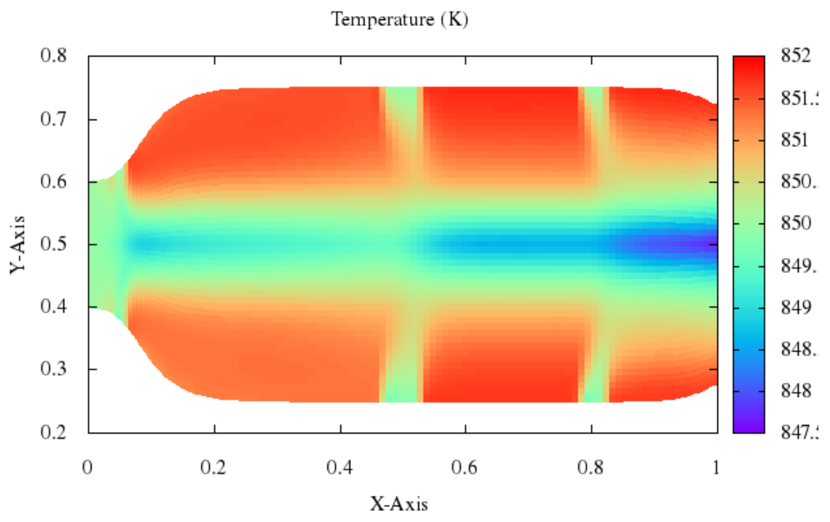
(a) Mach number contours



(b) Density contours

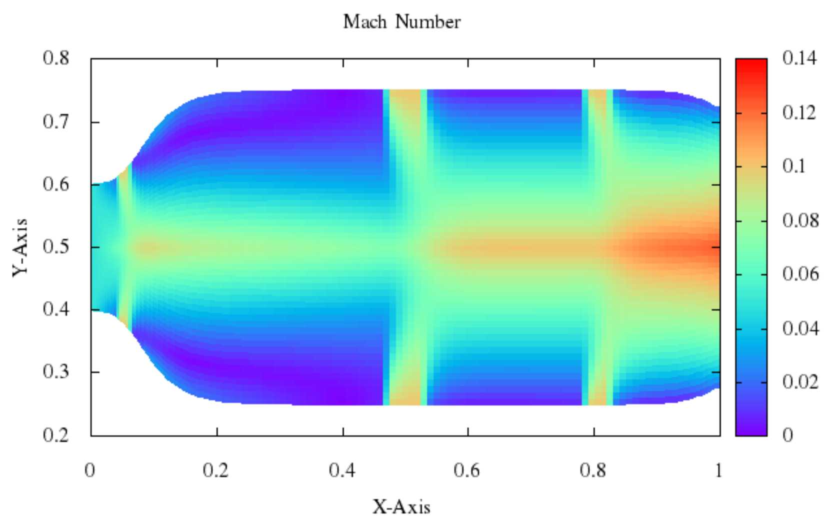


(c) Pressure contours

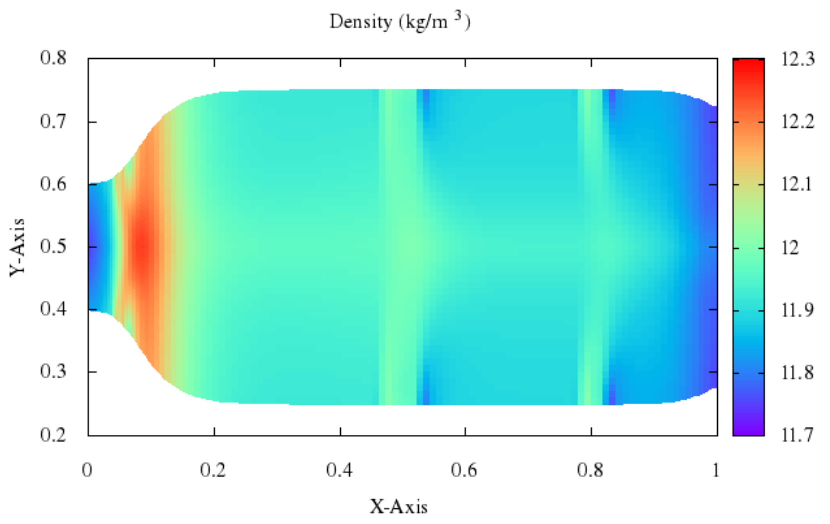


(d) Temperature contours

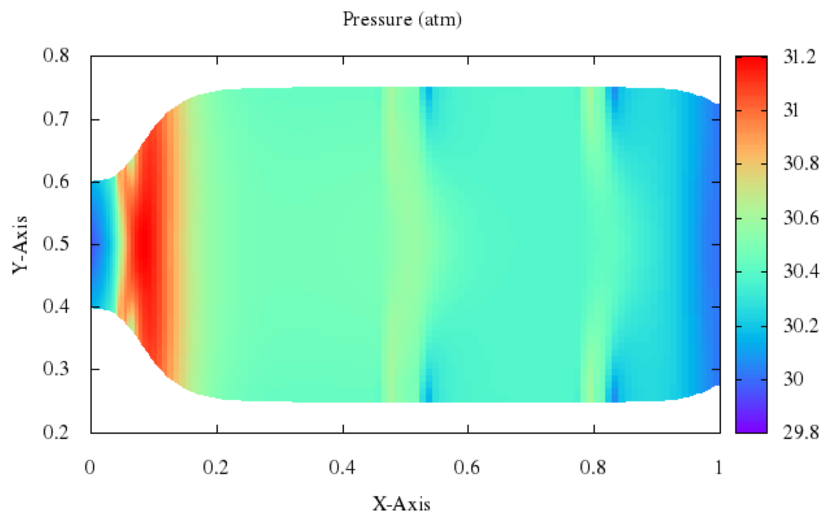
Figure 113: Non-reacting flow property contours for 110x50 grid of combustor can concept used for runs ($T_3 = 850K$).



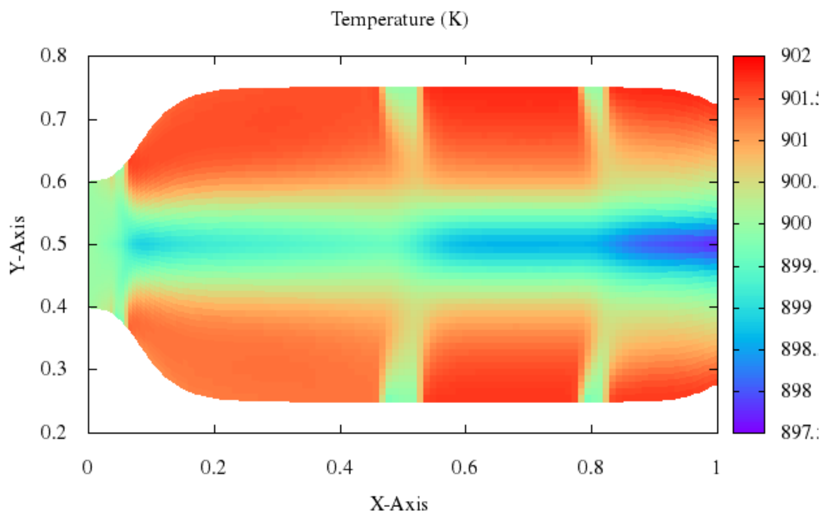
(a) Mach number contours



(b) Density contours

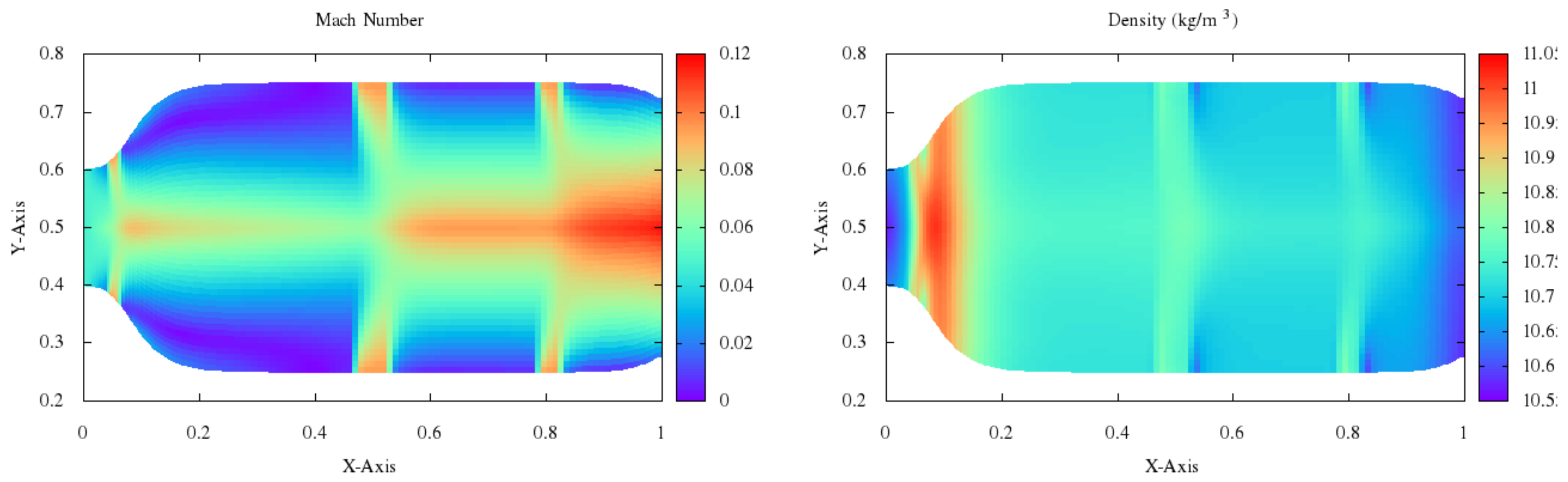


(c) Pressure contours



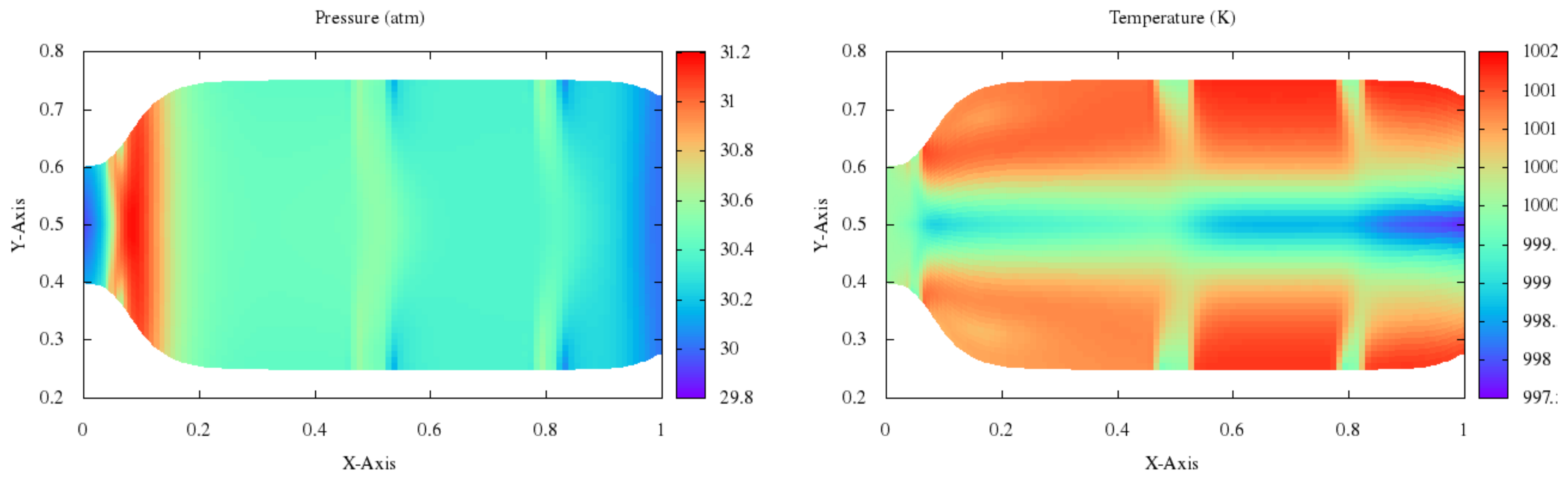
(d) Temperature contours

Figure 114: Non-reacting flow property contours for 110x50 grid of combustor can concept used for runs ($T_3 = 900K$).



(a) Mach number contours

(b) Density contours

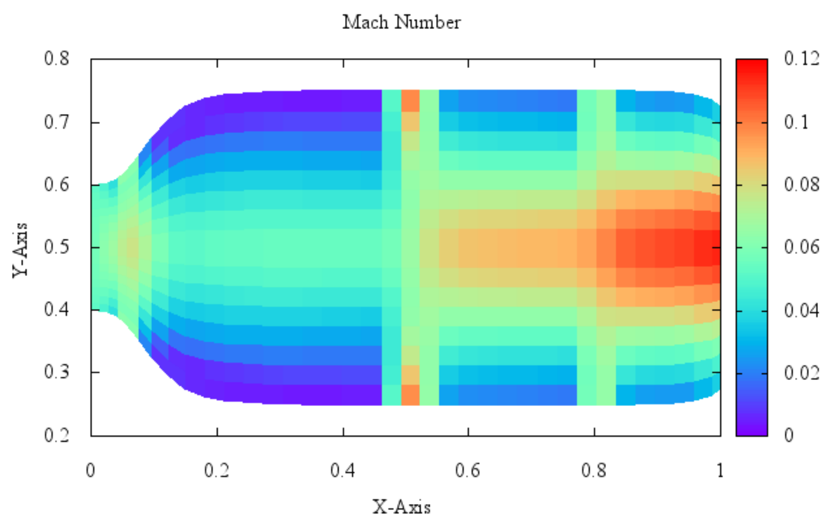


(c) Pressure contours

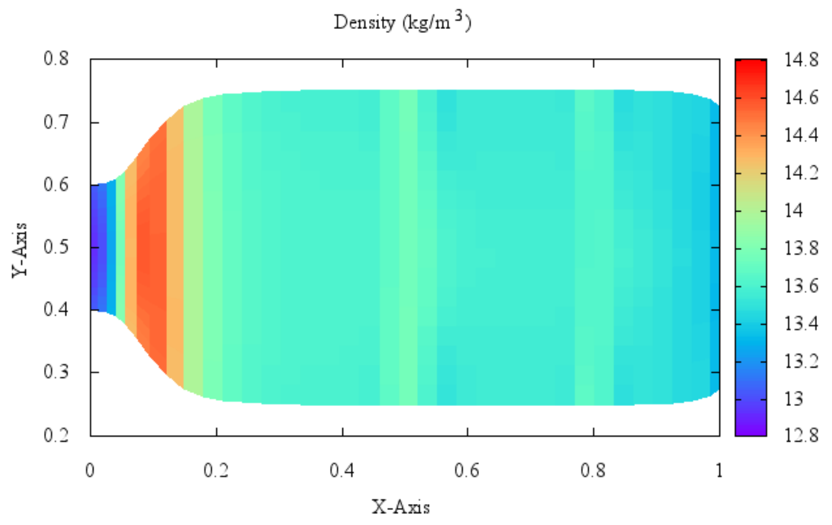
(d) Temperature contours

Figure 115: Non-reacting flow property contours for 110x50 grid of combustor can concept used for runs ($T_3 = 1000K$).

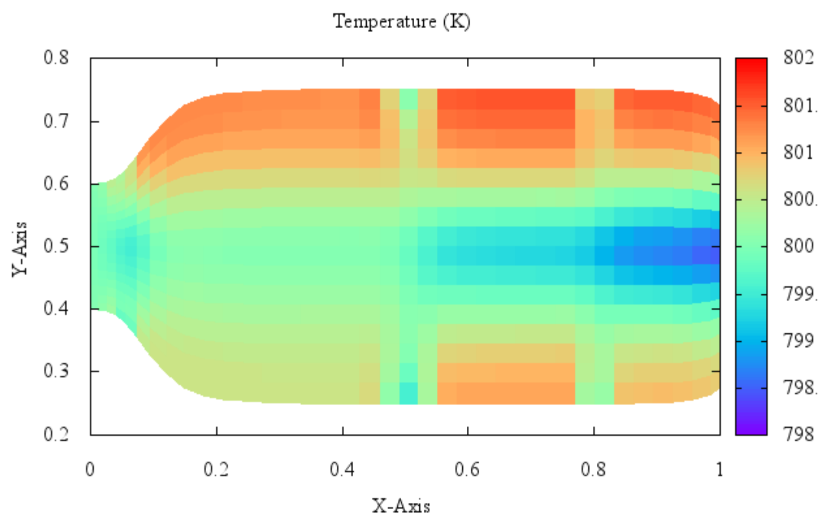
Though the 110x50 node grid is used for main comparisons against the RFCFD PPDF model, some select non-reacting flow solutions for other grids are also displayed below at $T_3 = 800K$.



(a) Mach number contours.

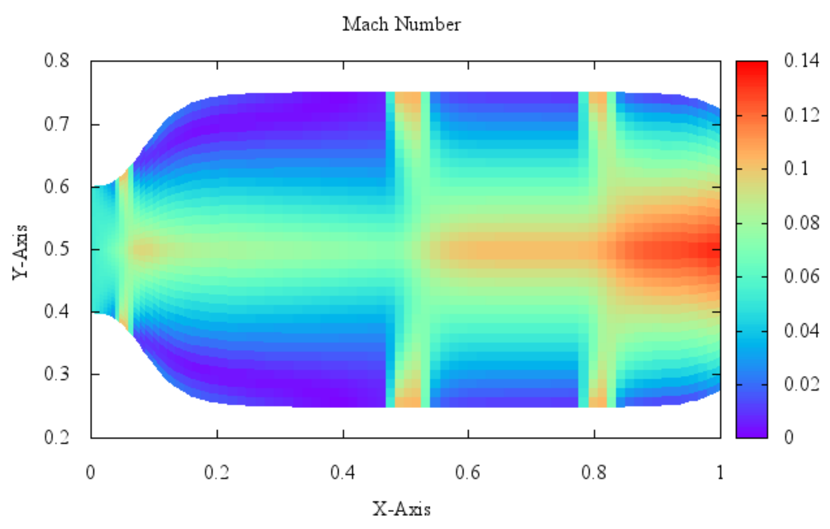


(b) Density.

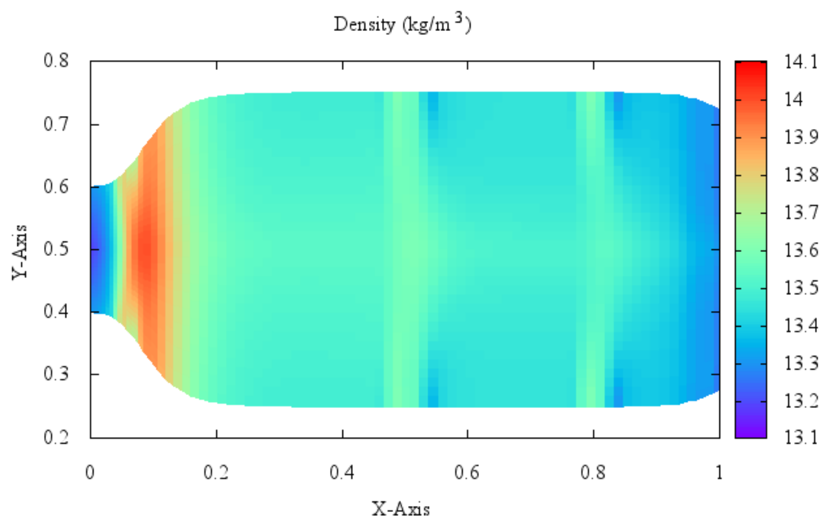


(c) Temperature contours.

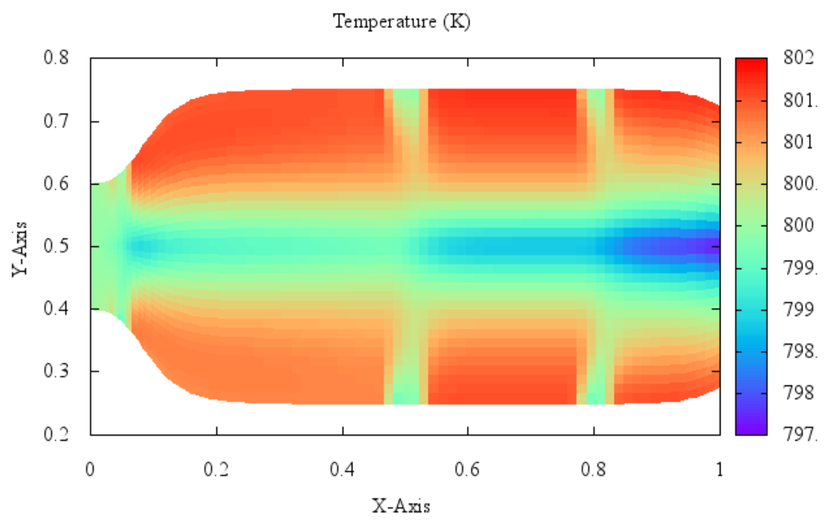
Figure 116: Property contours for 37x17 grid of combustor can concept used for runs ($T_3 = 800K$).



(a) Mach number contours.

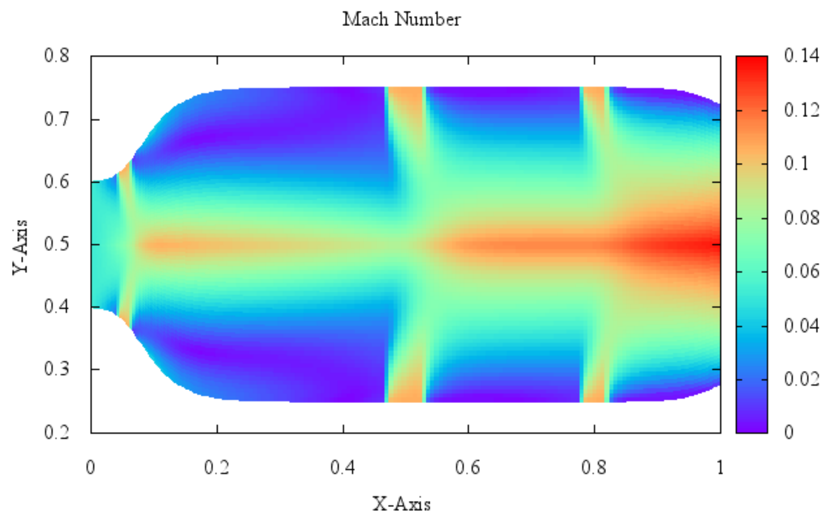


(b) Density.

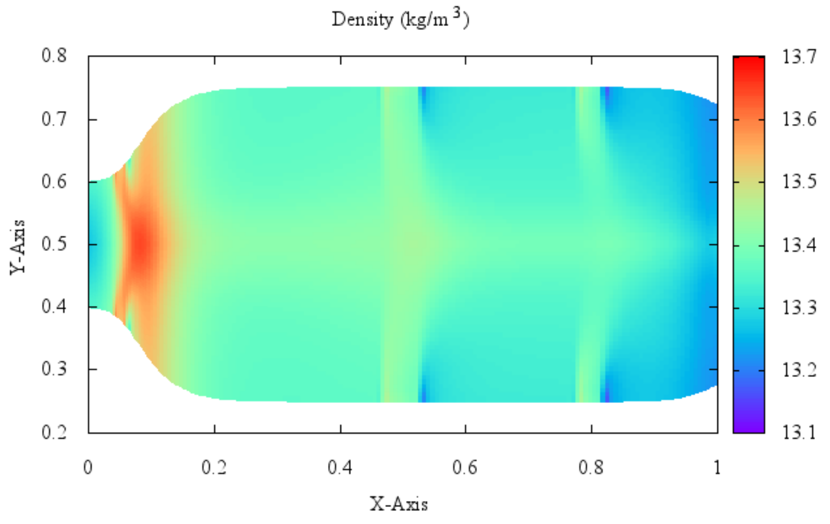


(c) Temperature contours.

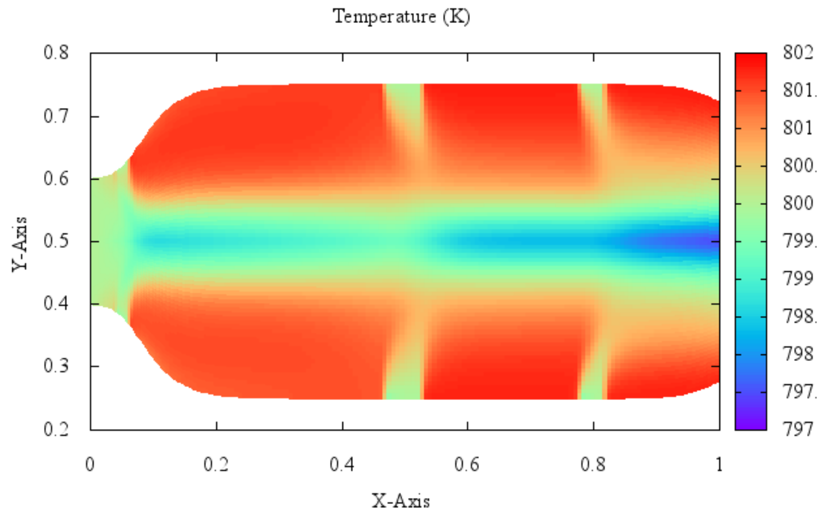
Figure 117: Property contours for 77x35 grid of combustor can concept used for runs ($T_3 = 800K$).



(a) Mach number contours.

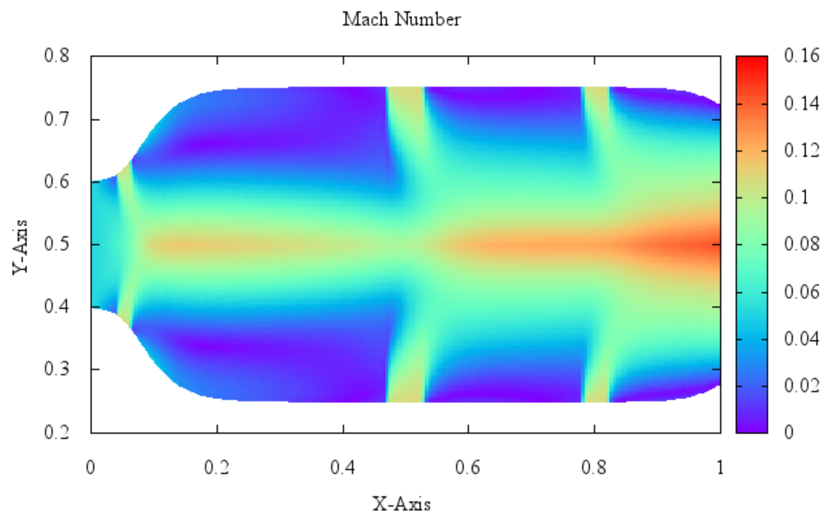


(b) Density.

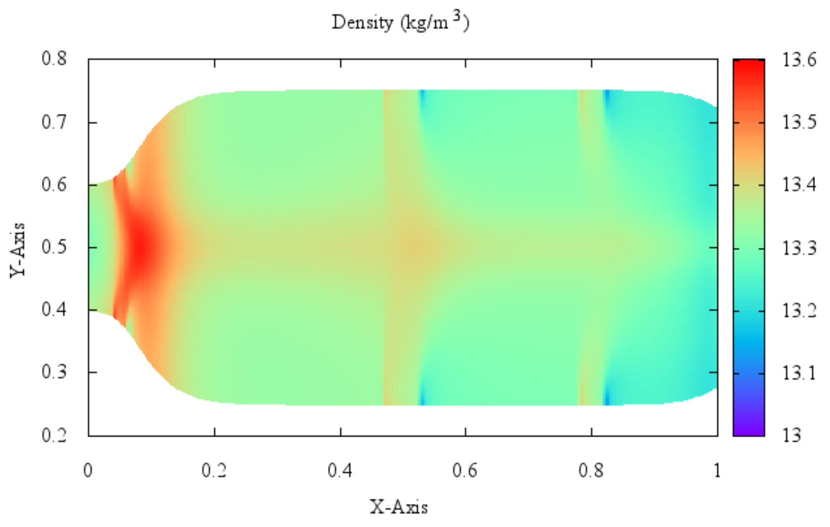


(c) Temperature contours.

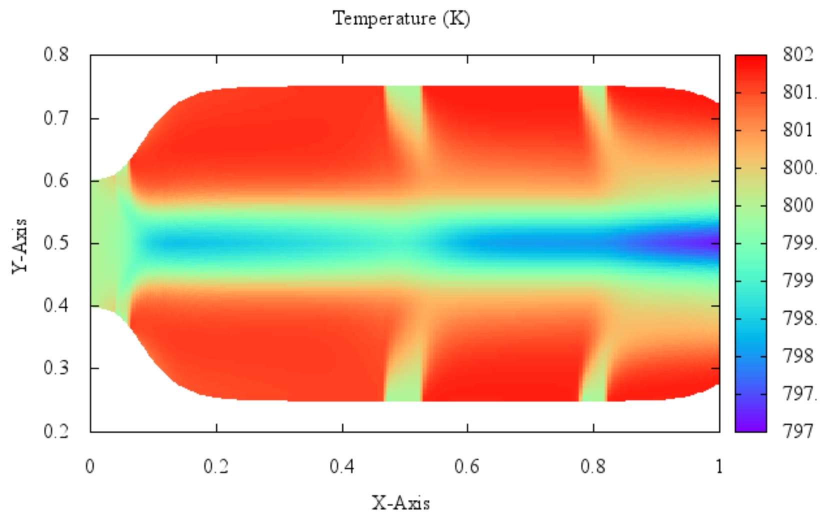
Figure 118: Property contours for 166x75 grid of combustor can concept used for runs ($T_3 = 800K$).



(a) Mach number contours.

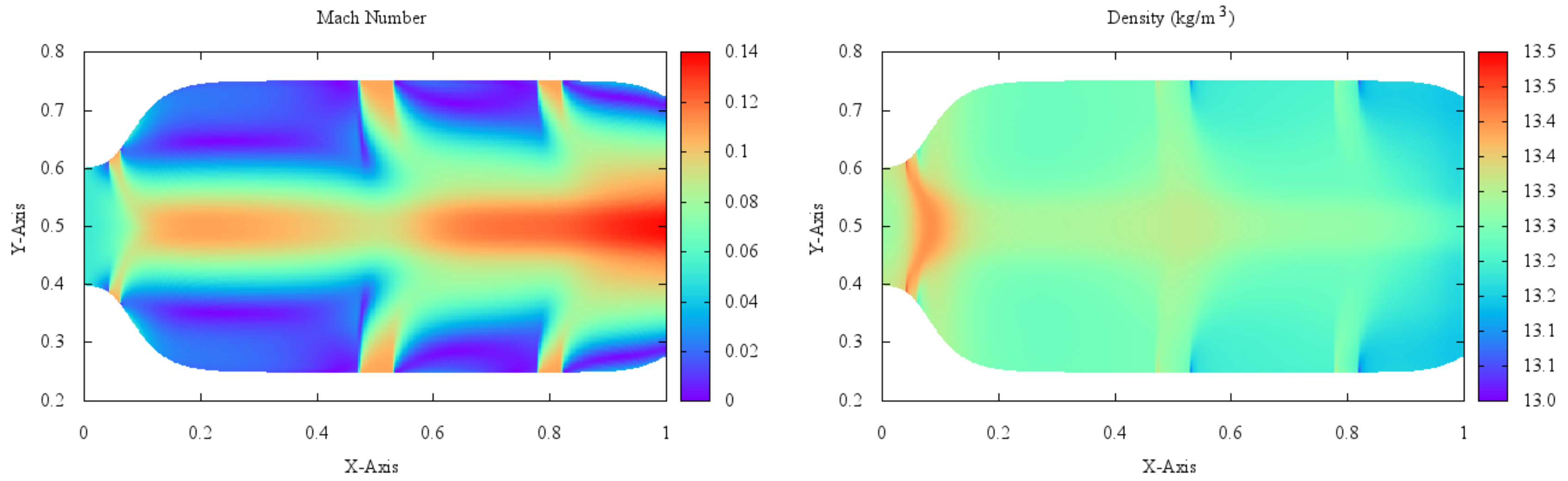


(b) Density.



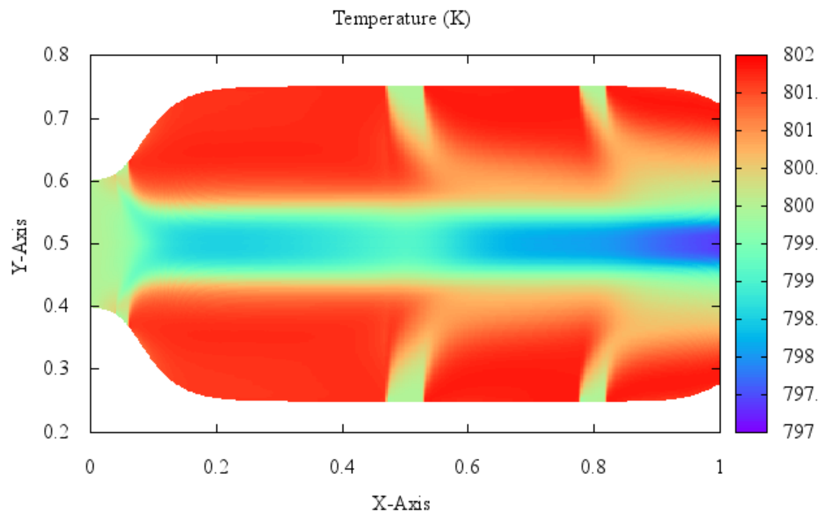
(c) Temperature contours.

Figure 119: Property contours for 222x100 grid of combustor can concept used for runs ($T_3 = 800K$).



(a) Mach number contours.

(b) Density.



(c) Temperature contours.

Figure 120: Property contours for 445x200 grid of combustor can concept used for runs ($T_3 = 800K$).

APPENDIX E

FUTURE WORK: SUPPLEMENTAL ALGORITHMS

This section contains supplemental material, including further thoughts about possible future extensions to the MoST algorithm, and the display of other algorithms developed as part of this work. The first few sections present information about how the method can be improved and optimized in its current structure. After, this chapter continues to present ideas for modifying certain parts of the algorithm to handle more general grid structures, in addition to ideas about how to lower the reactor count. These ideas involve more advanced simulation techniques, and are presented for consideration.

E.1 Steady State PSR Network

Section 6.2 briefly mentions the possibility and extension of the work to include steady-state CFD solvers as a means of decreasing runtime for steady solutions. In addition to the CFD side, steady solvers may help tremendously when paired with steady reactor networks. Steady-state PSR models exist in certain kinetics codes such as CHEMKIN, which in theory operate much faster than having to run an unsteady kinetics solver to steady-state (as must be done in Cantera). In the steady-state model, the time derivatives disappear, so that the ODEs in Equations 41 through 43 turn into algebraic equations (68 through 70).

$$\dot{m}_{in} = \dot{m}_{out}, \tag{68}$$

which is the steady-state conservation of mass equation,

$$\tau_{res}(Y_{in} - Y_i) + \frac{\dot{\omega}_i \bar{W}_i}{\rho} = 0, \tag{69}$$

which are the steady-state conservation of species equations, and

$$\tau_{res} \sum_i Y_{i,in}(h_{i,in} - h_i) - \frac{1}{\rho} \sum_{i=1}^{N} h_i \omega_i \bar{W}_i + \frac{\dot{q}_{in}}{\rho} = 0, \tag{70}$$

which is the steady-state conservation of energy equation. Cantera does not yet support steady PSRs, as algebraic solvers are difficult to implement robustly and involve advanced algorithms. However, their use allows one to expedite the solution process since one need not wait until steady-state solutions to ODEs are found, since each algebraic solution is the steady-state solution. With this, one only need obtain residence times to run the steady-state PSRs. The combination of an implicit steady CFD code and a steady-state PSR network should be fastest and most useful when looking for steady-state operating conditions. Additionally, the CFD code lacks turbulence modeling, which may be a big help in discerning flow features. Both CFD and kinetics codes may be swapped in and out, but tied together using MoST.

E.2 Three Dimensional Extension of the MoST Algorithm

Another extension of the method is to broaden the scope to three dimensions. This is a natural extension of the work, since reactors are assigned physical volumes. The ideas and the algorithm are exactly the same, except the CKN now consists of a three-dimensional array of reactors and controllers. Integration still occurs along the cell edges for fluxes, etc., and the algorithm still operates as it does in 2D (Figure 121).

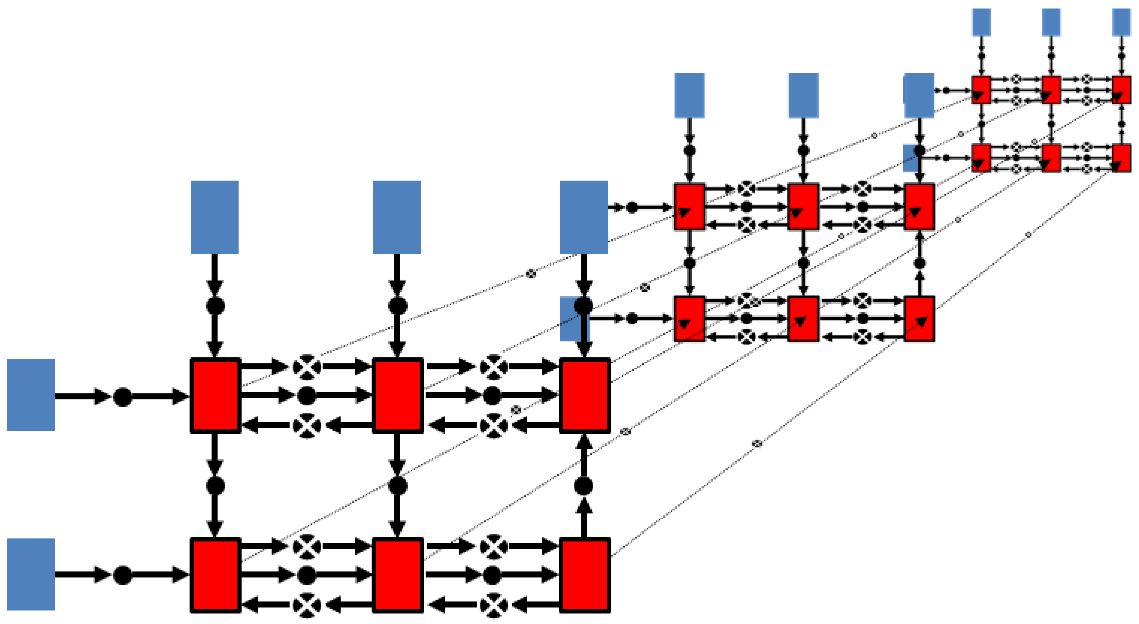


Figure 121: The MoST algorithm can be easily extended to three dimensions by utilizing a three dimensional array of reactors and controllers. Only a few reactors are shown here to keep the figure clean.

E.3 Optimizing the Iteration Scheme

Figure 44 in Section 4.3 shows the iterating scheme between the nonreacting CFD solver and the kinetics code. Within the MoST algorithm, fluxes between kinetics zones are recomputed at each and every time step. However, instead of recomputing fluxes between KM zones and inserting the source terms at each iteration, it may be more advantageous to run the CFD code to completion and then run the CKN to completion. This may prove especially useful if using a steady-state CFD code and a steady-state PSR network, as mentioned above. Running the CKN will give the steady-state solution for that specific CKN (with the original source terms inserted), but will not update the flow field. This will cause each CKN iteration to run faster, since the mass flow controllers are not being changed to new flow rates at each iteration. However, this will not couple the flow field with what is happening in the CKN as tightly as it was before. Instead, the steady-state solution from the CKN must be inserted into the flow field, which must converge again. This may take a longer time, however there may be some optimum number of iterations to wait between updating the mass fluxes, and future work should include finding this. This is the simplest change to avoid modifying any part of the actual code, and can be used as is.

E.4 Switching Reactor Types

In addition to the PSRs used in this work, different types of reactors may be used. For example, switching to PFRs may work well, however the equations are harder to solve and may take longer. A better approach may be to use steady-state PSRs as discussed previously, constant volume reactors, or a combination of reactor types.

In the case where steady-state PSRs are used, the residence time must be obtained, making the residence time calculation important since it tells the program how long the reactor network runs. Each reactor is connected to the same network, so all of the reactors run for the same residence time (unless of course the reactors are uncoupled from each other and are not in the same network in this case). Assuming they are in the same network, the smallest residence time is used in order to capture the temporal features most accurately. Solving for the residence time is a straight forward computation. First, one chooses two points on the grid between which the residence time is desired (the points representing the ends of the reactors), which comes in the form (x_1, y_1) and (x_2, y_2). Starting with the point (x_1, y_1), a straight line is drawn to the point (x_2, y_2). The next step is to draw a rectangular region around all of the points within the region connecting the two points which were input, as in Figure 122.

Next, all of the cells that the line connecting the points goes through (within the red square box) are determined. This is done by fitting lines between each of the top and bottom cell points to get equations for the lines. In order to do this, the slopes and a y-intercepts are computed. This is shown in Figure 123.

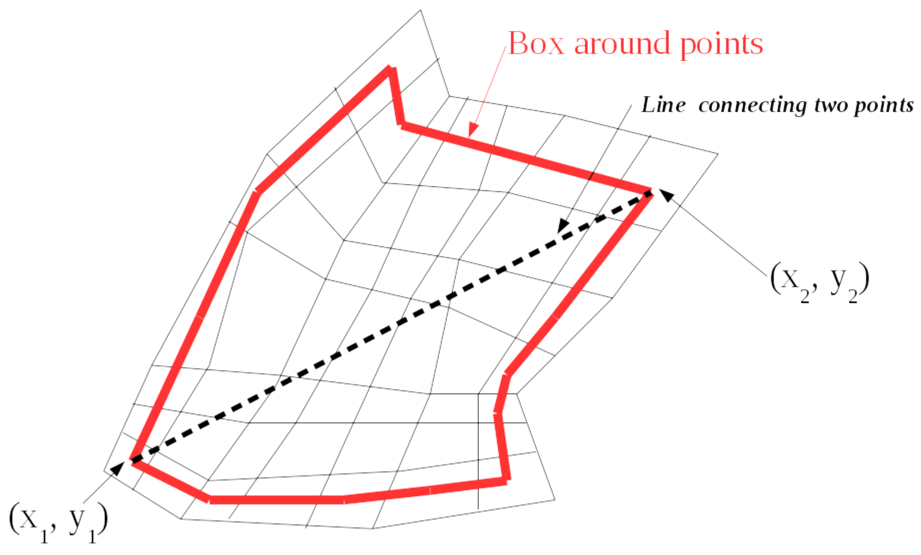


Figure 122: Computing the residence time between two points.

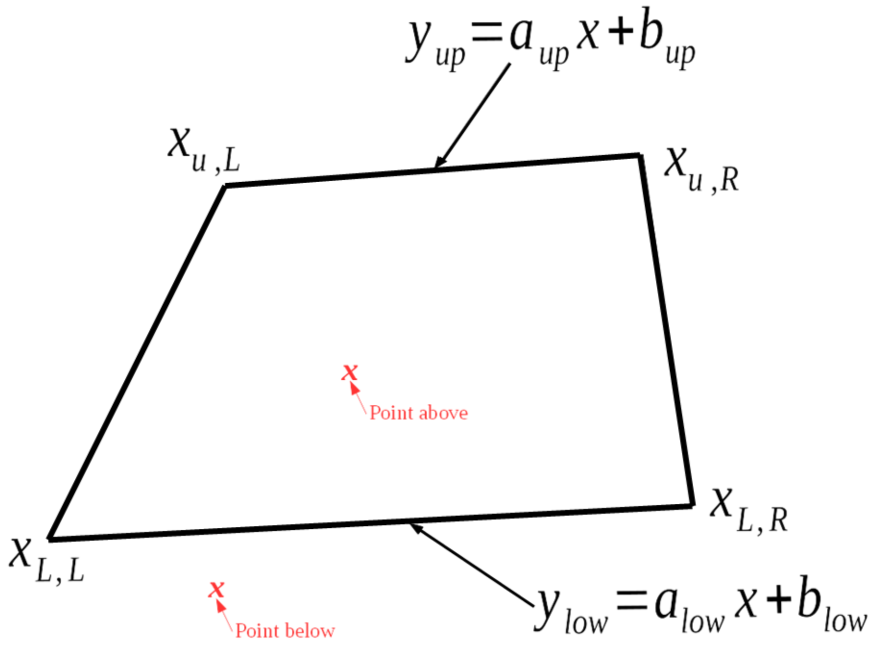


Figure 123: An individual cell in the red box in Figure 122. Equations for the top and bottom lines of the cells are found.

Utilizing the slope of the line, all of the points within in the rectangular region between the two given points are tested to see if the line passes inside or outside the cell. Figure 123 shows (in red) the points which would be considered within the cells and those which would be outside of the cells. To find if it is within the cell, test points are used. For x between $x_{L,L}$ and $x_{L,R}$, one determines if the test point lies above the line or below the line. Then, for x between $x_{u,L}$ and $x_{u,R}$, one again determines if the test point lies below the line or above it. If it turns out that the point lies within the cell, that velocity gets recorded.

After all of the velocities of the cells which get passed through are recorded, each component of the velocity is averaged. After, the Euclidean length between the two points gets measured. After both of these numbers are found, the equation

$$t = \frac{d}{v_{avg}} \tag{71}$$

is used to find the residence time for flow between the two points.

Additionally, one can average the velocities for all of the points within the red box in Figure 122, not just the ones which intersect the line. This may be useful when it comes to recirculation regions which are generally elliptic in shape (and can be approximated as rectangles). Though it may not be used, but the option is there.

E.5 Unstructured Grids

Though it depends on the problem, unstructured KGs have the ability to save copious amounts of computational time and burden. This may or may not involve using an unstructured CFD mesh. If an unstructured CFD mesh is employed, the KM can still lay on top of the CFD mesh, and a one to one mapping can be used between the two meshes. If this is not the case (ie., a structured CFD mesh with an unstructured KM), this is still okay, however more care must be taken. In this case, the same algorithm still holds, where mass flow rates are integrated along the boundaries. Here the boundaries are determined in the same way, however each KM cell may have a different shape and size. The downside to this is the fact that loop vectorizations will prove more difficult since connectivity is no longer in array order [141]. In addition to storing thermodynamic and spatial properties of each KM cell, information about connectivity must also be stored so the KM cells know how to communicate. Looking at it a different way, and unstructured KM can also be created using the same algorithm in Chapter 4.3 with the addition of another function. After the MoST algorithm splits zones, they may be recombined if they are next to each other, even though they were originally created due to differences in other parts of the flow domain. Taking Figure 52 as an example, zones $0, 0$ through $0, 4$ can be created due to differences in mixture fraction between zones $0, 4$ and $1, 4$, however zones $0, 0$ and $1, 0$ may still be very close together in terms of mixture fraction, and do not need to be split. Therefore, they may be split initially and recombined afterwords get an unstructured KM.

Along the same lines, all kinetics zones may be placed as precepted by the MoST algorithm, however certain reactors can be turned off if they do not offer any substantial chemical input. For instance, zones on the outskirts of the combustor (by the wall) may only have air, and therefore will not react with any fuel. Since connectivity must be maintained between zones, these reactors may still be placed, but chemistry turned off. This allows the flow to proceed and mix through the zone, yet the energy equation is not solved here since the mixture fraction would be too low to give a substantial heat release. The same may occur for the fuel stream down the center of the aforementioned combustor in Figure 66. This has the potential to save more time.

E.6 Recirculation Regions

One can greatly expedite the process of obtaining a solution by placing reactors in locations where there are only reactions and major flow features, as in Figure 45(b).

As explored in Chapter 4 above, this alone is not a complete idea, since calibration is not desired. However, calibration may still be avoided if the amount of flow in each direction (to each reactor) is known. If this is the case, many fewer reactors may be necessary. In order to obtain the positions and the amounts of flow to each reactor, particle tracking methods may be used. Here, the amount of flow into a given reactor must be computed based on the particle paths in the flow field, as well as how much is bypassed. One large issue is to determine where recirculation occurs, and if it is necessary to input more reactors in that region. Recirculation regions are important aspects of well designed combustors, since it is the main mechanism behind flameholding. Figure 124 shows a standard combustor with a recirculation region, and Figure 125 shows a zoomed version of the recirculation region.

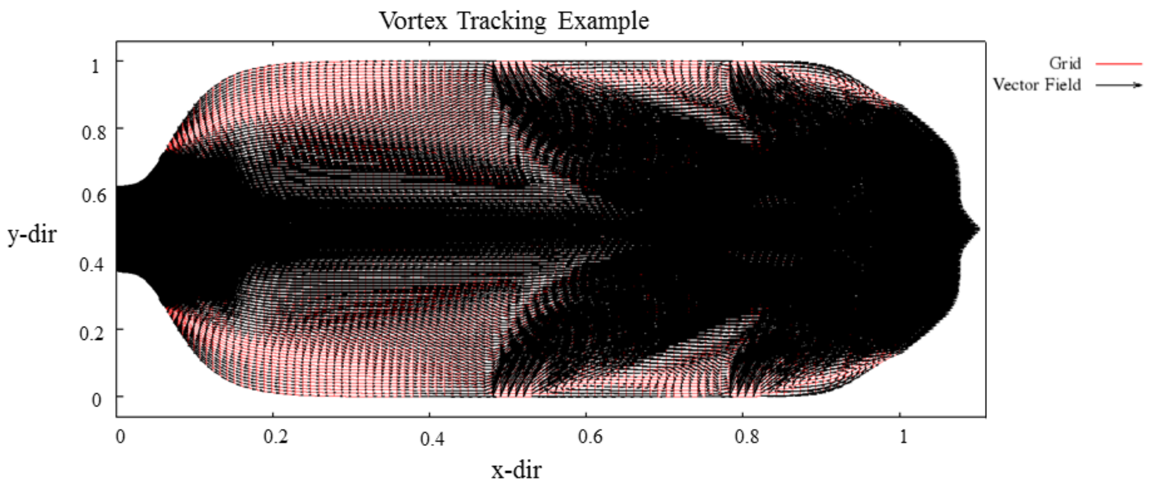


Figure 124: Entire 2-D combustor geometry that contains recirculation regions. The figure aspect ratio is skewed to show the arrows representing flow directions (the flow goes from left to right). The flow directions are stored at the center of the cell, but put on the cell edges to make the figure more clean.

The black lines in Figures 124 and 125 represent the velocity vector within the given cell, so a longer line means a higher velocity (these are results taken from one of the square combustor test cases). As shown in Figure 125, a recirculation region necessarily means a lower velocity, since the velocity is necessarily zero at the center

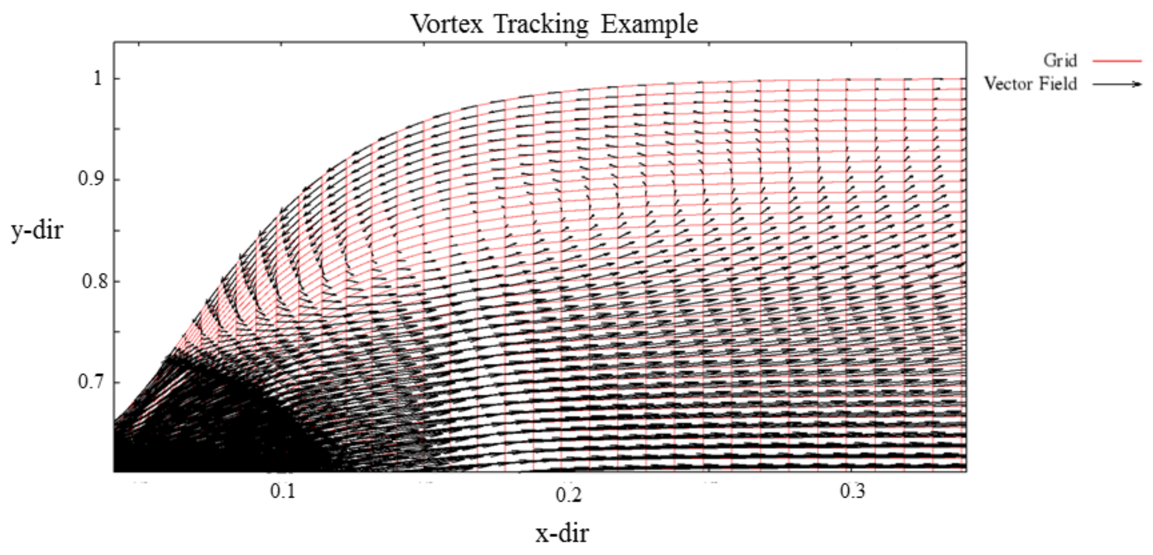


Figure 125: Zoomed in scale of one of the combustor's recirculation regions. A counter clockwise recirculation region can be seen. The figure aspect ratio is skewed to show the arrows representing flow directions. The flow directions are stored at the center of the cell, but put on the cell edges to make the figure more clean.

of an actual vortex region. Therefore, the entire combustor (and boundaries) can be viewed as a "design space" which may or may not contain recirculation regions. An easy way to find the center of the recirculation region is to solve an optimization problem to find a local minimum velocity, since this is a necessary condition (but not sufficient) for the presence of a recirculation region. The optimization problem can be stated as

$$\begin{aligned} \underset{\vec{u}}{\text{minimize}} \quad & f_0(\vec{u}) \\ \text{subject to} \quad & f_i(\vec{u}_i) \leq b_i, \; i = 1, \dots, n, \end{aligned} \tag{72}$$

where $\vec{u}$ is the velocity and b_i are the boundaries. Therefore, any minimization algorithm can be run to find local minima, which exclude points that are not potential recirculation zone centers. Additionally, there is another caveat that a recirculation region cannot occur at a wall, since flow cannot go back in the other direction past a boundary to complete the recirculation zone.

The first algorithm used for the minimization problem is a compass search, since it is easy to implement, does not require gradient information, and will "fall" into local minima traps (in this case, a desired output!) To ensure the capture of all local minima, one should start from many points. To be entirely sure, a compass algorithm may be run from every single starting point in the interior of the grid to ensure that no local minima are missed. This was done using the same combustor geometry as in Figure 124, and the results are shown in Figure 126.

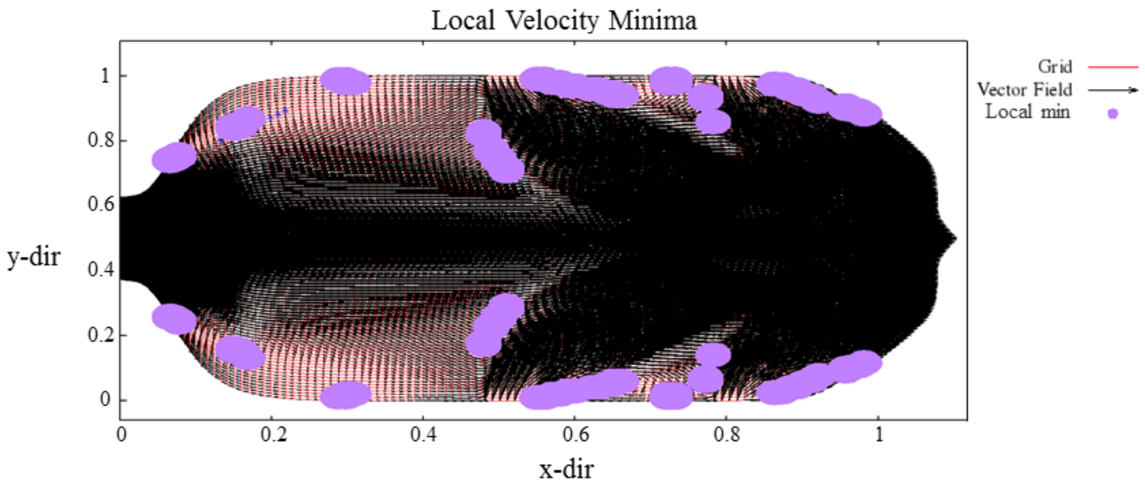


Figure 126: All of the local minima found when running a compass algorithm from every single interior node as a starting point. The cell is skewed to show all of the local mina so that the dots do not appear in the same place.

As can be seen, it captures many more features than just the center of the recirculation region, i.e. any local minimum within the flow field. Though the recirculation zone centers are not the only solution, this is still a good thing. Unlike many optimization problems, the interest is not the global optimum. Instead, the interest is as many local minima as necessary (although it can also mean just a local slow in velocity due to some other flow feature!) These other flow features are typically turns in the flow field or some other point of local minimization of velocity.

As it turns out, there is another parameter which can be used to calculate where a recirculation region occurs. Vorticity is defined as

$$\vec{\omega} = \vec{\nabla} \times \vec{u} \tag{73}$$

Note that this is a cross product, and the flow is only in two-dimensions. Therefore, the vorticity vector will always be in the z (or $-z$) direction in a Cartesian coordinate system. Each cell can be swept through, and the vorticity magnitude is simply

$$|\vec{\omega}| = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \tag{74}$$

A sample test case of a vorticity computation is shown in Figure 127.

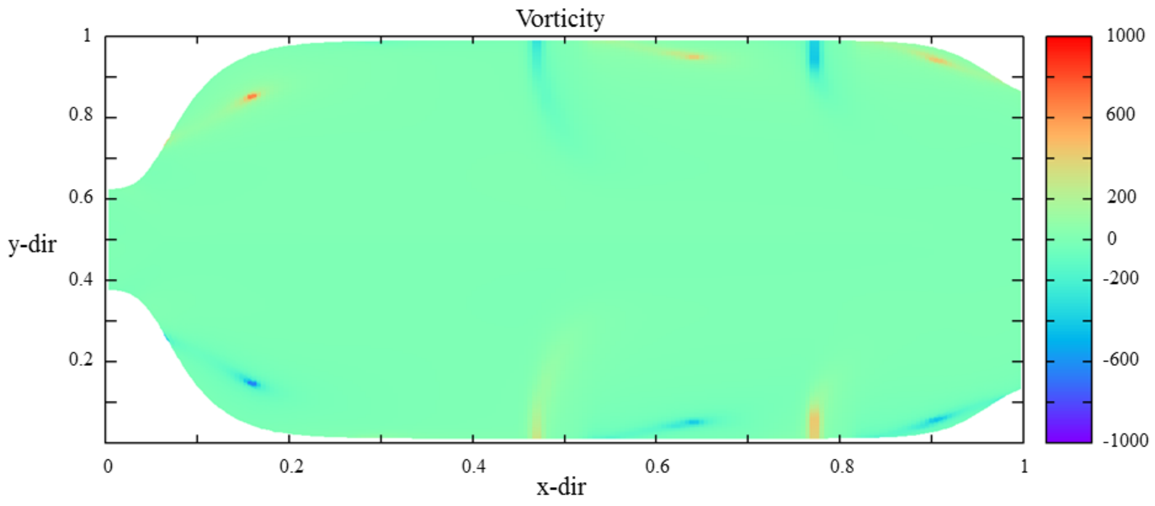


Figure 127: Value of the vorticity for a sample case combustor run. Note that the sign of the vorticity tells you which direction the vortex is turning the flow. This was done for a square combustor can, but stretched to show the vorticity contours.

To find the maximum vorticity and hence the recirculation center, one can use the same algorithm as above (an optimization algorithm such as a compass search, except the absolute value of the vorticity is maximized instead of the velocity minimized). Algorithms of this type already exist [140, 136, 144], however these algorithms only locate the vortex. For example, the vorticity can be computed which gives the location of a reactor center, but does not give the reciruclation zone size. The vortex size and location are sought for this work.

When running the optimizer to search for local maxima for the absolute value of vorticity, the computer actually tags many, as shown in Figure 128.

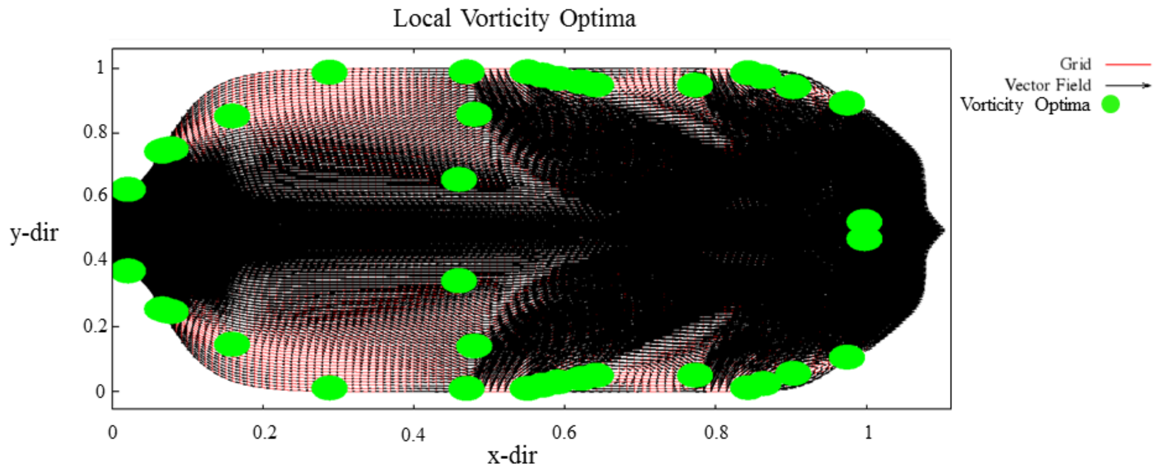
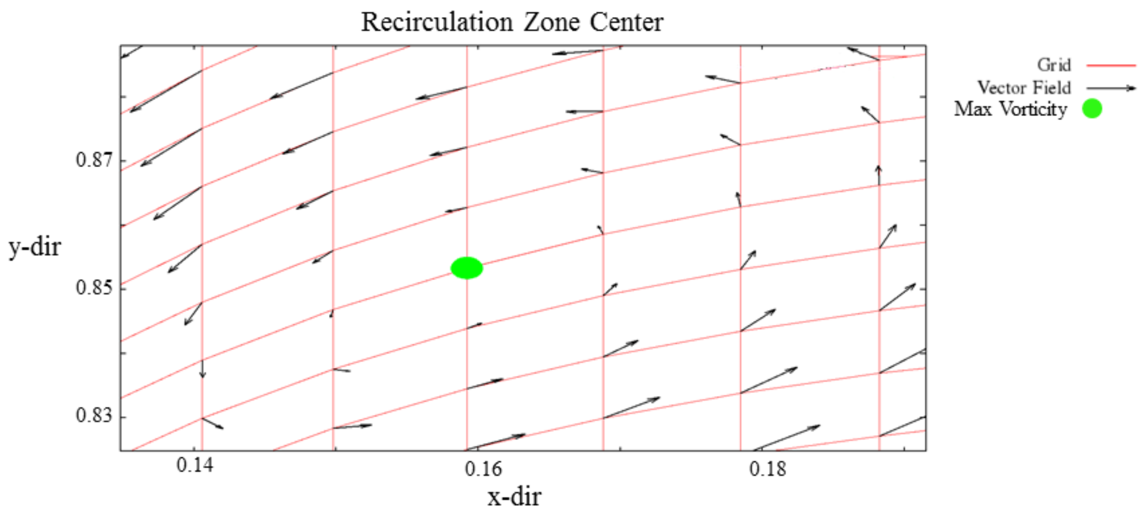


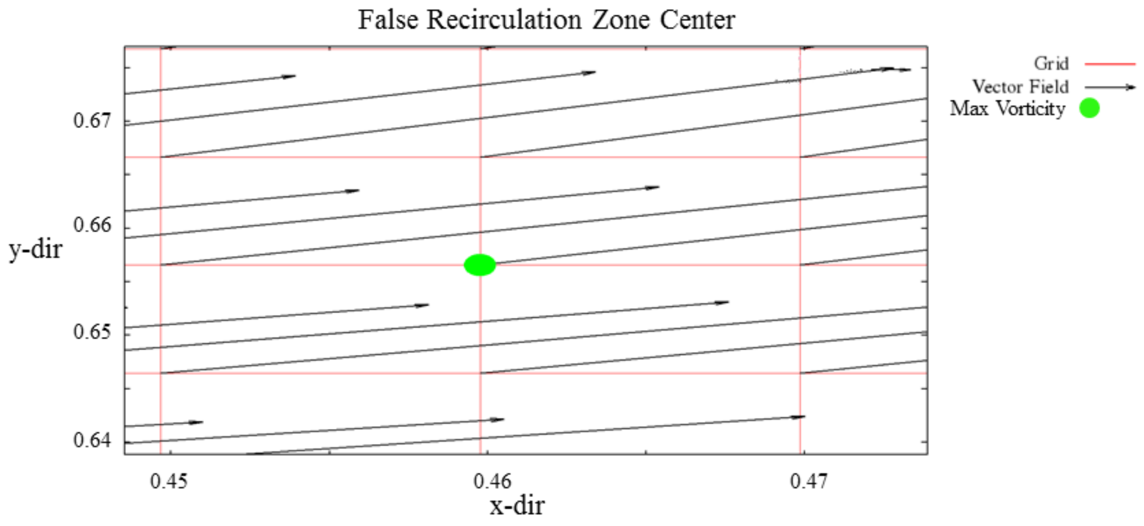
Figure 128: Value of all of the local maximum absolute values of vorticity. This is for a square can, but the mesh is skewed as to not cause points to overlap.

As can be seen in Figure 128, some of the points are the same as the minimum velocities, but some are not. Therefore, the maxima from the absolute value of vorticity and minima from the velocity field cannot simply be cross referenced to find the local recirculation centers. Instead, a different algorithm must be used in addition to both or either the local minimum of velocity or local maximum in absolute value of vorticity. Since the computer only tags a small subset of the number of points in the flow field (using both algorithms), a more direct method may be used to determine recirculation zone centers on top of the optimization algorithms. The next step is to perform a direct velocity vector search around each of the potential points to find out if the flow field makes a complete turn [140]. For example, consider Figures 129(a) and 129(b).

Here, it is very clear that the image in Figure 129(a) is a recirculation center and the image in 129(b) is not. Using this information, one is guaranteed to find the



(a) Zoomed in image of one of the recirculation centers.



(b) Zoomed in image of a local absolute value of vorticity maximum, but not a recirculation center.

Figure 129: Points that are maxima of local absolute values of vorticity or local velocity minima may not actually be a recirculation zone center. Square combustor can which is stretched in the figure so that it is shown clearly.

center of a recirculation region at least two cells in from the boundary of the domain. Here, each cell around the center is searched successively, and the angles between each consecutive velocity vector are tested. If the angle forms 2π radians, then this must be a recirculation zone center. If it does not, then it fails the test, and the point is thrown away. The point in Figure 129(a) passes the test, while the point in Figure 129(b) does not. Once this is completed for each point, the final results (the points that are actual vortex centers) are obtained. The results are shown in Figure 130.

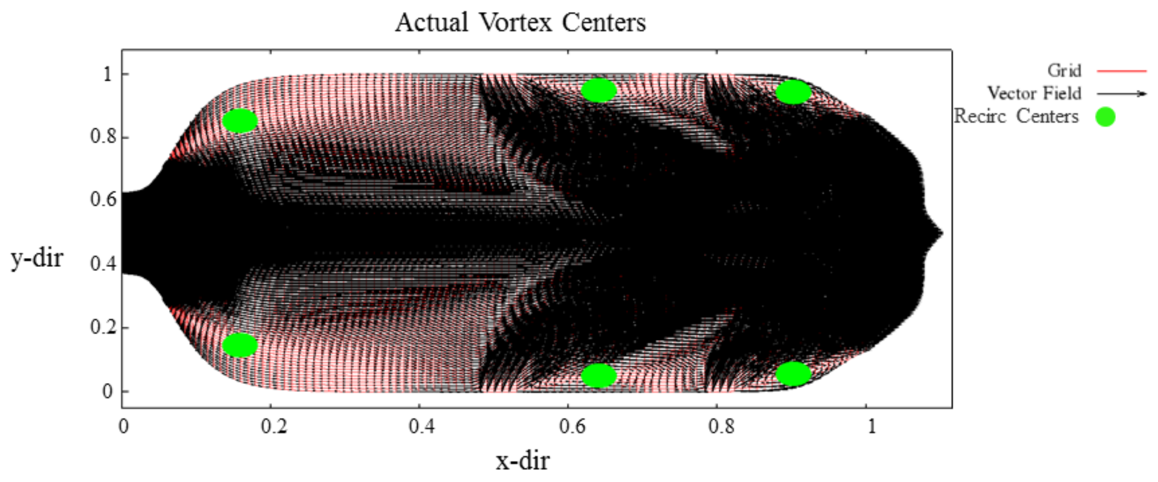


Figure 130: Actual vortex centers found by filtering points and then checking velocity flow vector angles. The grid is skewed for clarity.

The next step is to determine the size of the recirculation zone, which must be known in order to place reactors. Additionally, one must determine know how much mass flow to recirculate and how much to move out of the zone. In order to do this, a tracking algorithm is utilized which is based on the flow field. The algorithm is described below.

E.6.1 Points Lying withing Generalized 2-D Polygons

Before the actual algorithm is described, an algorithm is first needed to decide if a point lies within an arbitrarily shaped region or not, as it is useful later on. Many ideas were tested, including testing based on the corner points of the polygons, which ultimately failed. Specifically, all four edges are used in the test (since each cell is a quadrilateral, but the point may still lie outside even if it satisfies all edge criteria due to the uneven nature of the cell (note, this works for some shapes, such as a perfectly square cell, but does not work in general for a cell with skew. This is shown in Figure 131.

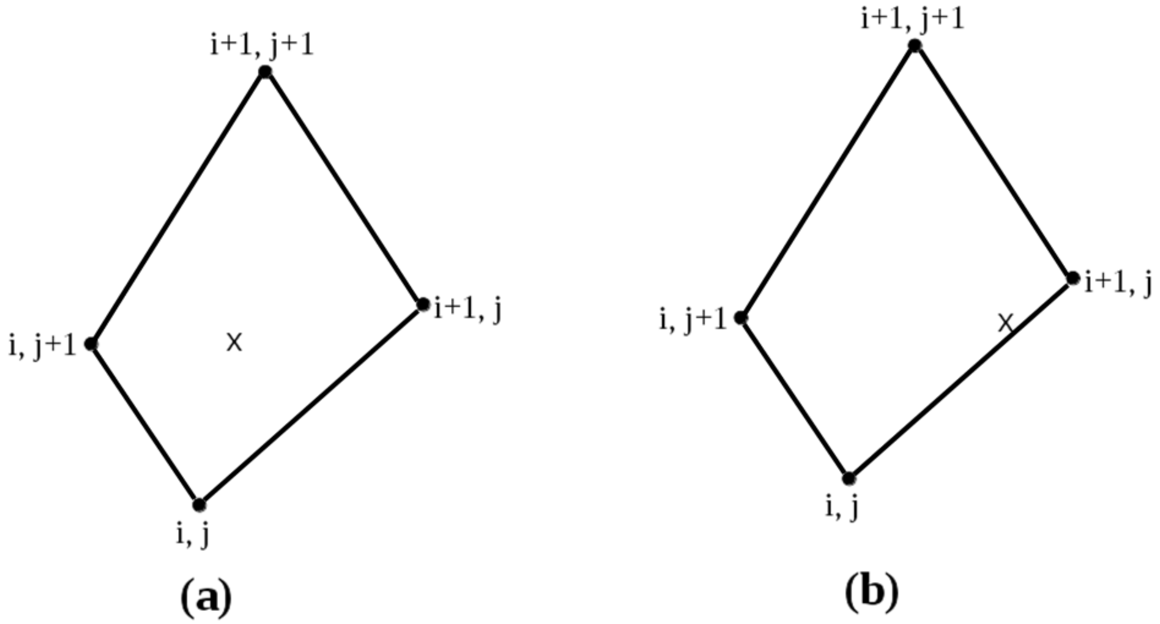


Figure 131: (**a**): If the point lies within all edges nicely, it may be captured by utilizing information from all four points. (**b**): This point does not satisfy all of the criteria even though it actually sits within the cell due to the cell's skew.

Therefore, a different method is necessary to determine if a point sits within a cell. Borrowing from the computer science/graphics literature, the winding angle algorithm works well [143]. The principle here is that the vector between any point and the corners of the shape sum to 2π radians if the point is within the box, however sum to 0 if the point lies outside of the box. This is shown in Figure 132.
The total winding angle is defined as

$$w = \sum_{a=0}^{a_{max}-1} cos^{-1}\left(\frac{\overrightarrow{v_i - P} \cdot \overrightarrow{v_{i+1} - P}}{|\overrightarrow{v_i - P}|\,|\overrightarrow{v_{i+1} - P}|}\right) \tag{75}$$

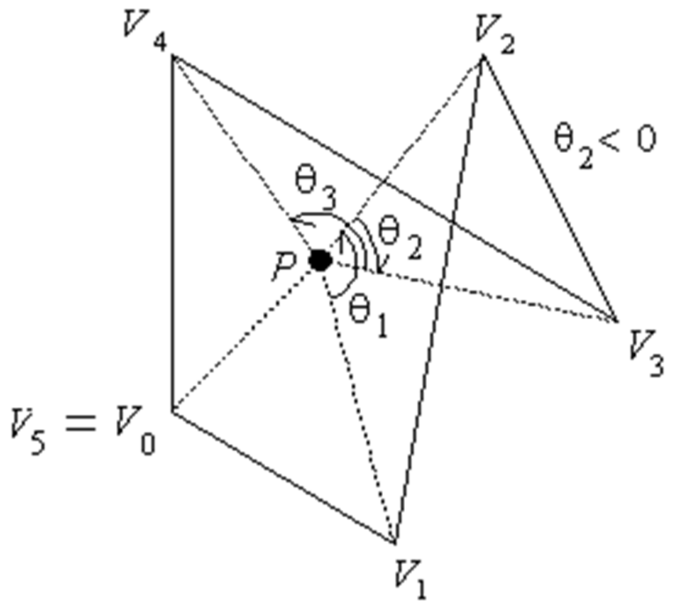


Figure 132: Any point within the box has a winding angle of 2π radians, and any point outside has a winding angle of 0.

where P is the point of interest. This is important since a time marching algorithm is used to determine the path of the flow (Section E.6.2), and hence the size of he recirculation region.

E.6.2 Vortex Algorithm

To go back to the original algorithm, each cell which is marked as a vortex center is taken as a starting point. A first order forward-time marching Euler Scheme is used to step through the domain and determine the size of the recirculation zone. The first order forward time stepping Euler method is shown in Equation 76.

$$\vec{X}_{new} = \vec{X}_{old} + \vec{V}(\Delta t). \tag{76}$$

Using this, the algorithm starts at the center of the recirculation region and steps in a given direction dictated by the velocity field. Each time the path created by the velocity field makes a complete circle, the points through which the path passes are saved by the code. These points are saved until the code runs out of the domain and are filtered afterwords. To summarize so far, the steps for the tracking algorithm are:

- Computed vorticity (and/or velocity in the entire domain.)
- Check areas of high vorticity for full recirculated flow using the points around it.
- Begin the particle track at the center of the recirculation zone.
- Use a first-order Euler forward temporal method to compute the particle's track.

To elaborate, each step is analyzed in detail. First, the algorithm starts from the center of the recirculation zone. The direction of movement is dictated by the local velocity field. To determine the next cell, the velocity for the current zone is used (since CFD algorithms for finite volume methods store the velocity in the center of the cells). When the next point is found in another part of the cell (or a different cell), an interpolation is used to find the velocity. Even if the algorithm lands in the exact center of the cell, a computer will not recognize it since floating data types do not recognize the same number using equality comparisons. Therefore, an interpolation for velocity is always used. Figure 133 shows how this is done.
As in the figure, the distances are computed using the standard equation in Cartesian coordinates. The inverse of these distances are used in combination with the velocities to get the velocity at point p as in Equation 77.

$$v_x = \frac{\sum_{i=1}^{4} v_{x,i} \frac{1}{r_i}}{\sum_{i=1}^{4} \frac{1}{r_i}}, \tag{77}$$

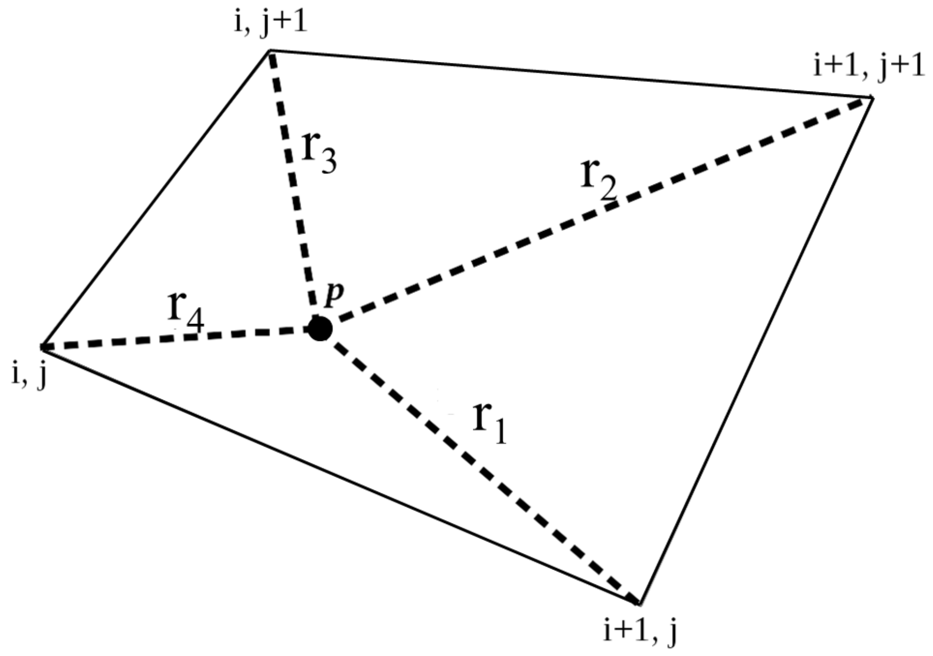


Figure 133: A velocity interpolation is used since landed in the cell will never be in the center where the state vector is stored. The velocity used at the point p is based on the inverse distances from all other points.

and the same for v_y. Note that the distances to the cell edges are used here, whereas the velocity is actually stored in the center of the cells. This is done for computational speed, but may of course be changed.

The distance moved in a given direction is dictated by the time step, Δt. A large Δt is always used (based on testing, $\Delta t = 1$ is a good starting point since the velocities are typically on the order of meters per second for the very outer parts of a recirculation region of a combustor). This commonly gives step sizes on the approximate order of meters. This is obviously way too large and can be cut down (and indeed is shortened during execution). If the step brings the area of interest outside of the 8 surrounding cells of the current cell, the algorithm automatically cuts the step size in half until the next successive step is within one of the eight surrounding cells (Figure 134).

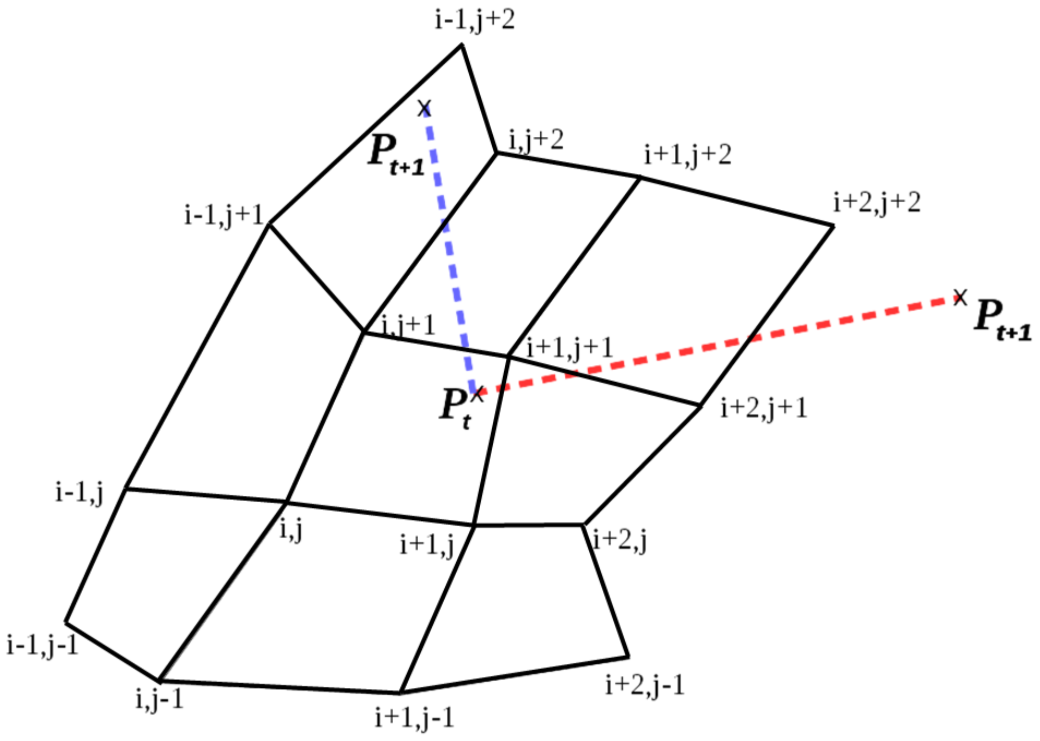


Figure 134: Good (blue) and bad (red) step sizes.

In Figure 134, the step with the red line is rejected by the algorithm. Instead, the step size is cut in half and a step is taken in the same direction, but much closer to the original point. The step with the blue dashed line is an example of a step which is accepted since it lands within one of the eight surround cells. Once the step size/velocity field combination brings the flow field to the edge of the domain, a maximum amount of cell recalculations are allowed before the algorithm finishes and allows the particle path to run out of the domain. At this point, the flow field tracking is complete. Each new cell that is entered in this step of the algorithm gets stored in an array for processing later.

As an example of the particle tracking method, Figure 135 shows the particle paths starting from the recirculation center all the way out of the domain.
The algorithm works very well on this design. It is tested again with the addition of a dilution jet, and the results are shown in Figure 136.

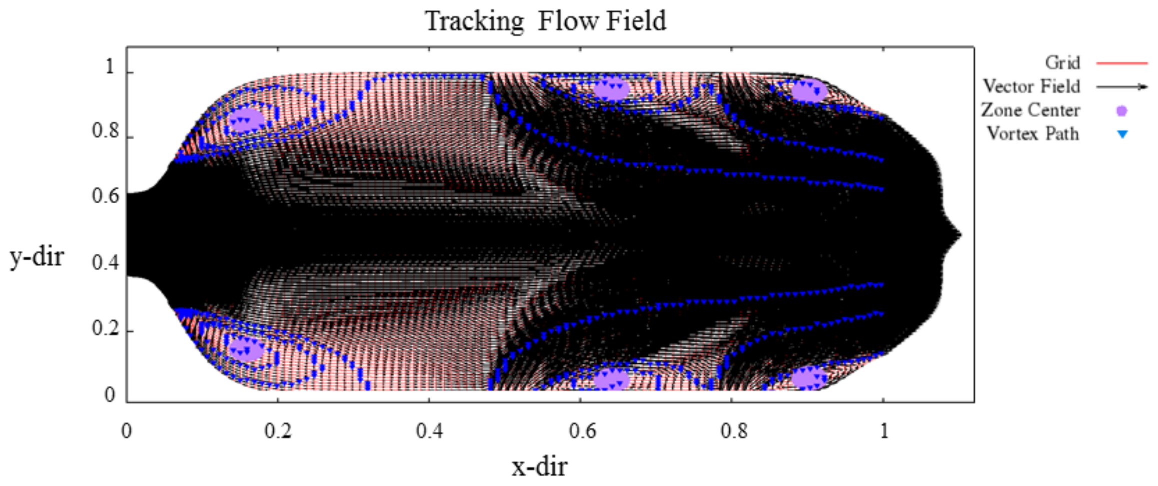


Figure 135: Equations 76 and 77 are used to capture recirculation by tracking the flow field starting from the center of a recirculation zone. The grid is a square, which has been skewed to show clarity in the particle tracking lines through the domain.

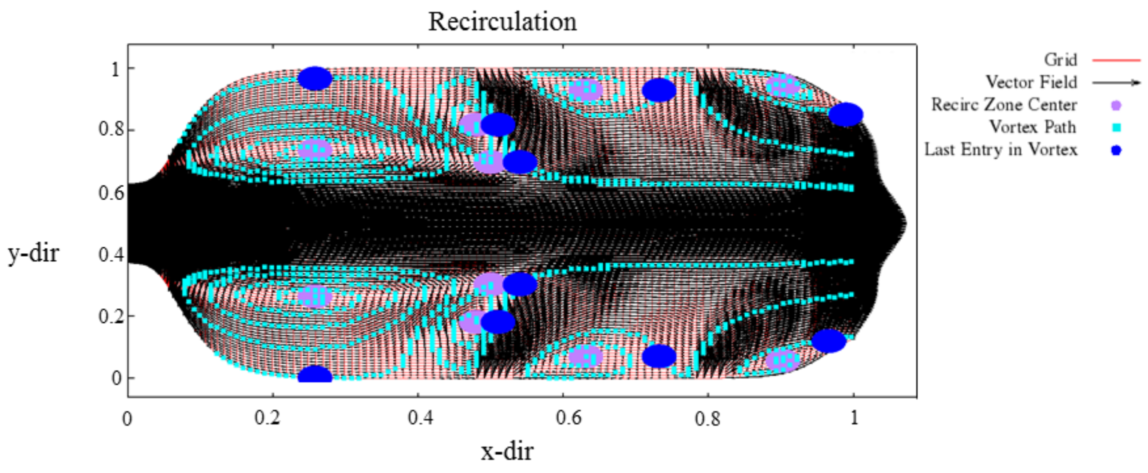


Figure 136: Recirculation is tracked again (for a differnt design than in Figure 135). Here, another dilution jet is added at $x = 0.05$, and this design has more recirculation zones. The grid is a square can which is warped to visualize the flow lones along the x-direction.

It appears that two recirculation zones overlap in Figure 136. Figure 137 shows a closer look at the lower one.

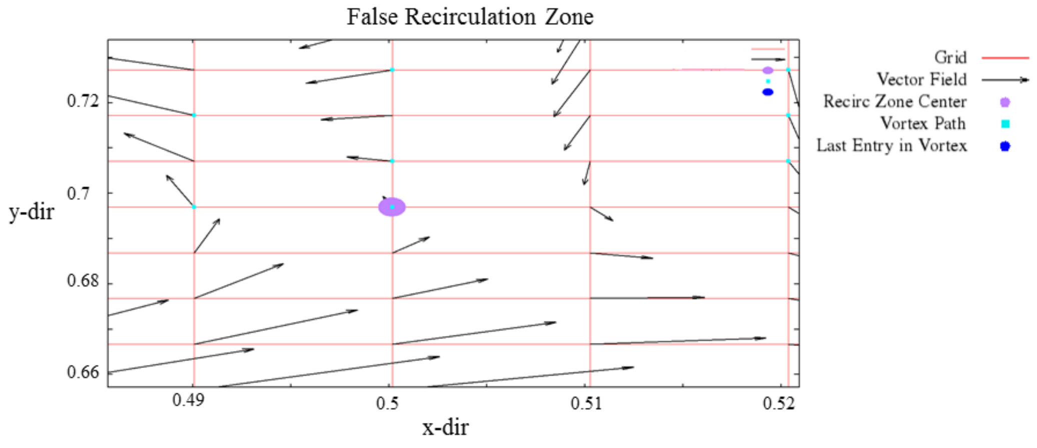


Figure 137: Zoomed in view of an area which was tagged as a recirculation zone (purple circle), but the flow vectors around it do not show that recirculation should occur. The grid is a square can which has been warped for clarity.

Based on the flow vectors in Figure 137, it does not look like recirculation should occur. However, tracking pathlines in Figure 136 shows that this line actually does recirculate, however only in the region above. Therefore, this area is left as a recirculation zone.

There is one additional pitfall which may arise. If there is slip at the walls, there can be what looks like very small recirculation zones at the wall. Figure 138 shows this.

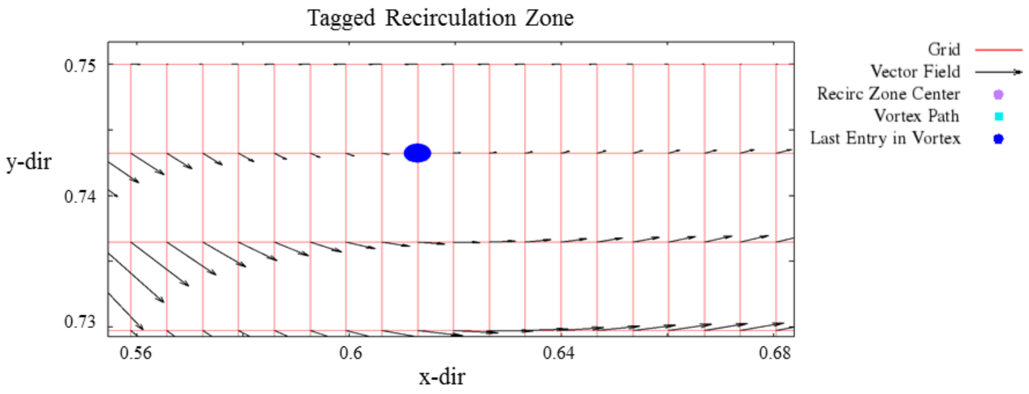


Figure 138: The blue dot is the tagged recirculation zone. You can see arrows moving to the right on the second line down, and arrows moving to the left on the upper wall.

When this occurs, the single point is left as the entire recirculation zone. As of now, this is not viewed as an actual recirculation zone, since wall slip is nonphysical.

It is very important to understand that this algorithm uses a first order Euler time stepping method to solve the ODEs for position as a function of time. One could very easily move to a higher order method (for example, a fourth order adaptive timestep Runge-Kutta routine), or even reduce the time step in the algorithm to make it more accurate. When this occurs (i.e., when it is very accurate), the flowfield forms a closed loop, as shown in Figure 139.

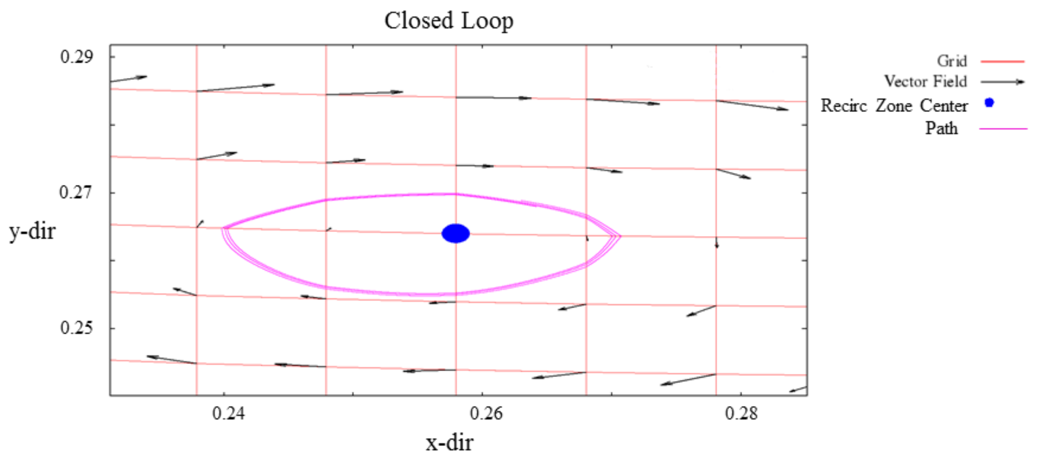


Figure 139: Lower time steps and more accurate methods cause the flow path loop to be closed.

This behaves as it is supposed to, however this is not the behavior that is intended. A higher order method would be used if a closed loop is desired (as is the situation for orbits [132, 133]), but this is not the case. Here, one seeks to take advantage of the numerical error between time steps in a recirculating vector field, so the movement is spatially outwards to track progress. It just so happens that the vector field increases in velocity away from the vortex center. With the accruing of numerical error, the particle path is therefore pushed outward, which is the desired outcome of this method. This is one of the few instances that numerical error actually works to one's advantage.

After each time a step is taken and a new position is found, the i, j combination is entered into a vector and stored for post-processing of the recirculation zone. This vector is then searched for the last entry which was recorded. Specifically, the post-processing code keeps track every quarter revolution from where the recirculation zone begins. If it hits a quarter revolution and then proceeds somewhere else in the flow field, the counter stops. Figures 140 and 142 show examples of this technique using different combustor geometries.

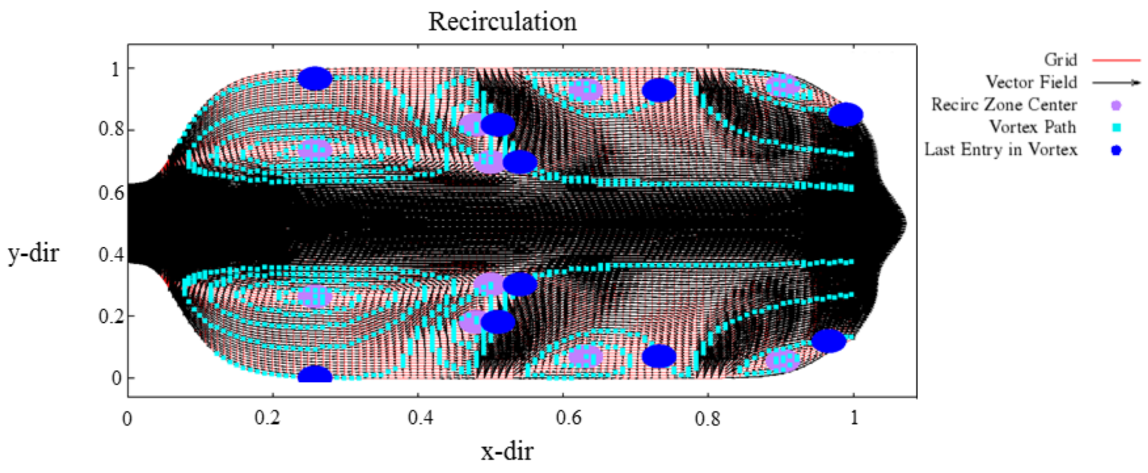


Figure 140: The blue circles show the ending point of the recirculation zones. This blue circle is the last point which will be counted as part of the "recirculation zone". The can is square, but the grid is stretched to better display the paths.

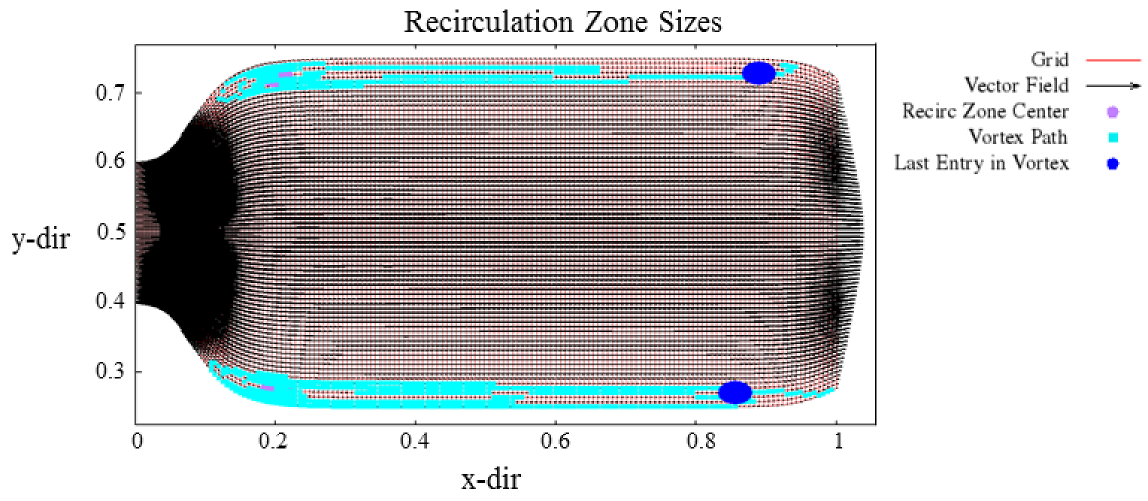


Figure 141: No jets coming in from the liner, solely flow incoming from the left at $30\frac{m}{s}$. The grid is 166 x 75 nodes, and the overall combustor has a length to height ratio of 2:1.

Now that the boundaries are known, the last part of the algorithm is to find exactly which points are to be considered within the recirculation zone, and which are not. It is important to note that this is not the only algorithm which can work. Another interesting algorithm which can be used is the method of multiple loops (MML). One could in effect use the vorticity to locate all potential vortex centers, filter through

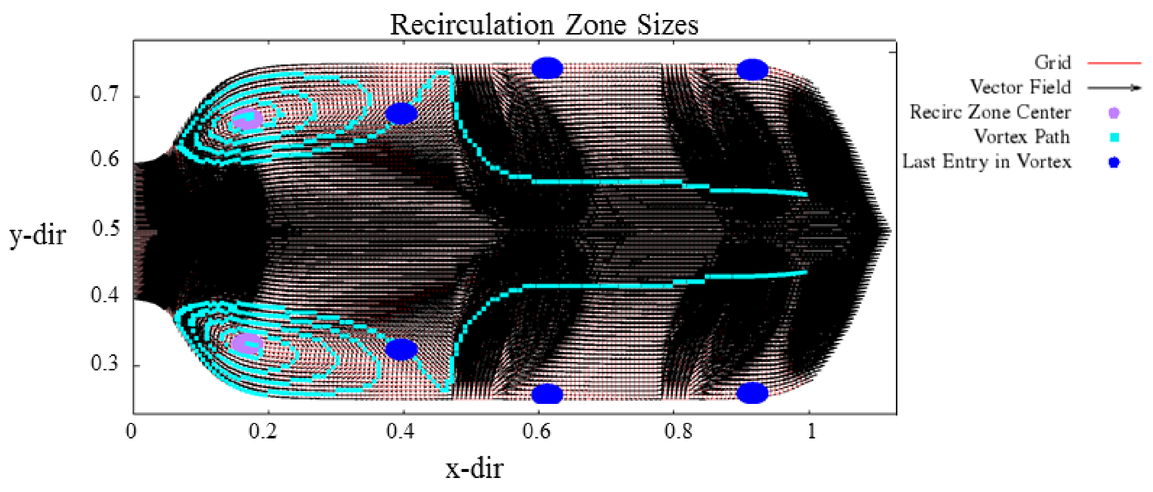


Figure 142: Three jets coming in from the liner with flow vectors $u = 12\frac{m}{s}$ and $v = 60\frac{m}{s}$ at $X = 0.05$, 0.5, and 0.8 along the top and bottom walls. The incoming flow is at $30\frac{m}{s}$. The overall combustor has a grid of 166 x 75 nodes and a length to height ratio of 2:1.

each of them by looking at the flowfield around it, and then test each zone around the vortex center individually. The algorithm for this involves starting at the vortex center and marching outwards in the i and j directions before running a higher order algorithm at each point to ensure that it forms a closed loop (as in Figure 139). Once it does, the algorithm moves on to the next point and keeps checking until a closed loop is no longer formed along all boundary cells. This is an example of when a higher order algorithm would be necessary.

To summarize this method, the code takes large time steps and (due to the nature of the flowfield within a vortex region) the locations are continuously "thrown outwards" from the center. By default, this causes expansion until the current cell is no longer within the vortex region. This allows one to check how large the vortex region is by using the numerical error to one's advantage. If this is to be useful, future steps include determining which points are going to lie within the recirculation zone, and which will not. As with many of the algorithms mentioned before, this may be easily extended to three dimensions. It may prove useful when developing an approach to create a more sparse CKN.

REFERENCES

[1] C.-M. Lee, C. T. Chang, J. T. Herbon, and S. K. Kramer, "Nasa project develops next-generation low-emissions combustor technologies," *AIAA paper*, vol. 540, p. 2013, 2013.

[2] A. H. Lefebvre and D. R. Ballal, *Gas Turbine Combustion: Alternative Fuels and Emissions, (2010).* CRC Press.

[3] E. Fleuti and J. Polymeris, "Aircraft nox-emissions within the operational lto cycle," *Unique (Flughafen Zrich AG) and Swiss Flight Data Services*, 2004.

[4] H. C. Mongia, "Combining lefebvres correlations with combustor cfd," *AIAA Technical Paper*, no. 2004-3544, 2004.

[5] P. J. Stuttaford and P. A. Rubini, "Assessment of a radiative heat transfer model for gas turbine combustor preliminary design," *Journal of propulsion and power*, vol. 14, no. 1, pp. 66–73, 1998.

[6] D. L. Allaire, I. A. Waitz, and K. E. Willcox, "A comparison of two methods for predicting emissions from aircraft gas turbine combustors," in *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pp. 899–908, American Society of Mechanical Engineers, 2007.

[7] S. R. Turns *et al.*, *An introduction to combustion*, vol. 287. McGraw-hill New York, 1996.

[8] K. Suder, J. Delaat, C. Hughes, D. Arend, and M. Celestina, "Nasa environmentally responsible aviation projects propulsion technology phase i overview and highlights of accomplishments," *AIAA Paper*, no. 2013-0414, 2013.

[9] J. D. Mattingly, *Aircraft engine design.* Aiaa, 2002.

[10] J. Seitzman, "Introduction to chemical kinetics." 2004.

[11] J. Seitzman, "Analyzing reaction mechanisms." 2004.

[12] P. Dagaut and S. Gaïl, "Kinetics of gas turbine liquid fuels combustion: Jet-a1 and bio-kerosene," in *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pp. 93–101, American Society of Mechanical Engineers, 2007.

[13] C. T. Bowman, "Control of combustion-generated nitrogen oxide emissions: technology driven by regulation," in *Symposium (International) on Combustion*, vol. 24, pp. 859–878, Elsevier, 1992.

[14] A. Lipanov and A. Bolkisev, "Solving chemical kinetics equations by explicit methods," *Journal of Applied and Industrial Mathematics*, vol. 8, no. 3, pp. 337–348, 2014.

[15] D. Forliti, "Geometrical flame holder study.".

[16] R. J. Roby, M. S. Klassen, L. D. Eskin, M. Ramotowski, and G. Gaines, "Development of a system for lean, prevaporized, premixed combustion," in *Proceedings of the Thirty-Sixth Turbomachinery Symposium, College Station, TX, Texas A&M University, Sept*, pp. 10–13, 2007.

[17] B. Khandelwal, D. Lili, and V. Sethi, "Design and study on performance of axial swirler for annular combustor by changing different design parameters," *Journal of the Energy Institute*, vol. 87, no. 4, pp. 372–382, 2014.

[18] E. Landry, S. Mikkilineni, M. Paharia, and A. McGaughey, "Droplet evaporation: A molecular dynamics investigation," *Journal of Applied Physics*, vol. 102, no. 12, p. 124301, 2007.

[19] A. Angersbach, D. Bestle, and R. Eggels, "Automated combustor preliminary design using tools of different fidelity," in *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition*, pp. V01AT04A030–V01AT04A030, American Society of Mechanical Engineers, 2013.

[20] G. Kumar, M. Rettig, H. Mongia, and C. Chauvette, "Automated cooling design methodology for combustor walls," in *36th AIAA Aerospace Sciences Meeting and Exhibit*, p. 836.

[21] N. Pegemanyfar and M. Pfitzner, "Development of a combustion chamber design methodology and automation of the design process," in *Proceedings of the 25th International Congress of the Aeronautical Sciences, ICAS*, 2006.

[22] A. Mellor and K. Fritsky, "Turbine combustor preliminary design approach," *Journal of Propulsion and Power*, vol. 6, no. 3, pp. 334–343, 1990.

[23] G. Sturgess and D. Shouse, "A hybrid model for calculating lean blow-outs in practical combustors," *AIAA paper*, no. 96-3125, pp. 1–3, 1996.

[24] P. J. Stuttaford and P. Rubini, "Preliminary gas turbine combustor design using a network approach," *Journal of engineering for gas turbines and power*, vol. 119, no. 3, pp. 546–552, 1997.

[25] A. Despierre, P. J. Stuttaford, and P. A. Rubini, "Preliminary gas turbine combustor design using a genetic algorithm," in *ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition*, pp. V002T06A007–V002T06A007, American Society of Mechanical Engineers, 1997.

[26] J. Gouws, R. Morris, and J. Visser, "Modelling of a gas turbine combustor using a network solver," *South African journal of science*, vol. 102, no. 11 & 12, pp. 533–536, 2006.

[27] A. H. Lefebvre, "Flame radiation in gas turbine combustion chambers," *International journal of heat and mass transfer*, vol. 27, no. 9, pp. 1493–1510, 1984.

[28] S. Mosier, R. Roberts, and R. Henderson, "Development and verification of an analytical model for predicting emissions from gas turbine engine combustors during low-power operation," *1973. 12*, 1973.

[29] S. A. Shakariyants, J. P. van Buijtenen, and W. P. Visser, "Aero-gasturbine emission reduction and simulation technology: Philosophy and approach," in *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, pp. 165–171, American Society of Mechanical Engineers, 2004.

[30] W. P. Visser and M. J. Broomhead, "Gsp, a generic object-oriented gas turbine simulation environment," in *ASME Turbo Expo 2000: Power for Land, Sea, and Air*, pp. V001T01A002–V001T01A002, American Society of Mechanical Engineers, 2000.

[31] R. Cant and E. Mastorakos, *An introduction to turbulent reacting flows.* World Scientific, 2008.

[32] S. M. Correa, "Turbulence-chemistry interactions in the intermediate regime of premixed combustion," *Combustion and Flame*, vol. 93, no. 1, pp. 41–60, 1993.

[33] J. Swithenbank, I. Poll, M. Vincent, and D. Wright, "Combustion design fundamentals," in *Symposium (International) on Combustion*, vol. 14, pp. 627–638, Elsevier, 1973.

[34] A. North, "Perfectly stirred reactor network modeling of nox and co emissions from a gas turbine combustor with water addition," 2012.

[35] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. J. Gardiner, *et al.*, "Gri 3.0," *Gas Research Institute, Chicago, IL, http://www. me. berkeley. edu/gri_mech*, 2000.

[36] T. Faravelli, L. Bua, A. Frassoldati, A. Antifora, L. Tognotti, and E. Ranzi, "A new procedure for predicting no x emissions from furnaces," *Computers & Chemical Engineering*, vol. 25, no. 4, pp. 613–618, 2001.

[37] G. Sturgess, J. Zelina, D. T. Shouse, and W. Roquemore, "Emissions reduction technologies for military gas turbine engines," *Journal of Propulsion and Power*, vol. 21, no. 2, pp. 193–217, 2005.

[38] R. H. Pletcher, J. C. Tannehill, and D. Anderson, *Computational fluid mechanics and heat transfer*. CRC Press, 2012.

[39] T. Poinsot and D. Veynante, *Theoretical and numerical combustion*. RT Edwards, Inc., 2005.

[40] P. Di Martino, G. Cinque, A. Terlizzi, G. Mainiero, and S. Colantuoni, “Reactive cfd analysis in a complete combustor module for aero engines application,” *Universita Degli Studi Di Napoli Federico II*, 2009.

[41] J. Kok, S. Matarazzo, and A. Pozarlik, “The bluff body stabilized premixed flame in an acoustically resonating tube: combustion cfd and measured pressure field,” 2009.

[42] W. G. Vincenti and C. H. Kruger, *Introduction to physical gas dynamics*, vol. 1. 1965.

[43] V. Fichet, M. Kanniche, P. Plion, and O. Gicquel, “A reactor network model for predicting nox emissions in gas turbines,” *Fuel*, vol. 89, no. 9, pp. 2202–2210, 2010.

[44] J. Floyd, H. Baum, and K. McGrattan, “A mixture fraction combustion model for fire simulation using cfd,” in *Proceedings of the International Conference on Engineered Fire Protection Design, Societey of Fire Protection Engineers*, pp. 279–290, 2001.

[45] A. Bourlioux and A. Majda, “An elementary model for the validation of flamelet approximations in non-premixed turbulent combustion,” *Combustion Theory and Modelling*, vol. 4, no. 2, pp. 189–210, 2000.

[46] R. Baurle, “Modeling of high speed reacting flows: established practices and future challenges,” *AIAA paper*, vol. 267, p. 2004, 2004.

[47] N. Pegemanyfar, M. Pfitzner, and M. Surace, “Automated cfd analysis within the preliminary combustor design system precodes utilizing improved cooling models,” in *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pp. 289–300, American Society of Mechanical Engineers, 2007.

[48] X. Zhang, D. J. Toal, N. W. Bressloff, A. J. Keane, F. Witham, J. Gregory, S. Stow, C. Goddard, M. Zedda, and M. Rogers, “Prometheus: A geometry-centric optimization system for combustor design,” in *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*, pp. V04AT04A061–V04AT04A061, American Society of Mechanical Engineers, 2014.

[49] D. Lee, J. Park, J. Jin, and M. Lee, “A simulation for prediction of nitrogen oxide emissions in lean premixed combustor,” *Journal of mechanical science and technology*, vol. 25, no. 7, pp. 1871–1878, 2011.

[50] M. Falcitelli, S. Pasini, N. Rossi, and L. Tognotti, "Cfd+ reactor network analysis: an integrated methodology for the modeling and optimisation of industrial systems for energy saving and pollution reduction," *Applied thermal engineering*, vol. 22, no. 8, pp. 971–979, 2002.

[51] S. A. Drennan, C.-P. Chou, A. F. Shelburn, D. W. Hodgson, C. Wang, C. V. Naik, E. Meeks, and H. Karim, "Flow field derived equivalent reactor networks for accurate chemistry simulation in gas turbine combustors," in *ASME Turbo Expo 2009: Power for Land, Sea, and Air*, pp. 647–656, American Society of Mechanical Engineers, 2009.

[52] R. Design, "Energico: Predictive analysis tools for clean-combustion design." Internet Brochure, www.reactiondesign.com, September 2015.

[53] B. Cummings, "How well preliminary sizing tools do in predicting actual nox values." Email correspondence, September 2015.

[54] A. Comsol, "Comsol multiphysics users guide," *Version: September*, 2005.

[55] A. Kumar and S. Mazumder, "Coupled solution of the species conservation equations using unstructured finite-volume method," *International Journal for Numerical Methods in Fluids*, vol. 64, no. 4, pp. 409–442, 2010.

[56] S. M. Jones, "An introduction to thermodynamic performance analysis of aircraft gas turbine engine cycles using the numerical propulsion system simulation code," 2007.

[57] D. Cline, D. Cardon, and P. K. Egbert, "Fluid flow for the rest of us: Tutorial of the marker and cell method in computer graphics," 2013.

[58] D. Cline, "Numerical simulation for computer graphics." September 2015.

[59] R. J. Kee, F. M. Rupley, E. Meeks, and J. A. Miller, "Chemkin-iii: A fortran chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics," *Sandia national laboratories report SAND96-8216*, 1996.

[60] D. G. Goodwin, H. K. Moffat, and R. L. Speth, "Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes." `http://www.cantera.org`, 2016. Version 2.2.1.

[61] J. P. Steinbrenner, J. P. Chawner, and C. L. Fouts, "The gridgen 3d multiple block grid generation system. volume 2. user's manual.," tech. rep., DTIC Document, 1991.

[62] ANSYS, *ANSYS ICEM CFD Help Manual*. ANSYS, 275 Technology Drive Canonsburg, 14.5 ed.

[63] A. Nonaka, A. Aspden, M. Zingale, A. Almgren, J. Bell, and S. Woosley, "High-resolution simulations of convection preceding ignition in type ia supernovae using adaptive mesh refinement," *The Astrophysical Journal*, vol. 745, no. 1, p. 73, 2011.

[64] M. P. Katz, M. Zingale, A. C. Calder, F. D. Swesty, A. S. Almgren, and W. Zhang, "White dwarf mergers on adaptive meshes. i. methodology and code verification," *The Astrophysical Journal*, vol. 819, no. 2, p. 94, 2016.

[65] J. F. Thompson, Z. U. Warsi, and C. W. Mastin, *Numerical grid generation: foundations and applications*, vol. 45. North-holland Amsterdam, 1985.

[66] J. D. Anderson and J. Wendt, *Computational fluid dynamics*, vol. 206. Springer, 1995.

[67] M. Zingale, "Computational hydrodynamics for astrophysics."

[68] C.-W. Shu, "High-order finite difference and finite volume weno schemes and discontinuous galerkin methods for cfd," *International Journal of Computational Fluid Dynamics*, vol. 17, no. 2, pp. 107–118, 2003.

[69] C. Wang, "Exact solutions of the steady-state navier-stokes equations," *Annual Review of Fluid Mechanics*, vol. 23, no. 1, pp. 159–177, 1991.

[70] J. Scheffel, "On analytical solution of the navier-stokes equations," tech. rep., TRITA-ALF-2001, 2001.

[71] V. D. Liseikin, *Grid generation methods.* Springer Science & Business Media, 2009.

[72] T. H. Pulliam, "Solution methods in computational fluid dynamics," *Notes for the von Kármán Institute For Fluid Dynamics Lecture Series*, 1986.

[73] Y.-L. P. Tsai and K. Hsieh, "Comparative study of computational efficiency of two lu schemes for non-equilibrium reacting flows," *AIAA paper*, no. 90-0396, 1990.

[74] K. Murawski, J. K. MURAWSKI, and P. Stpiczyński, "Implementation of muscl-hancock method into the c++ code for the euler equations," *TECHNICAL SCIENCES*, vol. 60, no. 1, 2012.

[75] M.-S. Liou, "Progress towards an improved cfd method: Ausm+," *AIAA paper*, vol. 1701, p. 155, 1995.

[76] J.-S. Shuen, M.-S. Liou, and B. Van Leer, "Inviscid flux-splitting algorithms for real gases with non-equilibrium chemistry," *Journal of Computational Physics*, vol. 90, no. 2, pp. 371–395, 1990.

[77] E. Sonnendrucker, "Numerical methods for hyperbolic systems." Online, July 2013.

[78] Z. J. Wang, L. Zhang, and Y. Liu, "Spectral (finite) volume method for conservation laws on unstructured grids iv: extension to two-dimensional systems," *Journal of Computational Physics*, vol. 194, no. 2, pp. 716–741, 2004.

[79] R. Saurel and R. Abgrall, "A simple method for compressible multifluid flows," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 1115–1145, 1999.

[80] B. Vanleer, "Flux-vector splitting for the 1990s," 1991.

[81] M.-S. Liou, "A generalized procedure for constructing an upwind based tvd scheme," 1987.

[82] R. Walters and J. Thomas, "Advances in upwind relaxation methods," in *IN: State-of-the-art surveys on computational mechanics (A90-47176 21-64). New York, American Society of Mechanical Engineers, 1989, p. 145-183.*, vol. 1, pp. 145–183, 1989.

[83] A. Baghlani, N. Talebbeydokhti, and M. Abedini, "A shock-capturing model based on flux-vector splitting method in boundary-fitted curvilinear coordinates," *Applied Mathematical Modelling*, vol. 32, no. 3, pp. 249–266, 2008.

[84] N. Hasan, S. M. Khan, and F. Shameem, "A new flux-based scheme for compressible flows," *Computers & Fluids*, vol. 119, pp. 58–86, 2015.

[85] K. Kitamura and M. Liou, "Comparative study of ausm-family schemes in compressible multiphase flow simulations," *Proceedings of ICCFD7, Big Island, Hawaii*, 2012.

[86] K. H. Kim, C. Kim, and O.-H. Rho, "Methods for the accurate computations of hypersonic flows: I. ausmpw+ scheme," *Journal of Computational Physics*, vol. 174, no. 1, pp. 38–80, 2001.

[87] W. Yao and M. Xu, "Modified ausmpw+ scheme and its application," in *System Simulation and Scientific Computing, 2008. ICSC 2008. Asia Simulation Conference-7th International Conference on*, pp. 740–743, IEEE, 2008.

[88] J. S. Park and C. Kim, "Extension of ausmpw+ scheme for two-fluid model," *Journal of the Korea Society for Industrial and Applied Mathematics*, vol. 17, no. 3, pp. 209–219, 2013.

[89] M. Lewis, J. Moss, and P. Rubini, "Cfd modelling of combustion and heat transfer in compartment fires," *Fire Safety Science*, vol. 5, pp. 463–474, 1997.

[90] W. K. Anderson and D. L. Bonhaus, "Aerodynamic design on unstructured grids for turbulent flows," 1997.

[91] J. W. Slater, "Converging-diverging verification (cdv) nozzle." Online, Jul 2008.

[92] S. Osher, "Convergence of generalized muscl schemes," *SIAM Journal on Numerical Analysis*, vol. 22, no. 5, pp. 947–961, 1985.

[93] M. Hubbard, "Multidimensional slope limiters for muscl-type finite volume schemes on unstructured grids," *Journal of Computational Physics*, vol. 155, no. 1, pp. 54–74, 1999.

[94] R. Bush, G. Power, and C. Towne, "Wind: The production flow solver of the nparc alliance," *AIAA paper*, vol. 935, p. 1998, 1998.

[95] J. Seitzman, "Coupled chemical and thermal analysis: Well-stirred reactor." Online, 2005.

[96] B. Franzelli, E. Riber, M. Sanjosé, and T. Poinsot, "A two-step chemical scheme for kerosene–air premixed flames," *Combustion and Flame*, vol. 157, no. 7, pp. 1364–1373, 2010.

[97] A. Bruyat, C. Laurent, O. Rouzaud, and G. Lavergne, "Simulation of laminar flame propagation in a multicomponent droplet stream," in *12th International Conference on Liquid Atomization and Spray Systems*, 2013.

[98] J. Luche, M. Reuillon, J.-C. Boettner, and M. Cathonnet, "Reduction of large detailed kinetic mechanisms: application to kerosene/air combustion," *Combustion science and technology*, vol. 176, no. 11, pp. 1935–1963, 2004.

[99] P. Gokulakrishnan, G. Gaines, J. Currano, M. Klassen, and R. Roby, "Experimental and kinetic modeling of kerosene-type fuels at gas turbine operating conditions," *Journal of Engineering for Gas Turbines and Power*, vol. 129, no. 3, pp. 655–663, 2007.

[100] F. Battin-Leclerc, R. Fournet, P. Glaude, B. Judenherc, V. Warth, G. Côme, and G. Scacchi, "Modeling of the gas-phase oxidation of n-decane from 550 to 1600 k," *Proceedings of the Combustion Institute*, vol. 28, no. 2, pp. 1597–1605, 2000.

[101] T. E. of Encyclopedia Britannica, "Kerosene," *Encyclopedia Britannica*, 2015.

[102] C. Developers, "Cantera: Reactors and reactor networks." 2016.

[103] R. L. Speth, "Cantera theory." September 2016.

[104] G. Lecocq, I. Hernández, D. Poitou, E. Riber, and B. Cuenot, "Soot prediction by large-eddy simulation of complex geometry combustion chambers," *Comptes Rendus Mécanique*, vol. 341, no. 1, pp. 230–237, 2013.

[105] T. Von Kármán and G. Millán Barbany, "The thermal theory of constant pressure deflagration," 1953.

[106] 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 (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[107] EASA, “Icao aircraft engine emissions databank.” Online, February 2016.

[108] N. Rizk and H. Mongia, “Semianalytical correlations for nox, co, and uhc emissions,” in *ASME 1992 International Gas Turbine and Aeroengine Congress and Exposition*, pp. V003T06A023–V003T06A023, American Society of Mechanical Engineers, 1992.

[109] A. Tsalavoutas, M. Kelaidis, N. Thoma, and K. Mathioudakis, “Correlations adaptation for optimal emissions prediction,” in *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pp. 545–555, American Society of Mechanical Engineers, 2007.

[110] D.-S. Han, G.-B. Kim, H.-S. Kim, and C.-H. Jeon, “Experimental study of nox correlation for fuel staged combustion using lab-scale gas turbine combustor at high pressure,” *Experimental Thermal and Fluid Science*, vol. 58, pp. 62–69, 2014.

[111] A. Lazzaretto, A. Toffolo, and S. Trolese, “Using experimental data for predicting performance and emissions of a real gas turbine plant,” in *ASME 2002 International Mechanical Engineering Congress and Exposition*, pp. 309–317, American Society of Mechanical Engineers, 2002.

[112] P. Moin and K. Mahesh, “Direct numerical simulation: a tool in turbulence research,” *Annual review of fluid mechanics*, vol. 30, no. 1, pp. 539–578, 1998.

[113] D. C. Wilcox *et al.*, *Turbulence modeling for CFD*, vol. 2. DCW industries La Canada, CA, 1998.

[114] N. Peters, *Turbulent combustion.* Cambridge university press, 2000.

[115] Z. Guo, H. Zhang, C. Chan, and W. Lin, “Presumed joint probability density function model for turbulent combustion,” *Fuel*, vol. 82, no. 9, pp. 1091–1101, 2003.

[116] J. Van Oijen, F. Lammers, and L. De Goey, “Modeling of complex premixed burner systems by using flamelet-generated manifolds,” *Combustion and Flame*, vol. 127, no. 3, pp. 2124–2134, 2001.

[117] P.-D. Nguyen, L. Vervisch, V. Subramanian, and P. Domingo, “Multidimensional flamelet-generated manifolds for partially premixed combustion,” *Combustion and Flame*, vol. 157, no. 1, pp. 43–61, 2010.

[118] A. Newale, “What are the pros and cons of the standard eddy breakup model as compared to ppdf equilibrium model?.” CD-adapco Steve Portal, August 2014.

[119] U. CD-adapco, “Guide: Star-ccm+,” 2009.

[120] H. Lehtiniemi, R. Rawat, K. Fröjd, and F. Mauss, "Efficient engine cfd with detailed chemistry," in *Future fuels for IC engines, ERC 2007 Symposium, Madison, WI (June 6–7, 2007) http://www. erc. wisc. edu/documents/symp07-Lehtiniemi. pdf [retrieved 25.01. 09]*, 2007.

[121] V. Nori and F. Xu, "Fundamental combustion calculations with dars and chemkin software packages: A comparison," in *ASME 2013 Gas Turbine India Conference*, pp. V001T03A004–V001T03A004, American Society of Mechanical Engineers, 2013.

[122] M. Abou-Ellail and H. Salem, "A skewed pdf combustion model for jet diffusion flames," *Journal of Heat Transfer*, vol. 112, no. 4, pp. 1002–1007, 1990.

[123] R. J. Bastiaans, "Turbulent premixed combustion with flamelet generated manifolds in comsol multiphysics®,"

[124] G. Price, K. Botros, and G. Goldin, "Cfd predictions and field measurements of nox emissions from lm1600 gas turbine during part load operation," *Journal of engineering for gas turbines and power*, vol. 124, no. 2, pp. 276–283, 2002.

[125] M. Talpallikar, C. Smith, M. Lai, and J. Holdeman, "Cfd analysis of jet mixing in low nox flametube combustors," *Journal of Engineering for Gas Turbines and Power*, vol. 114, no. 2, pp. 416–424, 1992.

[126] D. Baulch, "Evaluated kinetic data for high temperature reactions," *Evaluated kinetic data for high temperature reactions, by Baulch, DL Cleveland, CRC Press [1972-*, vol. 1, 1972.

[127] C. Baukal, "Everything you need to know about nox: Controlling and minimizing pollutant emissions is critical for meeting air quality regulations," *Metal finishing*, vol. 103, no. 11, pp. 18–24, 2005.

[128] W. May, "Reduction of thermal and prompt nox in exhausts of natural gas fueled boilers," *SFA International Website Library. www. sfainternational. com/library*, 2012.

[129] C. Schwerdt, "Modelling nox-formation in combustion processes," *MSc Theses*, 2006.

[130] C. E. Baukal Jr, *The John Zink Hamworthy Combustion Handbook: Volume 1-Fundamentals.* CRC Press, 2012.

[131] O. o. A. Q. P. Emission Standards Division and U. E. P. A. Standards, "Alternative control techniques document- nox emissions from stationary gas turbines," tech. rep., U.S. Environmental Protetion Agency, January 1993.

[132] Z. Anastassi and T. Simos, "An optimized runge–kutta method for the solution of orbital problems," *Journal of Computational and Applied Mathematics*, vol. 175, no. 1, pp. 1–9, 2005.

[133] J. M. Aristoff and A. B. Poore, “Implicit runge–kutta methods for orbit propagation,” in *AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, AIAA*, vol. 4880, p. 2012, 2012.

[134] B. Franzelli, E. Riber, L. Y. Gicquel, and T. Poinsot, “Large eddy simulation of combustion instabilities in a lean partially premixed swirled flame,” *Combustion and flame*, vol. 159, no. 2, pp. 621–637, 2012.

[135] I. Glassman, R. A. Yetter, and N. G. Glumac, *Combustion.* Academic press, 2014.

[136] M. Jiang, R. Machiraju, and D. Thompson, “Detection and visualization of,” *The Visualization Handbook*, p. 295, 2005.

[137] V. D. Liseikin, *Grid generation methods.* Springer Science & Business Media, 2009.

[138] N. Pegemanyfar, M. Pfitzner, R. Eggels, R. von der Bank, and M. Zedda, “Development of an automated preliminary combustion chamber design tool,” in *ASME Turbo Expo 2006: Power for Land, Sea, and Air*, pp. 327–336, American Society of Mechanical Engineers, 2006.

[139] N. Rizk, J. S. Chin, A. Marshall, and M. Razdan, “Predictions of nox formation under combined droplet and partially premixed reaction of diffusion flame combustors,” *Journal of engineering for gas turbines and power*, vol. 124, no. 1, pp. 31–38, 2002.

[140] I. A. Sadarjoen, F. H. Post, B. Ma, D. C. Banks, and H.-G. Pagendarm, “Selective visualization of vortices in hydrodynamic flows,” in *Visualization’98. Proceedings*, pp. 419–422, IEEE, 1998.

[141] R. W. Shonkwiler and L. Lefton, *An Introduction to Parallel and Vector Scientific Computation*, vol. 41. Cambridge University Press, 2006.

[142] G. Sturgess, S. Gogineni, D. Shouse, C. Frayne, and J. Stutrud, “Emissions and operability trades in the primary zones of gas turbine combustors,” *AIAA Paper*, vol. 2758, 1996.

[143] D. Sunday, “Inclusion of a point in a polygon.” Web, 2012.

[144] A. Van Gelder, “Vortex core detection: back to basics,” in *IS&T/SPIE Electronic Imaging*, pp. 829413–829413, International Society for Optics and Photonics, 2012.

VITA

Adam D. Siegel is a Ph.D. student at the Georgia Institute of Technology. He earned his B.S. in physics and his B.E. in Mechanical Engineering in 2011 from Stony Brook University in New York, and earned his M.S. in Aerospace Engineering from the Georgia Institute of Technology in 2012. In addition to his current research in propulsion systems design and optimization, he has also held research positions in the fields autonomous algorithms, applied numerical methods, and computational hydrodynamics. His current research at Georgia Tech focuses on computational methods for emissions prediction in gas turbine combustion systems.

A Computational Approach for Preliminary Combustor Design and Gaseous Emissions Evaluations Using a Method for Sparse Kinetics

Adam D. Siegel

260 Pages

Directed by Professor Dimitri Mavris

Despite numerous advances in the fields of numerical computing and experimentation, the development of new gas turbine combustors continues to be an extremely costly endeavor. Contemporary aircraft engine component design involves utilizing knowledge from previous architectures (in the form of trends from data, etc.), which the designer uses to develop an improved product. Burners are no different, as gas turbine combustors are conventionally designed using correlations for initial sizing and precursory emissions evaluation. With pollutants becoming an ever increasing concern in air-breathing propulsion systems, the International Civil Aviation Organization strictly regulates the amount of NOx which can be emitted from a given engine. This makes oxides of nitrogen one of the central drivers of modern combustor design, since NOx production considerations cause major restrictions on the properties of the combustor and the conditions under which it can operate.

Future engine cycles are constantly pushed to operate at conditions which tend to increase NOx emissions, and so new combustor concepts and technology must therefore be employed to achieve the lower levels called for by regulations. Utilizing correlations for these future designs however is not possible (since these combustors do not yet exist), and creating these correlations involves very expensive experimentation via computational fluid dynamics (CFD) and rig tests. To counter this cost ramification, many techniques oriented at combustor sizing and NOx prediction have been developed in the past. These approaches still fall short of what is needed, since most are either correlation based (or necessarily involve a calibration step), and all

are unable to predict NOx emissions quickly enough and accurately enough for a concept which does not yet exist.

This work aims to establish a new sizing technique which will give the combustor designer a starting point in the development of advanced concepts. By introducing a novel approach to indirectly couple chemical kinetics and conservation equations, the MoST (Method of Sparse Thermochemistry) algorithm offers a solution to this problem by utilizing sparse kinetics grids in combination with non-reacting CFD solutions. The two codes are tied together and controlled by an overarching algorithm that indirectly couples the flow field and kinetics in such a way that the flow field is still influenced by the chemical process happening within it. This eliminates the need to solve coupled nonlinear PDEs, and instead solves them as a smaller ODE (or algebraic) system. Research is presented which aids to understand the elements which must be captured to turn the flow field into a chemical kinetics network (CKN), exactly how to orient the reactors to best mock the physical processes within the flow field, and how to converge on a flow field solution while taking into account its response to chemical reactions. Grids with varying cell densities and chemical networks with varying numbers of reactors are run using this method and presented as proof of concept. Good agreement is shown between this approach and emissions data from reacting flow CFD.