Effects of Nozzle Geometry on PDRE Flow and Flexible Ceramics

a project presented to

The Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree

Master of Science in Aerospace Engineering

by

Abhi Srinath

August 2026

approved by

Dr. Periklis Papadopoulos
Faculty Advisor



© 2026
Abhi Srinath
All rights reserved.

ABSTRACT

Effects of Nozzle Geometry on PDRE Flow and Flexible Ceramics

by Abhi Srinath

Pulse detonation rocket engines (PDREs) performance is dominated by highly unsteady detonation, blowdown, and nozzle flow processes. This makes optimization and reliable performance predictions challenging. This project develops a staged ANSYS Fluent methodology for studying pressure-wave propagation in simplified PDRE relevant geometries. Pressure contours, centerline pressure histories, and x-t diagrams revealed distinct transient wave behavior among the configurations. Finally, the project will also evaluate the potential benefits of incorporating high temperature compliant flexible ceramic nozzles to mitigate the negative effects of unsteady combustion losses and thermal loading.

Acknowledgments

I would like to thank my graduate advisor Dr. Periklis Papadopoulos for his guidance and for his ongoing support on this project. I would like to thank Jonathan Contreras, my fellow student (alumnus) and colleague, for his advice. I would like to thank my family as well for their undying support.

Nomenclature

$\mathbf{u}$	fluid velocity vector
u_1	upstream velocity component normal to the shock
u_2	downstream velocity component normal to the shock
ρ_1	upstream density
ρ_2	downstream density
v	velocity
p_1	upstream pressure
p_2	upstream pressure
E	total energy
ρ_e	internal energy density
t	time
$F(t)$	instantaneous thrust at time t
$\dot{m}(t)$	mass flow rate at time t
v_e	exit velocity
p_e	exit pressure
p_a	ambient pressure
A_e	nozzle exit area
T	cycle period
τ	viscous stress tensor
u_g	mesh velocity
h_e	hess-like specific rate constant
ω	specific dissipation rate residual
k	turbulent kinetic energy
λ	divergence loss factor
a	divergence haf-angle

Contents

Nomenclature **4**

1 Introduction **7**
1.1 Problem Statement and Motivation 7
1.2 Literature Review 9
1.3 Project Proposal 14
1.4 Methodology 14

2 Computational Fluid Dynamics **16**
2.1 Modeling Approach and Assumptions 16
2.2 Governing Equations, Shock Wave Representation, and Dynamic Mesh . . . 17
2.3 Simulation Methodology and Possible ANSYS Implementation 18
2.3.1 Baseline Simulation: Static Shock Tube 18
2.3.2 PDRE Relevant Simulations 19

3 Static Shock Tube Simulation **20**
3.1 Geometry and Mesh Generation 20
3.2 Fluent Models and Set-Up 22
3.2.1 Models 22
3.2.2 Boundary Conditions and Initialization 22
3.3 Results 23
3.4 Discussion 28

4 Theoretical Predictions and Strategy Justification **29**
4.1 Motivation for a Nozzle Based Parametric Strategy 29
4.2 Governing Framework and Predications 29
4.2.1 Area Mach Number Relation 30
4.2.2 Exit Pressure and Expansion Condition 30
4.2.3 Specific Impulse 30
4.3 PDRE Relevant Simulation Cases and Justification 31
4.3.1 Baseline Reference Cone 31
4.3.2 Compact and High Divergence Configuration 32
4.3.3 High Mass Flow and Lower Expansion Ratio Configuration 32

5 PDRE Nozzle Geometries **33**
5.1 Geometry and Dimensions 33

5.1.1 Nozzle Configuration 1: 15° Half Angle . . . 33
5.1.2 Nozzle Configuration 2: 30° Half Angle High Divergence Cone . . . 35
5.1.3 Nozzle Configuration 3: Higher Mass Flow Cone . . . 36

6 Mesh Generation for PDRE Nozzle Configurations 38
6.1 Overall Meshing Philosophy . . . 38
6.2 Nozzle Configuration 1: Mesh Settings . . . 38
6.3 Nozzle Configuration 2: Mesh Settings . . . 41
6.4 Nozzle Configuration 3: Mesh Settings . . . 43
6.5 Rationale for this Meshing Approach . . . 47

7 Implementation and Simulation Settings 48
7.1 Turbulence Model Selection . . . 48
7.2 Flux Scheme, Time Integration, and Courant Number . . . 49
7.3 Boundary Conditions, Initialization, Patching, and Report Files . . . 51

8 CFD Results 53
8.1 Nozzle Configuration 1: Baseline . . . 53
8.1.1 Residuals and Centerline Pressure Report . . . 53
8.1.2 Pressure Contour . . . 54
8.2 Nozzle Configuration 2: High Divergence . . . 55
8.2.1 Residuals and Centerline Pressure Report . . . 55
8.2.2 Pressure Contour . . . 56
8.3 Nozzle Configuration 3: High Mass Flow . . . 57
8.3.1 Residuals and Centerline Pressure Report . . . 57
8.3.2 Pressure Contour . . . 58

9 x-t Diagrams and Discussion 59
9.1 x-t Diagrams and Benefit of Each Nozzle . . . 59
9.1.1 Nozzle 1 . . . 59
9.1.2 Nozzle 2 . . . 60
9.1.3 Nozzle 3 . . . 62
9.2 Discussion and Conclusion . . . 63

10 Future Work and Limitations 65

A MATLAB Code for x-t Diagram Construction 66

Chapter 1

Introduction

1.1 Problem Statement and Motivation

Pulse detonation rocket engines (PDREs) have been studied extensively due to their relatively simple geometry and potential performance advantages. PDREs represent a class of propulsion systems that utilize pressure gain combustion through repetitive detonative cycles rather than steady deflagration. Unlike conventional liquid rocket engines, where combustion occurs at approximately constant pressure, PDREs operate via intermittent detonation waves that generate significantly higher post combustion pressures and temperatures [1–3]. The theoretical thermodynamic advantage of pressure gain combustion shows potential for improved specific impulse and enhanced thrust efficiency compared to conventional constant pressure cycles [4]. Despite their conceptual simplicity (since they mainly consist of a detonation tube, injection system, ignition source, and an exhaust nozzle), PDREs present substantial modeling and optimization challenges. Each operational cycle involves the following phases:

- Propellant filling
- Ignition and detonation initiation
- Detonation wave propagation
- Blowdown and exhaust
- Purge and refill

Since the engine is inherently unsteady, performance depends primarily on transient gas dynamics, chemical kinetics, fill fraction, purge fraction, detonation frequency, tube geometry, and nozzle behavior [1,5]. Unlike steady rocket engines, thrust generation in PDREs cannot be approximated using steady chamber assumptions. Instead, time-resolved modeling of pressure history, wave propagation, and momentum flux is required. Studies have demonstrated that PDRE performance can be approximated using reduced-order, quasi-1D models that capture detonation behavior, chemistry, and blowdown dynamics. These models have reproduced experimentally measured pressure histories and thrust levels with reasonable accuracy. NASA Marshall Space Flight Center demonstrated that quasi-1D, time-domain

models can reproduce experimentally measured pressure traces and time averaged thrust to a good degree [1,6]. However, these models inherently neglect multi-dimensional wave interactions, shock reflections, boundary effects, and detailed transient flow structures that are critical in realistic PDRE operation.

As a result, computational fluid dynamics (CFD) has been proven to be a necessary tool for resolving the full transient and compressible flow physics associated with PDREs. High-fidelity CFD simulations are capable of capturing shock propagation, wave reflection, pressure decay, and boundary interactions in a manner that reduced-order models cannot. However, CFD modeling of detonation-driven systems remains computationally intensive and requires careful validation through simpler canonical problems. Some key gaps remain in the literature, particularly when considering CFD-based approaches:

- Limited integration of nozzle geometry as a primary optimization variable
- Restricted validation across multiple operating conditions
- Detonation wave propagation
- Lack of unified frameworks suitable for systematic multi-parameter optimization
- Limited exploration of adaptive or compliant nozzle concepts

Additionally, research has indicated that nozzle configurations play a significant role in determining time averaged thrust and specific impulse. Assuming a simple fixed geometry nozzle may underpredict achievable performance for pulse detonation engines. Most studies assume rigid nozzle geometries, despite the highly unsteady nature of the exhaust flow. Recent advances in ultra-high-temperature ceramic matrix composites (UHTCMCs) and flexible ceramic architectures capable of surviving extreme thermal loads raise the possibility of nozzle concepts better suited to unsteady detonation exhaust environments [7,8]. For the aerospace engineering community, this problem is significant because pressure gain combustion is a major area of research for next-generation propulsion systems, and because accurate high-fidelity modeling is required before large-scale experimental investment. Furthermore, CFD-based validation frameworks are essential for transitioning PDREs from feasibility demonstrations to engineered propulsion systems.

Therefore, the problems addressed in this project is primarily:

- the development of a validated CFD-based framework for transient relevant flow simulation for PDREs (canonical shock-tube validation and extending toward detonation-tube behavior).
- use CFD data to analyze and better PDRE performance and optimize thrust.
- evaluation of boundary effects, including nozzle geometry and compliant-wall behavior, within unsteady compressible flow environments. This problem is of direct interest to the aerospace engineering community, as high fidelity modeling tools are essential for advancing pressure gain combustion systems toward practical implementation.
- evaluate the potential benefits of incorporating high temperature compliant flexible ceramic nozzles to mitigate the negative effects of unsteady combustion losses and thermal loading.

1.2 Literature Review

PDEs and PDREs have been researched extensively on an analytical, numerical, and experimental level over the past several decades due to their potential for pressure-gain combustion and high thermodynamic efficiency. Early work showed that detonative combustion can, in principle, produce higher stagnation pressures than other systems. This motivated the exploration of pulsed detonation devices for both air-breathing and rocket propulsion applications. However, the unsteady nature of PDRE operation introduces significant modeling and design challenges, requiring time-resolved treatment of propulsion. As mentioned earlier, the conceptual foundation of the PDRE originates from the theoretical investigation of pressure gain combustion cycles. Conventional rocket engines operate under constant pressure combustion, in which entropy generation across the flame front helps limit the rise in the stagnation pressure. In contrast, detonation based combustion propagates via a coupled shock-reaction front in which chemical energy release occurs almost instantaneously behind a leading shock. The fundamental motivation for PDRE research lies in the thermodynamic differences between deflagration and detonation. In conventional rocket engines, combustion proceeds through deflagration, which is a subsonic flame process where chemical energy release occurs behind a propagating flame front. Across such a flame front, stagnation pressure generally decreases due to entropy increase. Although stagnation temperature rises significantly, the stagnation pressure ratio across the combustion process remains limited. As a result, the available expansion in the nozzle is constrained by the pressure ratio achievable in the chamber. In contrast, detonation based combustion propagates as a supersonic shock-reaction structure. A detonation wave consists of a leading shock that compresses and heats the reactants to extremely high pressures and temperatures. This is immediately followed by a thin reaction zone in which chemical energy release occurs on microsecond timescales. The combined shock and reaction structure can increase stagnation pressure across the wave. The Chapman-Jouguet (CJ) condition defines the state at which reaction products are sonic relative to the detonation front. Under CJ conditions, the flow immediately behind the detonation wave satisfies conservation of mass, momentum, and energy across the shock-reaction structure. The following relations describe these conservation constraints:

$$\rho_1 u_1 = \rho_2 u_2 \tag{1.1}$$

$$p_2 - p_1 = \rho_1 u_1 \left(u_1 - u_2\right) \tag{1.2}$$

$$h_2 - h_1 + \frac{u_2^2 - u_1^2}{2} = 0 \tag{1.3}$$

Under idealized assumptions, the CJ solution predicts a significantly higher post combustion pressure than deflagration for similar initial states. Heiser and Pratt demonstrated through ideal cycle analysis that pressure gain combustion cycles can theoretically yield higher stagnation pressure ratios and improved specific impulse relative to constant pressure cycles [2]. This thermodynamic advantage provides the principal motivation for PDRE research. However, these idealized analyses neglect unsteady losses associated with blowdown, incomplete filling, and nozzle mismatch. Therefore, while the thermodynamic basis is sound, true system

level performance depends critically on transient gas dynamics. Hinkey et al. [1] developed quasi-1D transient solvers to model detonation propagation and blowdown processes within a detonation tube. These models typically solve the unsteady compressible Euler equations coupled with simplified chemical energy release representations. In many implementations, the detonation wave is approximated using Chapman-Jouguet conditions with a moving shock boundary, while the expansion after detonation is resolved using characteristic methods or finite difference time marching. NASA Marshall Space Flight Center developed these models for PDRE performance prediction using finite rate chemistry and unsteady gas dynamics. These models resolve detonation wave propagation within the tube, followed by transient blowdown into a downstream boundary, and compute thrust by integrating the momentum flux and pressure forces over the cycle. As mentioned earlier, these results from these studies demonstrated that quasi-1D frameworks can reproduce experimentally measured pressure histories and time-averaged thrust with reasonable accuracy, while remaining computationally sound. Importantly, these models highlight that PDRE performance is strongly dependent on operating frequency, initial conditions, and tube geometry, rather than solely on ideal Chapman-Jouguet detonation properties. Hinkey et al. [1] implemented solvers that integrate the unsteady compressible Euler equations in conservative form:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ u(E+p)\end{bmatrix}=0 \tag{1.4}$$

The detonation front is tracked as a moving boundary, and post detonation expansion is computed using finite-difference time marching. Finite-rate chemistry corrections were later used to improve thermodynamic realism. Thrust is computed by integrating the instantaneous momentum flux and pressure forces at the exit:

$$F(t)=\dot{m}(t)\,v_e(t)+(p_e(t)-p_a)\,A_e \tag{1.5}$$

Time averaged thrust is then obtained over a full cycle:

$$\bar{F}=\frac{1}{T}\int_0^T F(t)\,dt \tag{1.6}$$

A key finding from NASA studies was that quasi-1D models reproduce experimental pressure histories and time averaged thrust with reasonable accuracy while remaining computationally efficient [1,6]. This demonstrated that high-fidelity CFD is not strictly necessary for real performance prediction. However, these models revealed strong dependence of performance on frequency, fill fraction, initial pressure, and tube geometry. One of the major findings from NASA's work was that reduced-order quasi-1D models can reproduce experimentally measured pressure histories and time averaged thrust with reasonable accuracy. This was a significant development because full Navier-Stokes reactive CFD simulations are computationally expensive and not well suited for optimization studies with multiple parameters. The quasi-1D framework therefore offers a tractable method for exploring design trade spaces. Additionally (briefed over earlier), as demonstrated in early thermodynamic analyses by Heiser and Pratt [2], the ideal detonation cycle theoretically can produce a higher stagnation pressure ratio than deflagration based cycles (rapid subsonic combustion) for

equivalent reactant states. This increase in stagnation pressure translates directly into greater available expansion work and, therefore, potentially higher specific impulse. Kailasanath's seminal review [3] provided one of the first comprehensive syntheses of detonation propulsion research. That work detailed detonation structure, Chapman-Jouguet conditions, wave stability considerations, and deflagration detonation transition (DDT) mechanisms. More importantly, it highlighted that detonation based engines cannot be analyzed purely through steady thermodynamic cycles. Instead, wave dynamics, transient boundary interactions, and confinement geometry play central roles in determining performance.

Pandey et al. [4] later emphasized that while pressure gain combustion is thermodynamically more lucrative, practical implementation is hindered by severe engineering challenges. Detonation initiation requires careful control of mixture preparation and ignition timing. Moreover, unsteady wave reflection at tube exits strongly affect the generation of impulse. This preliminary work established the theoretical performance potential of detonative propulsion but also made clear that transient gas dynamics must be resolved to evaluate system-level behavior. Although early research concentrated on air breathing pulse detonation engines, interest gradually expanded toward rocket-mode systems in which both oxidizer and fuel are stored onboard. PDREs remove inlet and atmospheric coupling complexities, making them attractive for upper-stage or space propulsion applications. Recent advances in ultra-high-temperature ceramic matrix composites (UHTCMCs) have expanded the limits of survivability for propulsion components [7,8]. These materials show a lot of improved thermal shock resistance and high temperature durability when compared to traditional metallic alloys. Although current PDRE studies primarily assume rigid metallic nozzles, cyclic thermal loading in PDRE may be of benefit from materials capable of tolerating rapid temperature transients. Literature on reusable rocket nozzle ceramics demonstrates successful endurance testing under extreme heating conditions [7]. However, integration of such materials into unsteady detonation environments has not been analyzed yet. Subsequent NASA investigations extended these modeling approaches to include nozzle effects and cycle based thrust evaluations. Studies that examined PDRE nozzle integration showed that assuming a simple choked or fixed area exhaust boundary can make achievable performance very unpredictable. Instead, treating the nozzle quasi-steadily at each time step (instantaneous chamber pressure and temperature define the effective expansion process) produces higher predicted thrust and specific impulse. These findings suggest that nozzle flow physics play a central role in converting pressure generated by detonation into useful impulse. It also cements the fact that nozzle design must be considered an integral part of PDRE optimization. However, NASA's investigations also revealed strong sensitivity to operating frequency. As detonation frequency increases, time averaged thrust increases proportionally up until a certain limit, but impulse per cycle may decrease if the tube does not fully refill or if blowdown is terminated prematurely. This coupling between filling time, detonation timing, and blowdown duration makes PDRE performance dependent on multiple parameters.

The PDRE operational cycle can be decomposed into five primary phases: filling, ignition, detonation propagation, blowdown, and purge. Each phase introduces unique physical processes that influence overall performance. During filling, propellant mixture must be injected uniformly to ensure reliable detonation initiation. Incomplete filling reduces available chemical energy per cycle, directly decreasing impulse. Studies have shown that fill fraction is a dominant performance parameter [11]. Ignition and DDT processes determine whether a

stable detonation wave forms. In many experimental systems, Shchelkin spirals or obstacle arrays are used to promote DDT, but these introduce additional pressure losses and structural mass penalties. Detonation propagation generates an abrupt pressure rise typically on the order of tens of atmospheres above initial chamber conditions. The CJ detonation speed depends on mixture composition and initial thermodynamic state. Accurate representation of this phase requires resolving the shock front and immediate post-detonation thermodynamic properties. The blowdown phase dominates thrust generation. After the detonation wave exits the tube, high-pressure combustion products expand through the nozzle. Unlike steady rocket chambers, pressure decays rapidly over milliseconds. This transient expansion means that the nozzle operates under time-varying chamber conditions. Finally, purge removes residual combustion products to prevent misfire in the subsequent cycle. Excessive purge, however, reduces effective fill fraction and degrades performance. Literature consistently emphasizes that these phases are strongly coupled. Modeling any one phase independently is insufficient for accurate performance prediction. Experimental validation has been central to PDRE research. Schauer et al. [9] performed high-frequency detonation tube experiments using hydrogen-oxygen mixtures, recording pressure histories with microsecond resolution. These experiments demonstrated repeatable detonation propagation and consistent impulse measurements across cycles. There has been some work that expands on multi-cycle and higher-frequency operation [10–12]. These studies confirmed some important trends:

- Time averaged thrust scales approximately linearly with frequency at moderate ranges.
- Impulse per cycle decreases when fill fraction falls below optimal values.
- Tube's length:diameter ratio affects detonation development and blowdown duration.
- Overly aggressive purge fractions significantly reduce net impulse.

These datasets are critical because they produce pressure vs time curves. Reduced-order models can therefore be validated against transient waveforms rather than only steady metrics. However, most experimental studies are conducted at specific geometric configurations and limited operating conditions. Few studies explore wide sweeps where comprehensive validation of predictive models are limited. The modeling strategies used in the literature primarily fall into three categories:

- Full reactive Navier-Stokes CFD
- Quasi-1D unsteady models
- Thermodynamic cycle approximations

High-fidelity CFD resolves detonation structure, chemical kinetics, viscous effects, and shock interactions in detail. Since it's been established that thermodynamic cycle approximations treat PDREs as idealized pressure gain devices but ignore unsteady blowdown and nozzle interactions, it is known that these models often overpredict performance. Experimental investigations have provided validation data for these modeling efforts. NASA and collaborating institutions have conducted PDE and PDRE experiments with high-frequency

pressure transducers and thrust measurement systems, reporting chamber pressure, impulse, and thrust across a range of operating frequencies and propellant combinations. This data demonstrates characteristic pressure decay following detonation, strong repeatability at stable operating conditions, and trends between frequency and time averaged thrust. Such measurements form a critical benchmark for validating reduced-order simulation frameworks and will serve as primary comparison targets for the present project.

Beyond single condition validation, multiple studies have explored parametric trends in PDRE operation. Investigations into fill fraction and purge fraction have shown that incomplete filling or excessive purge can significantly degrade performance by reducing effective detonation energy release per cycle. Tube length-to-diameter ratio has also been shown to influence detonation development and blowdown efficiency, with longer tubes generally increasing impulse-per-cycle but imposing penalties in mass and packaging. These findings indicate that PDRE design inherently involves multi-parameter trade spaces rather than single-variable optimization. Recent research has begun to formalize this trade space using optimization techniques. Multi-objective nozzle optimization studies for rocket-type PDEs have demonstrated that nozzle throat area and expansion ratio can be tuned to maximize thrust while minimizing losses associated with over or under expansion across the blowdown process. These studies support the notion that systematic optimization is necessary to approach PDRE performance limits. Quasi-1D unsteady solvers offer a balance between fidelity and efficiency. NASA's modeling framework [1,6] represents the detonation as a propagating discontinuity and resolves post-detonation expansion using compressible flow equations. While these models neglect multi-dimensional effects, they capture the dominant longitudinal dynamics that determine thrust.

Despite their advantages, many quasi-1D models treat nozzle expansion simplistically, assuming choked flow at the exit or constant area expansion. Only a limited number of studies integrate time-varying nozzle expansion within the transient solution. Nozzle design in PDRE systems is more complex than in steady rocket engines. Because chamber pressure decays rapidly during blowdown, the nozzle may experience alternating over-expansion and under-expansion within a single cycle. Kim et al. [13] conducted one of the few systematic studies examining nozzle geometry optimization for rocket-type PDEs. Their results demonstrated that throat area and expansion ratio significantly affect predicted thrust when incorporated dynamically rather than assumed fixed. Despite the substantial PDRE literature, several gaps remain relevant to the present work. First, many existing models focus on demonstrating feasibility rather than providing a unified, validated framework suitable for broad optimization studies. Second, validation is often limited to a small number of operating points, leaving uncertainty regarding model robustness across frequency, geometry, and fill conditions. Third, while nozzle effects are recognized as important, relatively few studies integrate nozzle geometry as a primary optimization variable within a time-domain PDRE solver. Nevertheless, most literature treats nozzle geometry as secondary to detonation modeling. Few studies perform multi-objective optimization incorporating both operational parameters and geometric variables. Furthermore, validation of optimized configurations against experimental data remains limited. From the cumulative literature, several consistent gaps emerge:

- CFD analysis for PDREs have proven to be expensive and intensive over and over

again.

- Many reduced-order models demonstrate feasibility but are not structured for systematic optimization.
- Validation across wide operational envelopes remains limited.
- Nozzle geometry is rarely treated as a primary optimization variable.
- Performance modeling and materials feasibility are rarely considered within the same framework.

1.3 Project Proposal

The objective of this project is to develop, validate, and apply a staged CFD framework in ANSYS Fluent capable of capturing transient compressible flow behavior relevant to pulse detonation rocket engines. The framework will begin with a static shock-tube simulation and then move onto PDRE relevant nozzle simulations to establish numerical accuracy and physical consistency.

The project further aims to evaluate the influence of boundary conditions, including nozzle geometry and wall compliance, on transient pressure evolution, shock propagation, and overall flow behavior. This objective is specific, measurable, and achievable within two semesters. Validation targets will be defined in terms of agreement with analytical shock-tube solutions, x–t diagram structure, and comparison with published experimental or numerical pressure traces [21, 23]. Outcomes will be presented through pressure-time histories, contour plots, and comparative analyses.

1.4 Methodology

The methodology will follow a staged simulation and validation process centered around CFD implementation in ANSYS Fluent, supported by analytical and reduced-order modeling where appropriate. The governing equations will consist of the compressible Navier–Stokes equations, which describe conservation of mass, momentum, and energy in transient, compressible flow regimes.

The first stage of the methodology involves simulating a classical shock tube to validate solver setup, mesh resolution, and time-stepping strategy. The simulation will reproduce known shock-tube behavior, including incident shock propagation, contact surface evolution, and reflected shock formation. Results will be analyzed through x–t diagrams constructed from pressure or density data along the tube centerline. Following validation of the static shock tube, the methodology will extend to nozzle geometries. In this next stage, the CFD framework will be applied to PDRE-relevant configurations by replicating published detonation-tube or pulse detonation engine studies. Geometry, boundary conditions, and initial conditions will be defined based on literature sources, allowing comparison of pressure histories and transient flow structures. At each time step, pressure fields will be resolved throughout the computational domain, allowing direct observation of shock dynamics, expansion behavior,

and boundary interactions. Validation will be performed by comparing CFD results with analytical shock-tube solutions, x–t diagram structures, and published experimental or numerical datasets. Where appropriate, reduced-order quasi-1D models may be used to provide baseline comparisons for pressure evolution and performance trends. These models will assist in interpreting CFD results and identifying deviations due to multi-dimensional effects.

Finally, the project evaluates the influence of nozzle geometry on transient flow behavior. Results will be analyzed in terms of pressure evolution, wave interaction, and potential implications for PDRE performance and structural feasibility.

Chapter 2

Computational Fluid Dynamics

2.1 Modeling Approach and Assumptions

PDRE modeling requires a framework capable of resolving highly transient, compressible flow phenomena. It is characterized by strong shock waves, rapid pressure gradients, and unsteady expansion processes through different stages. Fundamentally, PDRE operation presents a tradeoff between real world feasibility and predicted computational efficiency. As mentioned in earlier sections, while reduced-order models provide valuable insights, they do not capture features of multi-dimensional flow, wave reflections, or boundary interactions. These are super critical to the nature of detonation based flows. For this reason, this project adopts a computational fluid dynamics based approach using ANSYS Fluent. The flow within the detonation tube is assumed to be 1D, unsteady, and compressible. These assumptions are justified by the dominance of axial wave propagation in PDRE operation. The working fluid will be modeled as an ideal gas with constant specific heat ratio γ. Thermodynamic properties are assumed to be uniform across all cross-sections of the tube. The detonation process will be modeled using the CJ approximation (the detonation wave is treated as an instantaneous discontinuity). Viscous effects, wall heat transfer, and boundary layer growth are neglected in the base model. The nozzle flow is treated as quasi-steady. At each time step, the flow properties observed at the nozzle are assumed to adjust instantaneously to the current chamber conditions. The modeling strategy is structured progressively. The initial focus is on smaller scale compressible flow problems, such as shock tubes, to establish accuracy and understanding.

Numerical modeling of turbulent flows can be approached through several paths, namely Reynolds-Averaged Navier-Stokes (RANS), Large Eddy Simulation (LES), and Direct Numerical Simulation (DNS). Among these, the RANS approach is widely regarded the best in engineering practice, however, it requires addressing what is commonly referred to as the closure problem [14]. Two broad strategies exist for tackling this: the Boussinesq hypothesis and Reynolds Stress Model (RSM) transport equation modeling. The Boussinesq hypothesis establishes a relationship between the Reynolds stresses and the mean velocity gradients of the flow through the concept of eddy viscosity. Since the turbulent stresses are treated similar to laminar viscous stresses and the turbulence itself is assumed to be isotropic, an eddy viscosity coefficient is introduced to close the system [14][15]. Turbulence models are often classified

by the number of additional transport equations they introduce. Algebraic models such as Prandtl's Mixing-Length Model require no transport equations. While this is straightforward in formulation, it is limited in the range of applications [16]. The Spalart-Allmaras model represents a one-equation approach, solving a single transport equation for a modified eddy viscosity. Greater versatility is found in two-equation models, of which the Standard k-ε and k-ω models are the most prominent examples. The k-ε model solves for turbulent kinetic energy and the turbulence dissipation rate ε (the rate at which kinetic energy is converted to thermal energy through viscous action). This model performs well in fully turbulent, wall-distant flows [17]. However, it is not free of faults. Its known shortcomings include poor handling of strong flow separation, pronounced streamline curvature, and low Reynolds number near wall-like regions. The k-ω model replaces the dissipation rate with the specific dissipation rate ω, offering improved near-wall performance and removing the need for wall functions. However, it introduces sensitivity to freestream boundary conditions [18].

To better use the complementing strengths of both mentioned approaches, the Shear Stress Transport (SST) model blends k-ω behavior in such wall-like regions with k-ε behavior in the freestream. By doing so, this produces a robust hybrid that has seen widespread adoption [19]. At the higher end of RANS complexity, the Reynolds Stress Model (RSM) abandons the eddy viscosity concept entirely. Instead, solving individual transport equations for each component of the Reynolds Stress Tensor alongside a dissipation equation. This equation totals seven equations, hence its occasional designation as a seven-equation model [20]. While RSM is better equipped to capture turbulence anisotropy and complex stress interactions, its considerable computational overhead and numerical stiffness make two equation models the more practical choice for most RANS based analyses. The models discussed thus far are largely rooted in incompressible flow assumptions. However, when flow velocities become a significant fraction of the speed of sound, compressibility effects can no longer be ignored. In compressible flows, density variations become a primary driver of flow physics. It couples the momentum equations directly to the energy equation and introduces phenomena such as shock waves, expansion fans, and strong pressure-dilatation interactions. These effects fundamentally alter the behavior of turbulence and present additional modeling challenges that extend beyond the standard incompressible RANS framework. This motivates the need for compressibility corrections and carefully selected turbulence closures suited to high-speed flow environments.

2.2 Governing Equations, Shock Wave Representation, and Dynamic Mesh

The flow field is governed by the compressible Navier-Stokes equations. The conservation of mass, the momentum conservation (Cauchy momentum), energy conservation, and ideal gas equations are shown below.

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

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2.2}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot [\mathbf{u}(\rho E + p)] = \nabla \cdot (k \nabla T) + \phi \tag{2.3}$$

$$p = \rho R T \tag{2.4}$$

Shock waves are central to both the shock tube and PDRE physics. These discontinuities involve abrupt changes in pressure, density, and temperature over very small scales of spaces. Numerically, they must be resolved using schemes that capture shocks and avoid spurious oscillations. In this project, a density-based solver is used, as it is better suited for high mach compressible flows. Temporal resolution will be governed by the Courant-Friedrichs-Lewy (CFL) condition, ensuring stability while capturing transient wave motion. The accuracy of shock representation is strongly dependent on mesh resolution and timestep selection. These parameters will be systematically refined to make sure that physically meaningful results are produced. The continuity equation now becomes:

$$\frac{d}{dt} \int_V \rho \, dV + \int_S \rho(\mathbf{u} - \mathbf{u}_g) \cdot \mathbf{n} \, dS = 0 \tag{2.5}$$

This formulation allows the computational domain to deform in response to prescribed boundary motion which will be governed by the equation. The plan is to use this capability to emulate compliant-wall effects relevant to flexible ceramic concepts that can be extrapolated to PDRE systems. The governing equations and numerical framework described above form the foundation for modeling the relevant flows for this project. Shock propagation, pressure decay, and wave reflections observed in shock tubes are directly relevant to processes occurring in detonation tubes. The CFD framework can be extended toward realistic PDRE simulations.

2.3 Simulation Methodology and Possible ANSYS Implementation

The simulation workflow will be implemented using ANSYS Fluent. Geometry creation will be performed using DesignModeler or SpaceClaim, followed by mesh generation in ANSYS Meshing. The simulations will be executed in Fluent using transient compressible solvers, and results are post-processed using Fluent and CFD-Post. The methodology follows a staged progression, ensuring that each level of complexity is validated before moving on to the next.

2.3.1 Baseline Simulation: Static Shock Tube

The initial simulation focuses on a classical shock tube configuration. The domain consists of a high-pressure driver section and a low-pressure driven section, separated initially by a diaphragm. The diaphragm rupture is modeled by initializing different pressure regions within the same computational domain. The simulation is performed using a transient density-based solver with ideal gas assumptions. Walls are treated as adiabatic, and the flow is initially considered laminar. A structured mesh is used to ensure accurate resolution of wave propagation along the tube.

A key output of the shock tube simulation is the x-t diagram, which tracks the position of flow features as a function of time. This is constructed by extracting pressure data along the centerline of the tube. The x-t diagram will reveal the trajectories of the incident shock, contact surface, expansion fan, and reflected shock. This will serve as a primary validation metric for the CFD model.

2.3.2 PDRE Relevant Simulations

The final stage of the methodology will involve replicating PDRE-relevant configurations based on published literature [23]. These simulations will aim at reproducing pressure data and flow structures observed in detonation tubes. By matching geometry, boundary conditions, and initial states, the CFD model can be validated against experimental or numerical benchmarks, establishing confidence in its predictive capability. Once again, mesh quality plays a critical role in capturing shock dynamics. A structured mesh will be used, with refinement in regions where strong gradients are expected. Solver settings include transient time integration, compressible flow formulation, and appropriate boundary conditions. Sensitivity studies will be conducted to ensure mesh independence and timestep convergence.

Chapter 3

Static Shock Tube Simulation

3.1 Geometry and Mesh Generation

Based on the source, the geometry that needed to be constructed in Ansys is as shown below. Start off in SpaceClaim by 'Ending Sketch'. After doing so, click on 'Sketch Mode'. This allows selection of the plane on which to construct the 2D structure. After selecting the XY plane, sketch a simple rectangle anywhere on the plane, with the base along the x-axis, through the origin. After doing so, click on 'Dimension'. This allows making of specific dimensions for the rectangle. Select a length of 1000mm and a height of 30mm. Once these dimensions are set, dimension the distance between the left bottom edge of the rectangle (which lies on the x-axis) and the origin to be 500mm. By doing this, the bottom of the rectangle is now bisected by the origin. Next, the 'Split' tool was used to cut the rectangle in half at the origin. This results in the geometry shown. This is the completed geometry. Update the geometry on ANSYS, and then close SpaceClaim.

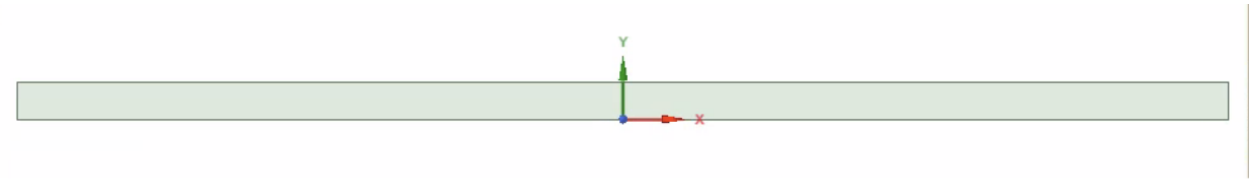


Figure 3.1: Shock Tube geometry constructed in SpaceClaim (1000mm x 30mm)

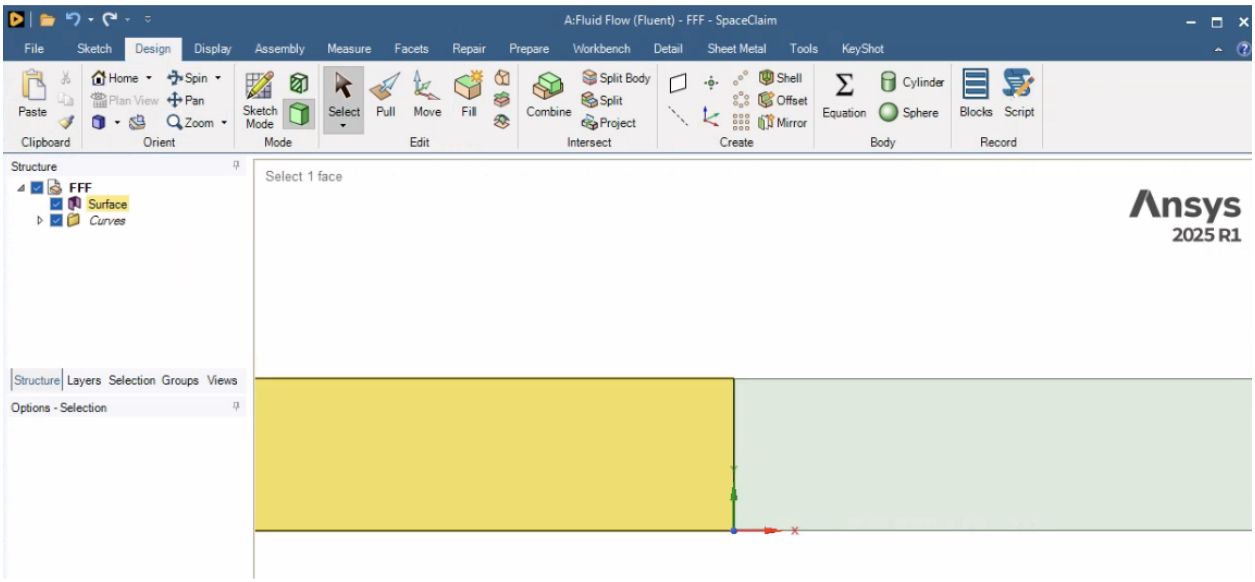


Figure 3.2: Close-up of divided shock tube in SpaceClaim

Once the geometry appears on the Meshing desktop, ensure that the split of the rectangle was successful and both the halves are of the same color. Next, insert a 'Face Meshing' and select both halves of the structure using the Face Select tool and hit apply. Next, insert a 'Sizing'. This time, using the Edge Select tool, select all the horizontal lines of the structure (2 on top, 2 in the bottom) and click apply. Change the 'Type' to 'Number of Divisions' and enter 500 divisions (1 for each mm of length). Next, insert another Edge Sizing, and this time select all the vertical lines (3 of them) and click apply. Once again, change the 'Type' to 'Number of Divisions' and enter 30 divisions (1 for each mm of height). After this, click on 'Generate Mesh'. The result is shown below.

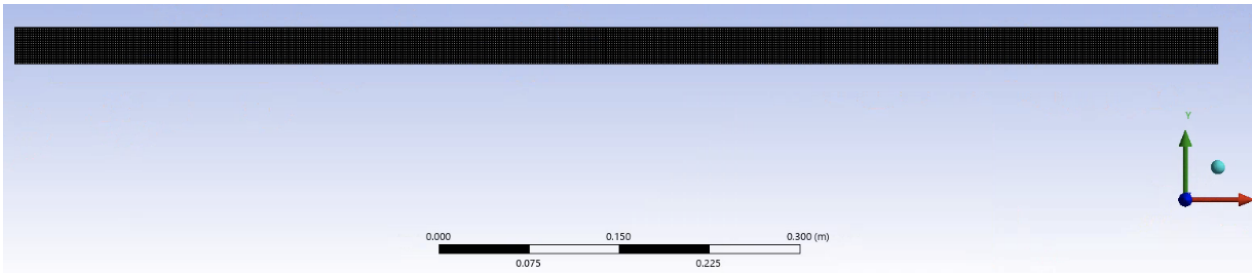


Figure 3.3: Shock Tube Mesh

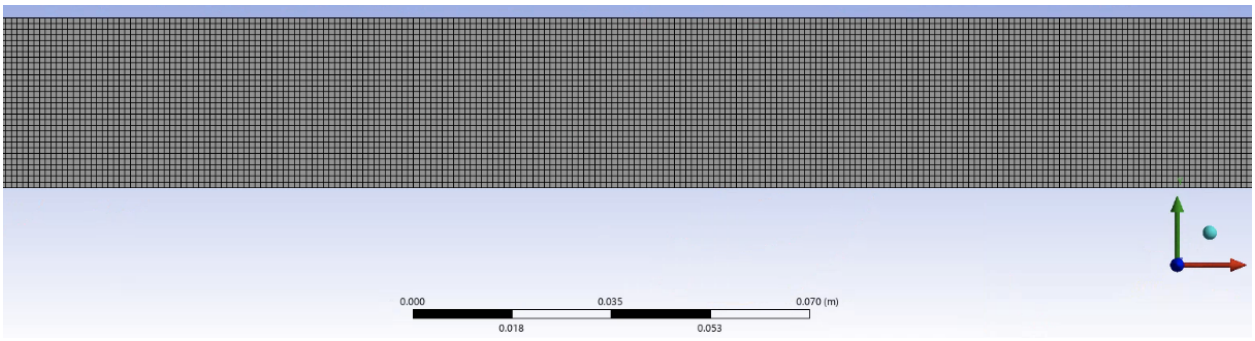


Figure 3.4: Close-up of Mesh

Finally, to allow setting boundary conditions in the future, define a few 'Named Selection'. First, select the left half of the structure using Face Select and name this as 'Driver'. Do the same with the right half and name it 'Driven'. Next, using the Edge Select tool, select all the boundary lines (do not select the middle cut) and name them 'Wall'. Finally, select the middle line that cuts the rectangle and name it 'Diaphragm'. Update the mesh on ANSYS. This mesh can be refined further for the turbulent flow models to obtain more accurate results.

3.2 Fluent Models and Set-Up

3.2.1 Models

Inviscid Flow: The simulation is a density based sim with transient time. In the 'Models' selection, choose to turn on 'Energy' equations. And for the viscosity, for this flow, select 'Inviscid' flow. Since the structure is enclosed, without any inlet or outlet, under the 'Physics' tab, change the pressure of the operating conditions to be 0 Pa.

Laminar Flow: In the 'Models' selection, choose to turn on 'Energy' equations. And for the viscosity, for this flow, select 'Laminar' flow, as opposed to inviscid in the previous model. Again, since the structure is enclosed, without any inlet or outlet, under the 'Physics' tab, change the pressure of the operating conditions to be 0 Pa.

Turbulent Flow - Spalart Allmaras (1 equation): After setting a density based sim with transient time once again, in the 'Models' selection, choose to turn on 'Energy' equations. And for the viscosity, for this flow, select 'Spalart Allmaras' flow. Under the 'Physics' tab, change the pressure of the operating conditions to be 0 Pa.

Turbulent Flow - k–ω (2 equations): Keeping the density based sim with transient time once again, in the 'Models' selection, choose to turn on 'Energy' equations. And for the viscosity, for this flow, select k–ω flow. Under the 'Physics' tab, change the pressure of the operating conditions to be 0 Pa again.

3.2.2 Boundary Conditions and Initialization

As for the 'Boundary Conditions', for this baseline sim, an enclosed tube is considered. However, check each zone in the dropdown to see if they are described correctly. Diaphragm

described as 'interior' type, interior-driven described as 'interior', interior-driver described as 'interior', wall-driven described as 'wall', and wall-driver described as 'wall'.

In the part of the initialization, the driver and driven pressures need to be set. This can be achieved by using the 'Patch...' option. However, initially, this option is blocked out. So, click on 'Initialize' without changing any of the options. This initializes the problem. Once this is done, select 'Patch...' and set Pressure as the desired variable. After that, selecting driven as the zone, set a value of 10000 Pa and click patch. Similarly, after deselecting driven and selecting driver, set a value of 100000 Pa and click patch. Next, add a pressure contour and orient it to the bottom. This will help with depicting the pressure changes in the shock tube (The contour shown below). Once this initialization set-up is complete, the simulation can be run.

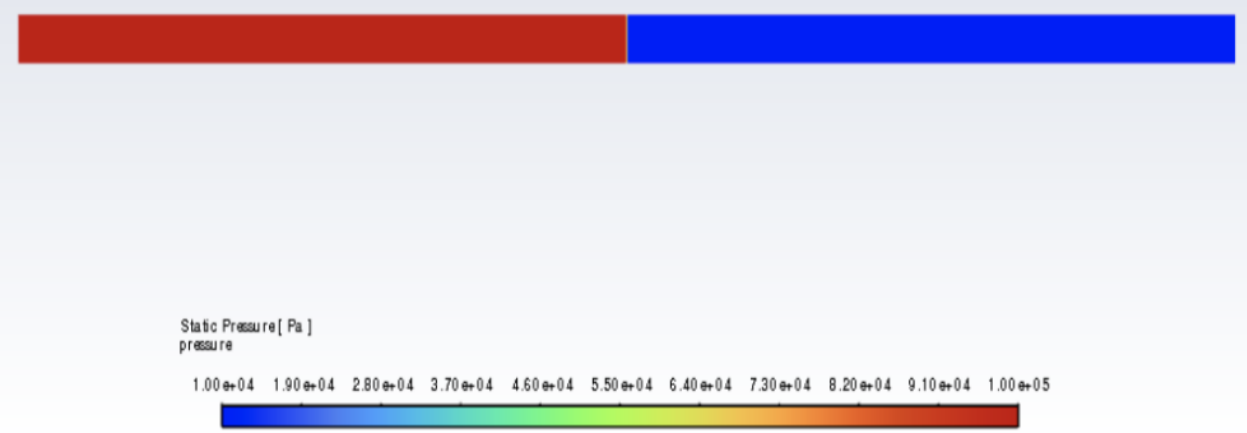


Figure 3.5: Static Shock Tube Contour

3.3 Results

Before looking at this result, a quick recap on the simulation set-up helps. On the left side of the shock tube (Driver) is air at a higher pressure than the Driven. The two are separated by a diaphragm. The x-t diagram monitors the changes in pressure as a function of time and position, as soon as this diaphragm breaks or opens - creating an incident shockwave (at time 0ms). This mimics the PDRE base set up.

This incident shockwave slowly begins to travel through the tube. There is expansion on the driver side and compression on the driven side (hereby increasing pressure in the driven and lowering pressure in the driver). This is depicted by the left side of the x-t diagram where an expansion wave has formed which is travelling to the left from position 0m to -0.5m (through the Driver) and the wave covers more ground as the time increases. Simultaneously, the shock wave is moving from the position 0m through the driven section, at the forefront of which a contact surface is formed, which increases the pressure of the driven section as it travels from 0m to 0.5m, covering more ground with time. After this, the shock is reflected from the driven wall, leaving a trail of higher pressure and this wave continues to propagate (as seen from time 1.75ms and beyond).

Inviscid Flow:

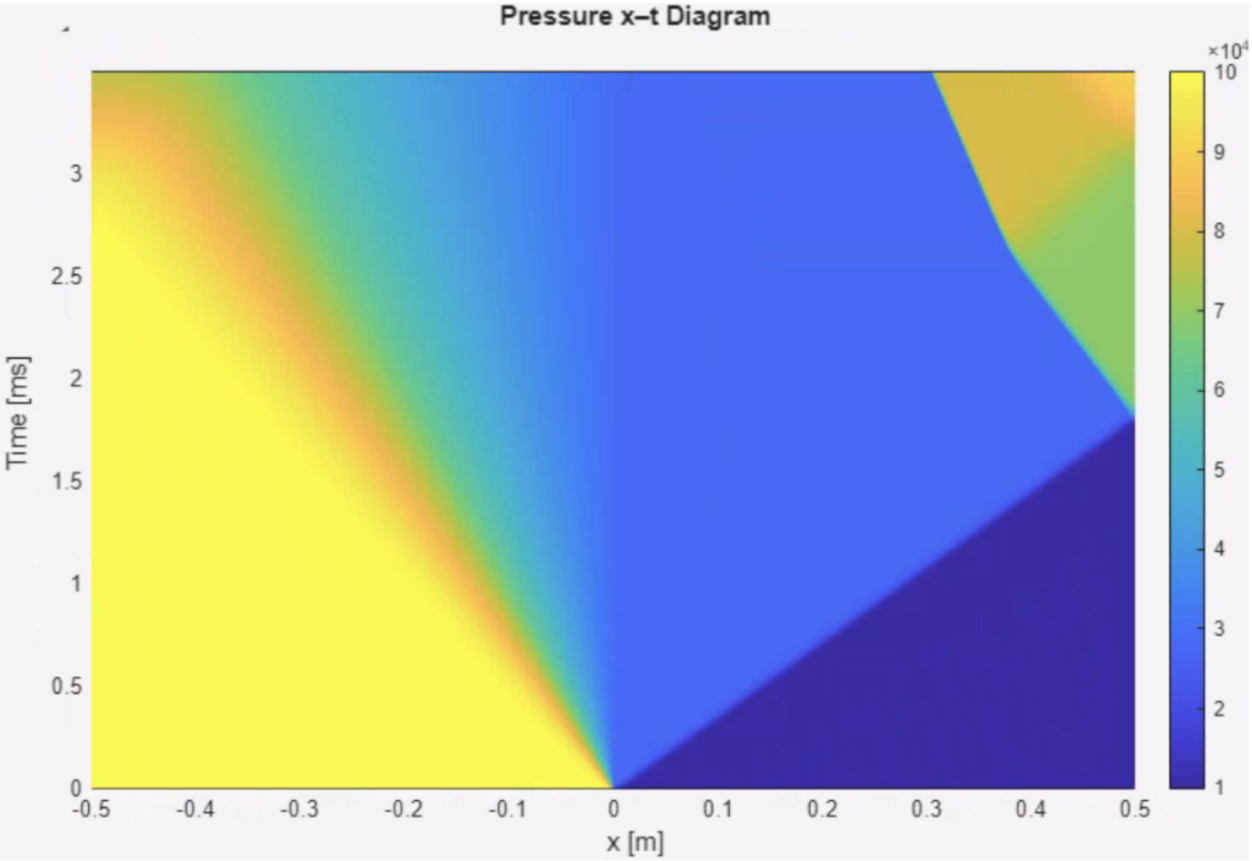


Figure 3.6: x-t diagram for inviscid flow with air

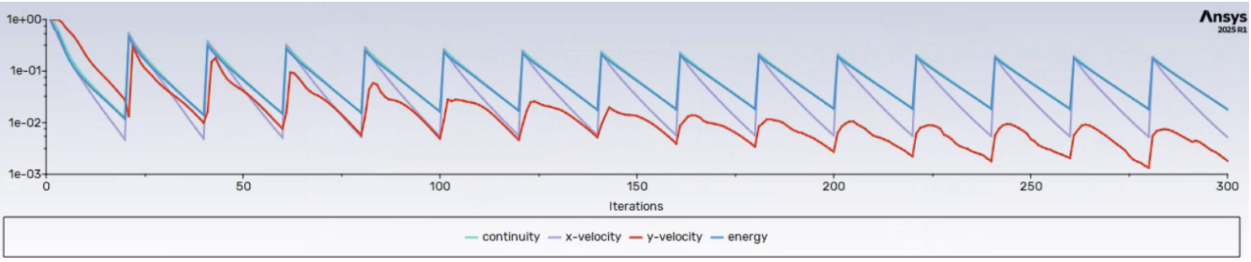


Figure 3.7: Residuals for inviscid flow with air

Laminar Flow:

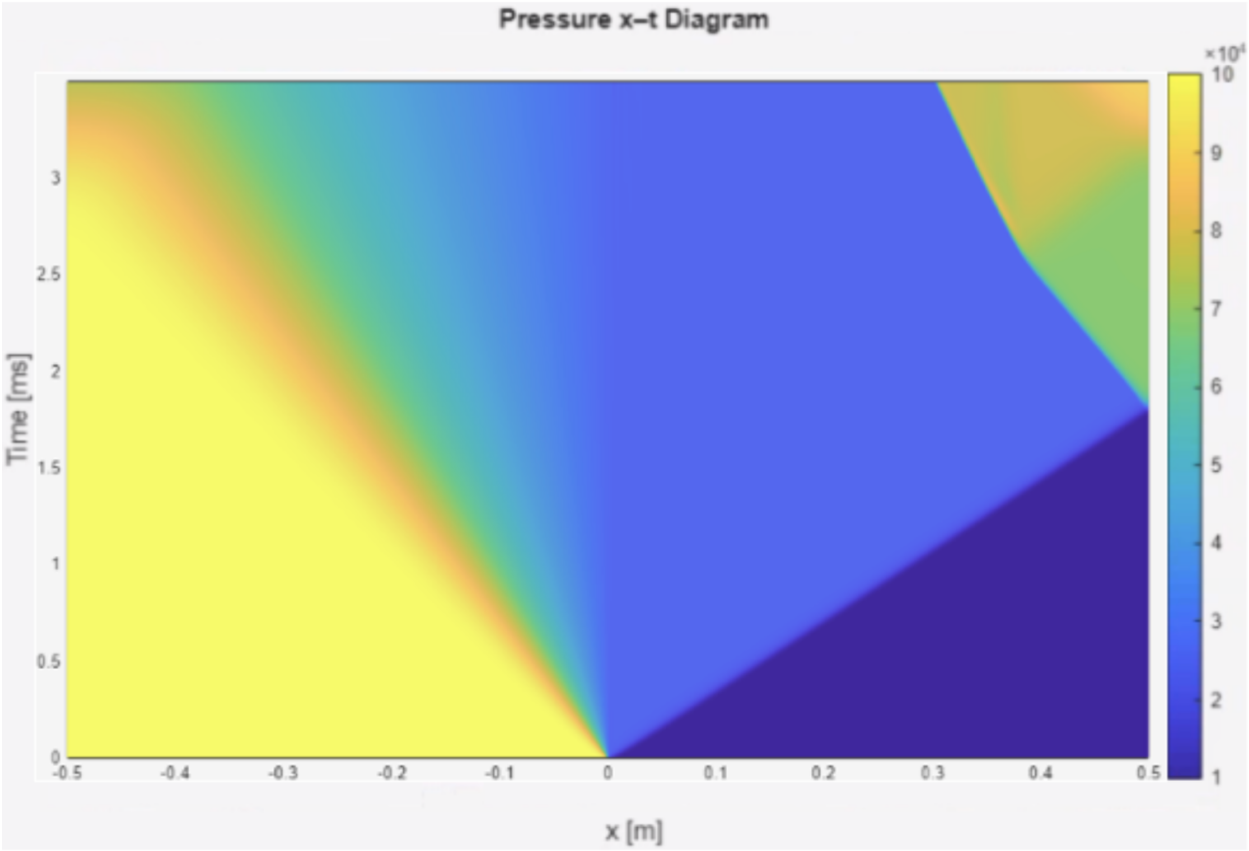


Figure 3.8: x-t diagram for laminar flow with air

Turbulent Flow: Spalart Allmaras (SA)

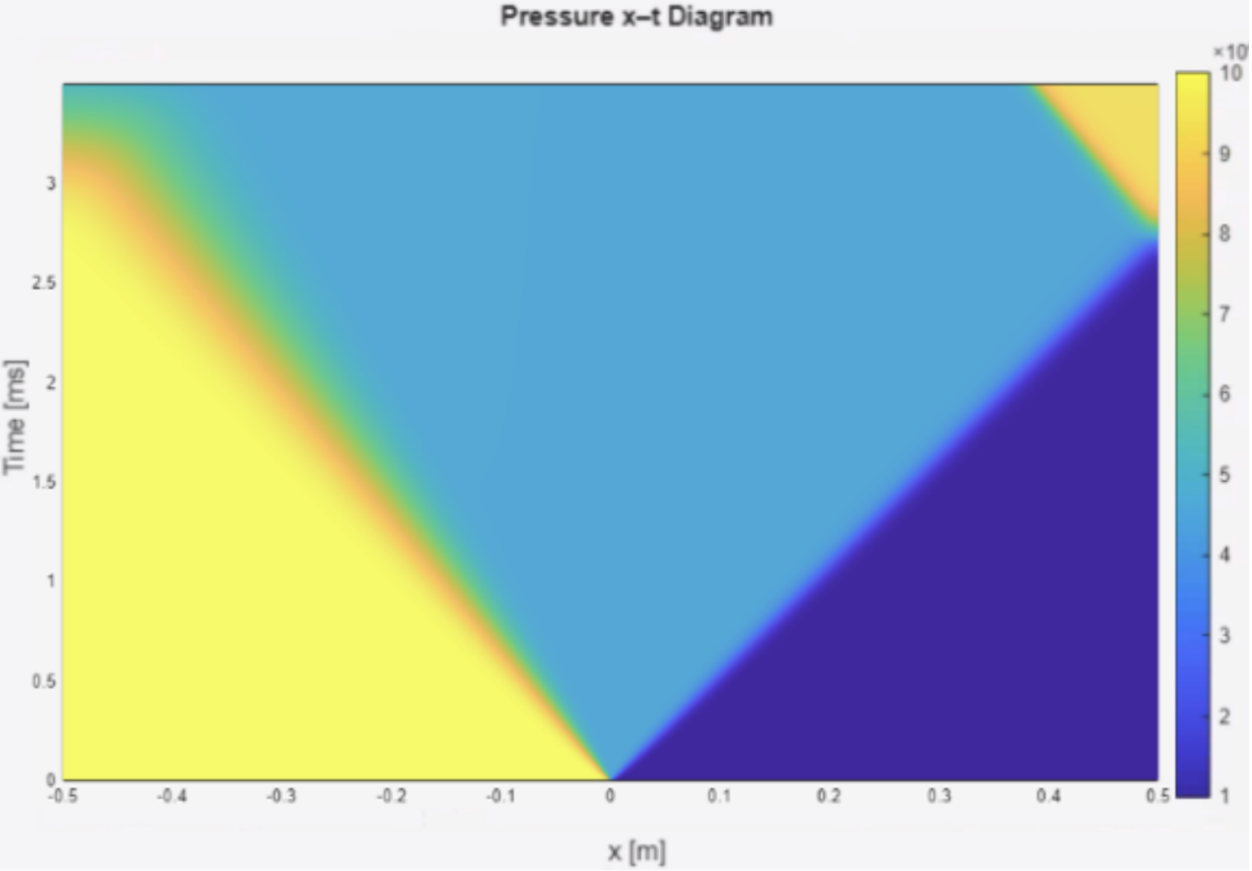


Figure 3.9: x-t diagram for turbulent flow SA with air

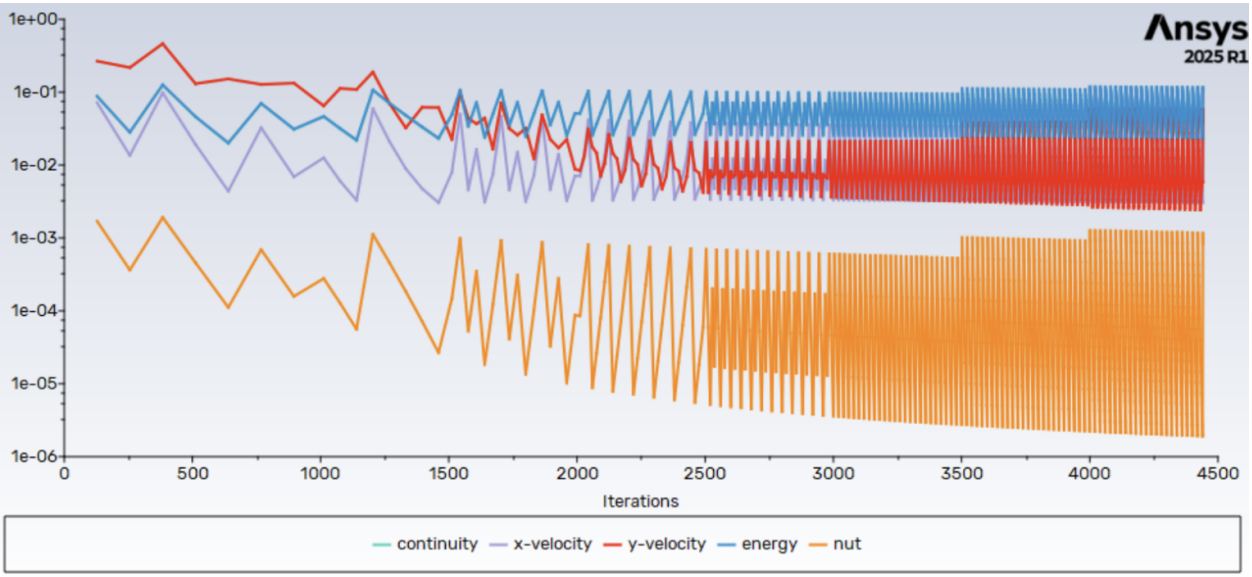


Figure 3.10: Residuals for turbulent flow SA with air

Turbulent Flow - k–ω

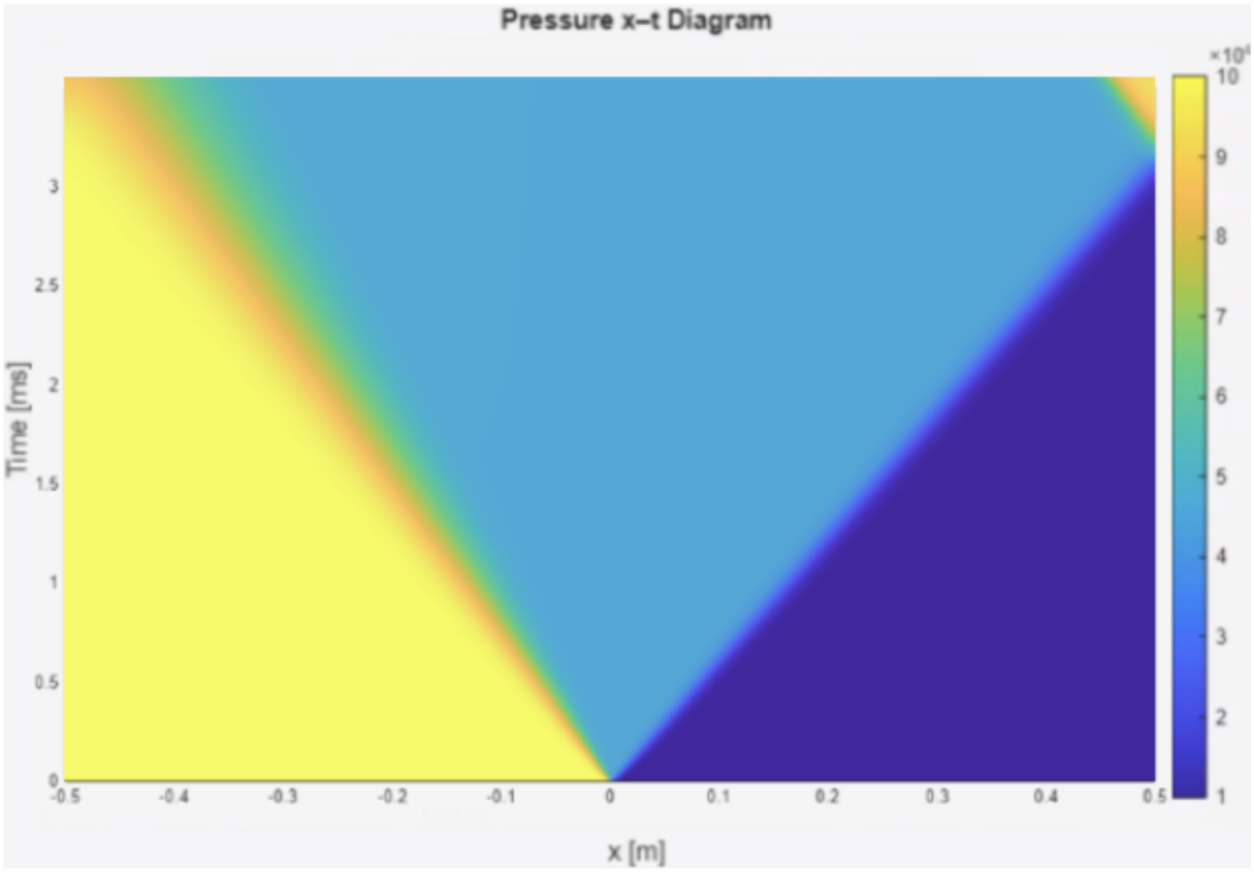


Figure 3.11: x-t diagram for turbulent flow k–ω with air

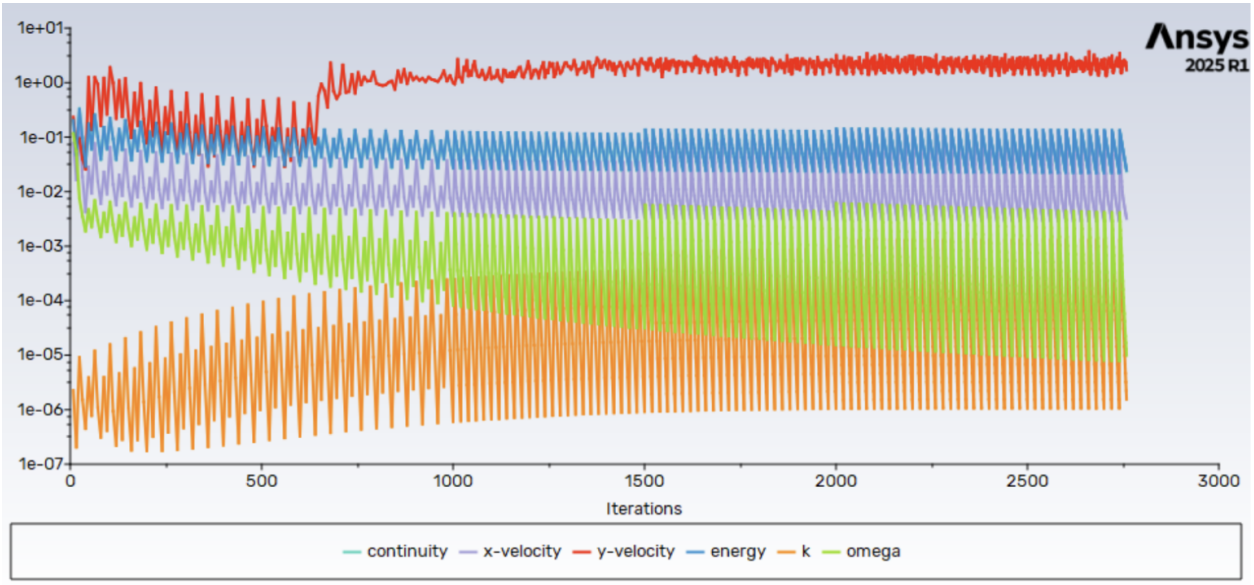


Figure 3.12: Residuals for turbulent flow k–ω with air

3.4 Discussion

The x–t diagrams for inviscid and laminar flow (Figures 3.6 and 3.8) are nearly indistinguishable. This is consistent with 1D compressible flow theory, which is governed by the inviscid Euler equations. This agreement confirms that our numerical solutions are accurate and that the similarity of the two x-t diagrams is meaningful.

Turbulent flow theory has certain recognizable features that are used here to discern if the simulated flow is in agreement with theory. Turbulence increases mixing of pressure - hereby allows for stronger diffusion of the pressure gradients. This in turn increases diffusion at the contact surface as well, giving the x-t diagram a less sharpened look. By observing Figures 3.9 and 3.11, one can see that the contact surface (the slope-like line from 0m to 0.5m) is more diffused and softer (compared to Figures 3.6 and 3.8 which appear sharper). With turbulent flows, the shock front is thicker, causing the shock line to be rounded, particularly at the point of reflection. In Figures 3.9 and 3.11, at around 2.75 ms, the x-t diagrams are curved and the reflection lasts for a longer duration. Compare it to Figures 3.6 and 3.8, the reflections are sharper and occur over a shorter period. This is also in agreement with the theoretical conclusion of a thicker shock front for turbulent flows. The most prominent feature from theoretical predictions is that turbulent flow reduces the effective shock propagation speed. This is supported by our simulations as well. Figures 3.9 and 3.11 show a delayed propagation speed. While Figures 3.6 and 3.8 show shockwave reflection around 1.7 ms, Figures 3.9 and 3.11 show a shockwave reflection at around 2.75 ms and 3 ms respectively.

Now compare Figures 3.9 and 3.11 to one another. According to turbulent flow theory, the main difference between the two turbulent models chosen is that the k-omega flow model produces higher Eddy viscosity when compared to the SA model. This higher viscosity is observable since it delays the propagation of the shock further. This can be observed in the x-t diagrams. The graphs appear to be visually similar, however, Figure 3.11 shows a delay in reaching the other end before being reflected back (around 3ms), when compared to Figure 3.9 (which has a reflection around 2.75 ms). The findings through simulation support the theoretical predictions.

Chapter 4

Theoretical Predictions and Strategy Justification

4.1 Motivation for a Nozzle Based Parametric Strategy

Nozzle geometry is consistently treated as a secondary consideration in PDRE modeling. There is strong evidence that it governs how efficiently the pressure gain achieved during detonation is converted into thrust and impulse. Kim et al. [13] showed that throat area and expansion ratio significantly alter predicted thrust. The broader literature review done in Chapter 1 showed "limited integration of nozzle geometry as a primary optimization variable" as one of the central, unresolved gaps in current PDRE research. This study strategy is designed to address that gap directly, using the same validated solver setup, meshing approach, and boundary condition strategy already demonstrated for the static shock tube in Chapter 3, rather than introducing new numerical complexity through mesh motion.

Three cone configurations have been selected, each isolating a distinct geometric parameter relevant to real nozzle design trade-offs: half-angle, expansion (area) ratio, and throat diameter. Each case is justified in terms of the physical question it is meant to answer, followed by a dedicated subsection of theoretical predictions describing the flow behavior expected from each configuration based on classical nozzle theory, quasi-1D compressible flow relations, and the empirical trends reported in the literature review (Section 1.2).

4.2 Governing Framework and Predications

Before describing the individual cases, it is useful to establish the theoretical tools that will be used to generate predictions and later interpret the CFD results. Because the nozzle flow in this project is treated as quasi-steady (Section 2.1) at each instant of the blowdown process, classical isentropic quasi-1D nozzle relations remain directly applicable, despite the driving chamber pressure itself being transient.

4.2.1 Area Mach Number Relation

For isentropic, quasi-1D compressible flow through a converging diverging nozzle passage, the ratio of local area A to throat area A^* related to the Mach M of flow by:

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{\frac{\gamma+1}{\gamma-1}}$$

This relation shows that for a fixed throat area, a larger exit:throat area ratio corresponds to a higher exit Mach number, provided the flow remains supersonic and attached through the divergent section. This relation will be the primary tool used to predict exit Mach number trends across the three geometries.

4.2.2 Exit Pressure and Expansion Condition

The relationship between stagnation pressure p_0 , exit pressure p_e , and exit Mach number M_e is given by:

$$\frac{p_0}{p_e} = \left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{\gamma}{\gamma-1}}$$

Chamber pressure decays rapidly during PDRE blowdown. Hence, a nozzle designed with a fixed geometric area ratio will pass through a range of expansion conditions over a single cycle. These being early in blowdown, when chamber pressure is high, the flow may be underexpanded or ideally expanded. Later in the cycle, as chamber pressure falls, the same geometry can become overexpanded. This transient mismatch is a central motivation for testing multiple nozzle geometries rather than assuming a single "perfect" expansion ratio, since no single fixed geometry can remain ideally expanded throughout the entire blowdown.

4.2.3 Specific Impulse

Specific impulse is considered the primary performance metric tying these effects together. The equation that can be used to govern this is:

$$I_{sp} = \frac{F}{\dot{m}\, g_0}$$

Since $\dot{m}(t)$ is set upstream by the detonation and blowdown process (largely independent of nozzle shape, to first order, for a fixed throat area), differences in I_{sp} across the three cases are expected to closely track differences in thrust, and therefore in exit velocity and expansion behavior.

4.3 PDRE Relevant Simulation Cases and Justification

Three cone geometries have been selected to isolate three distinct, independently controllable design variables, while holding the driver (detonation tube) geometry and initial patched pressures identical to the values validated in Chapter 3 (100,000 Pa driver / 10,000 Pa driven) so that only the nozzle geometry itself changes between cases. This design mirrors a classic single-variable sweep: each case changes exactly one geometric parameter relative to the Chapter 4 baseline cone (15° half-angle, 17.86 mm throat radius, 68.33 mm slant length), so that any differences observed in the x–t diagrams, pressure histories, and derived thrust/impulse metrics can be attributed to that one parameter rather than a confounded combination of several.

4.3.1 Baseline Reference Cone

A 15° half-angle cone with a 17.86 mm throat radius and 68.33 mm slant length. A 15° half-angle is a widely used as a conventional compromise value in real conical rocket nozzle design. This is because it is large enough to keep the nozzle short and lightweight, while being small enough that the divergence loss factor, called λ, remains close to 1. This means very minimal axial thrust is sacrificed to non-axial exit flow.

The case is expected to produce a relatively clean shock and expansion structure in the x–t diagram, similar in qualitative shape to the shock tube turbulent SA results already obtained in Chapter 3 (Figure 3.9), but modified by the contact surface and reflected shock should still be visible. However, wave speeds and reflection timing will differ from the constant area shock tube because the changing area introduces additional compression (in the convergent section) and expansion (in the divergent section) effects not present in the shock tube case. This case serves as the theoretical and numerical anchor point for evaluating whether Cases 2 and 3 represent an improvement or a degradation in performance.

4.3.2 Compact and High Divergence Configuration

The second case substantially increases the half angle (almost double) while holding the throat area fixed and shortening the slant length so that the same overall exit radius is achieved over a shorter axial distance. This case is designed to isolate the effect of divergence angle. This configuration is motivated by a real-world engineering trade-off relevant to PDRE applications: because PDREs are pulsed, cyclic devices subject to repeated thermal and pressure loading, a shorter, more compact nozzle carries real practical advantages. However, classical nozzle theory predicts that this compactness is not free. The larger divergence angle should reduce the effective axial thrust captured by the λ factor above, and is also expected to enhance flow separation risk in the divergent section under overexpanded conditions.

Since the area ratio is matched to Case 1, the ideal (attached flow, isentropic) exit Mach number and exit pressure should be nearly identical to Case 1 according to the area Mach relation in Section 4.2.1. The key expected difference is therefore not in the core 1D flow quantities, but in a measurably reduced axial thrust component, and (2) an increased likelihood of boundary-layer separation or a more turbulent, less axially aligned exit flow structure. In the x–t diagram, the shock and expansion structure is expected to remain broadly similar in timing to Case 1 (since the throat area and driver conditions are unchanged), but is expected to show a less clean, more diffused reflected shock structure near the divergent section.

4.3.3 High Mass Flow and Lower Expansion Ratio Configuration

The third case holds the 15° half angle from Case 1 fixed but increases the throat radius, thereby increasing the overall throat area while testing a different area ratio than Case 1. This isolates the effect of expansion ratio and throat sizing, independent of divergence angle. This case is motivated by a equally important design question raised throughout the literature review - for a given detonation tube diameter, is it more advantageous to expand the exhaust gas aggressively or to size the throat larger to pass more mass flow with a comparatively lower expansion ratio?

This mirrors the classic rocket engine design trade off between maximizing velocity and maximizing mass flow. By the area Mach relation, a larger expansion ratio is expected to correspond to a higher exit Mach number and correspondingly lower exit pressure for the same driver conditions. Since the driver/driven initial pressures are being held fixed at the Chapter 3 values (100,000 Pa / 10,000 Pa) across all three cases, a larger expansion ratio nozzle is at greater risk of becoming overexpanded relative to the (0 Pa gauge) ambient condition used in the shock tube setup. The mass flow rate through the throat, however, should scale with throat area for choked flow. This configuration is expected to accommodate the same detonation-driven pressure differential but distribute it through a wider throat before expanding it over a larger area ratio.

Chapter 5

PDRE Nozzle Geometries

5.1 Geometry and Dimensions

5.1.1 Nozzle Configuration 1: 15° Half Angle

As mentioned earlier, the baseline nozzle configuration serves as the reference geometry. Nozzle configurations 2 and 3 are evaluated against this baseline geometry, isolating divergence angle and expansion-ratio effects respectively. The geometry consists of the shock tube in Chapter 3: a constant-area tube followed by a converging diverging nozzle section.

Table 5.1: Baseline Nozzle Dimensions

Part	**Dimension**
Driver/driven tube length	65 mm
Tube height (radius)	15.18 mm
Convergent section axial length	10 mm (top and bottom, symmetric)
Divergent section slant length	68.33 mm
Nozzle exit height (radius)	45.37 mm
Nozzle half-angle	15°

The 65 mm constant-area section upstream of the nozzle represents the detonation tube. At the origin, the tube walls converge symmetrically over a short 10 mm axial run before immediately transitioning into the 15° half angle divergent section. It is worth noting that the geometry does not include a discrete constant-area throat segment: the convergent and divergent walls meet directly at a single vertex on the centerline. This is a deliberate simplification. Instead of modeling a filet-radius, the throat is treated as a sharp geometric transition point. This keeps the geometry description reducible to a small number of driving variables and avoids introducing an additional parameter (fillet radius) that is not the subject of this project.

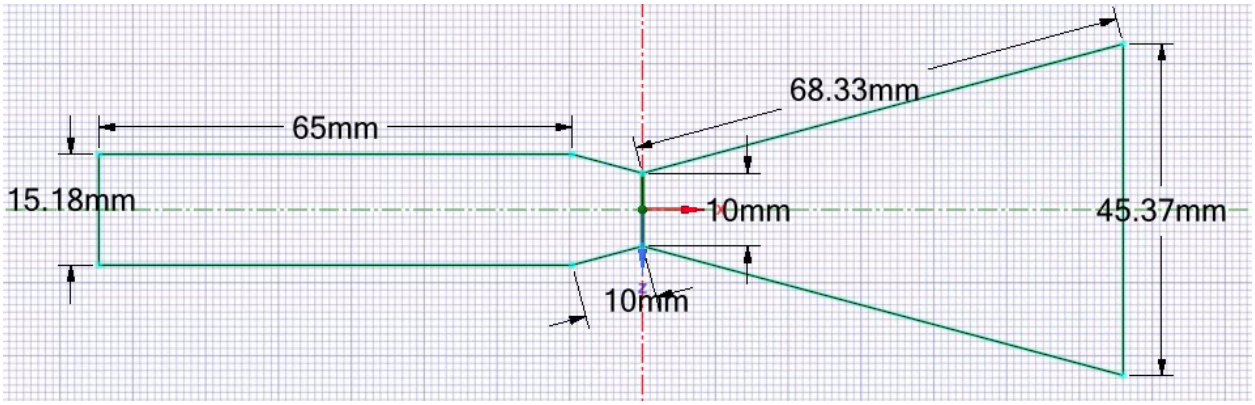


Figure 5.1: Nozzle 1 Geometry and Dimensions

SpaceClaim Procedure and Dimension Rationale

- Open the geometry in SpaceClaim and click Sketch Mode, then select the XY plane as the sketch plane.
- Sketch the constant-area section first: draw two horizontal lines (top and bottom tube walls) originating from a point well to the left of the origin, running along the x-axis direction. Use the Dimension tool to set the full length of this straight section to 65 mm and the height to 15.18 mm.
- From the downstream end of the section, sketch two converging lines that angle inward toward the centerline. Dimension the axial length of this convergent run to 10 mm (top and bottom).
- Continue the sketch by drawing two diverging lines outward from the point where the convergent lines meet the centerline. Dimension the half-angle of these lines to 15°, and dimension the slant length to 68.33 mm. The resulting exit height should measure 45.37 mm.
- Once the full profile is closed and dimensioned, use the Split tool to bisect the entire body through the origin, perpendicular to the centerline (the same operation used in Chapter 3 to separate Driver and Driven regions of the shock tube). This produces two halves: the driver/upstream half (containing the constant-area tube and convergent section) and the driven/nozzle half (containing the divergent section and exit).
- Update the geometry and close SpaceClaim.

The choice to go with a milimeter scale structure rather than a larger geometry was driven by several practical and methodological considerations specific to the project. A smaller physical domain allows a fine mesh to be achieved with a manageable total cell count. At larger scales, it would multiply the cell count substantially, extending solve times well beyond what is feasible for this project.

The solver used throughout this project is governed by the Courant-Friedrichs-Lewy condition (Section 2.2). Smaller cells require correspondingly smaller time steps to maintain stability.

5.1.2 Nozzle Configuration 2: 30° Half Angle High Divergence Cone

Configuration 2 isolates the effect of divergence angle while holding the driver geometry, throat region, and exit radius identical to configuration 1.

Table 5.2: High Divergence Nozzle Dimensions

Part	Dimension
Driver/driven tube length	65 mm
Tube height (radius)	15.18 mm
Convergent section axial length	10 mm (top and bottom, symmetric)
Divergent section slant length	35.37 mm
Nozzle exit height (radius)	45.37 mm
Nozzle half-angle	30°

The driver tube length (65 mm), the half-height of the tube (15.18 mm), the length of the convergent section (10 mm) and the exit half-height (45.37 mm) are unchanged from baseline. The only dimension that was altered is the divergent section's slant length, reduced from 68.33 mm in the baseline to 35.37 mm here. Since both configurations must reach the same exit radius, cutting the slant length in half necessarily increases the divergence half-angle by a corresponding amount. The exact resulting half-angle should be read directly from the SpaceClaim dimension callout for this sketch.

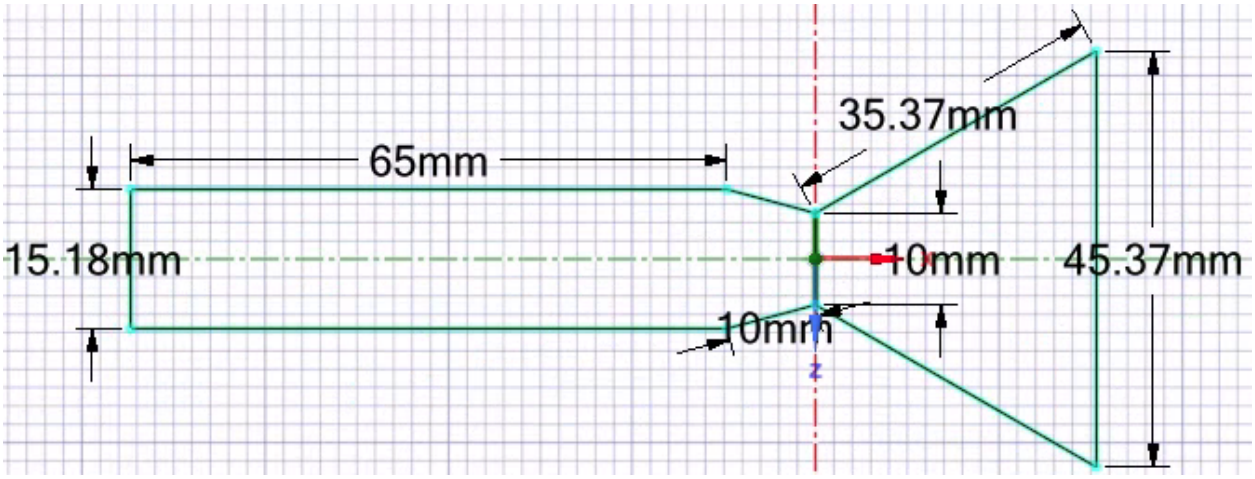


Figure 5.2: Nozzle 2 Geometry and Dimensions

SpaceClaim Procedure and Rationale

- Open a new SpaceClaim sketch and confirm the XY sketch plane is selected.
- Sketch the constant-area driver/driven section exactly as in Configuration 1: 65 mm total length, 15.18 mm height.

- Sketch the convergent section identically to the baseline. 10 mm axial length, tapering the wall inward toward the centerline.
- From the point where the convergent lines meet the centerline, sketch the divergent section outward, but this time dimension the slant length to 35.37 mm. Dimension the exit height as 45.37 mm and allow the half-angle to be the resulting quantity. This ensures the two configurations are directly comparable at the exit plane.
- Use the Split tool to bisect the body through the origin, producing the Driver and Driven face regions.
- Update the geometry in ANSYS and close SpaceClaim.

Since PDREs are pulsed and cyclic devices, they are subject to repeated pressure loading on every cycle. A shorter and compact nozzle carries genuine practical advantages such as reduced mass, reduced exposure area, and simpler packaging. Configuration 2 tests whether that compactness can be achieved simply by increasing the divergence angle while holding the throat and exit areas fixed. Classical nozzle theory predicts that this compactness is not as easy solution. A larger divergence angle reduces the divergence loss factor λ, meaning a measurable fraction of the exit velocity is directed non-axially and does not contribute to useful axial thrust. A steeper divergence angle is also expected to increase the risk of flow separation in the divergent section under overexpanded conditions. Configuration 2 was therefore selected specifically because it isolates this trade-off of compactness versus axial thrust efficiency.

5.1.3 Nozzle Configuration 3: Higher Mass Flow Cone

Configuration 3 isolates the effect of throat sizing and expansion ratio, holding the 15° half-angle from the baseline fixed while increasing the throat area. The question this section will try and address is if it's more advantageous to expand the exhaust gas aggressively (as shown in configuration 2) or to size the throat larger and pass more mass flow through a comparatively lower expansion ratio.

Table 5.3: High Flow Rate Nozzle Dimensions

Part	**Dimension**
Driver/driven tube length	65 mm
Tube height (radius)	21.04 mm
Convergent section axial length	20 mm (top and bottom, symmetric)
Throat half-angle	2 mm
Divergent section slant length	55 mm
Nozzle exit height (radius)	48.47 mm
Nozzle half-angle	15°

Every dimension in this configuration is larger than the corresponding baseline value: the driver tube half-height increases from 15.18 mm to 21.04 mm, the convergent section is

lengthened from 10 mm to 20 mm, and the exit height increases modestly from 45.37 mm to 48.47 mm.

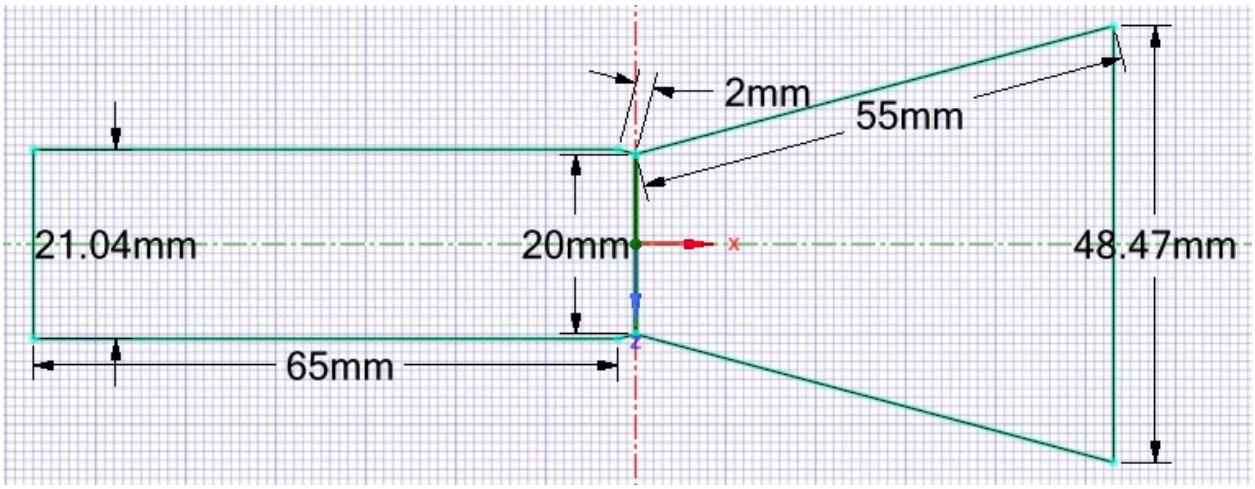


Figure 5.3: Nozzle 3 Geometry and Dimensions

SpaceClaim Procedure and Rationale

- Open a new sketch on the XY plane in SpaceClaim.
- Sketch the constant-area driver section of 65 mm total length, with a height of 21.04 mm.
- Sketch the convergent section tapering inward from the 21.04 mm tube wall over a 2 mm slant length .
- From the throat, sketch the divergent section outward at a 15° half-angle, dimensioning the slant length to 55 mm. Confirm the resulting exit height matches the labeled 48.47 mm.
- Use the Split tool to bisect the body through the origin, producing the Driver and Driven face regions.
- Update the geometry in ANSYS and close SpaceClaim.

Configuration 3 was chosen to directly test the trade between maximizing exit velocity and maximizing mass flow, applied to the PDRE context. Rather than assuming aggressive expansion (as in Configuration 2) is automatically the better strategy, this configuration asks whether, for a fixed detonation tube diameter, it is more advantageous to pass a larger mass flow through a bigger throat at a comparatively lower expansion ratio. By the Area-Mach relation (Section 4.2.1), a lower expansion ratio is expected to correspond to a lower exit Mach number and correspondingly higher exit pressure than the baseline for the same conditions.

Chapter 6

Mesh Generation for PDRE Nozzle Configurations

6.1 Overall Meshing Philosophy

Every nozzle configuration is meshed using the same general strategy. First, a structured, quadrilateral face mesh across the full domain. Then, a graded bulk element size controlling resolution in the interior region, and a finer edge sizing applied specifically to the wall boundaries. This section helps outline the reasoning behind that general strategy.

The underlying physics motivating this approach is the same across all three configurations which is that the flow field is dominated by a shock front, contact surface, and reflected waves that traverse the entire axial length of the nozzle over the course of each transient run. The quantities of shock strength, pressure decay, and velocity gradients near the walls are most sensitive to resolution close to the solid boundaries. This is because this is where viscous and turbulent effects are concentrated. This motivates the mesh refinement toward the walls rather than distributing a uniformly fine mesh across the whole domain, which would be wasteful.

Since every configuration in this study is a simple symmetric (along the axis) structure with no complex internal features, a mapped face mesh can be applied across the entire domain rather than resorting to triangulation, in every case. This keeps orthogonal quality high and skewness low across all the three geometries.

6.2 Nozzle Configuration 1: Mesh Settings

A Face Meshing was applied to both the driver and driven faces, consistent with the meshing approach validated for the shock tube in Chapter 3. A Face Sizing of 0.0001 m (0.1 mm) was then applied to both parts, with a growth rate of 1.2. This establishes the target element size across the bulk of the domain and controls how quickly cell size increases moving away from any locally refined regions (such as the edges).

A growth rate of 1.2 means each successive row of cells is permitted to grow by at most 20 percent relative to the previous row. This conservative value was chosen to avoid abrupt

Table 6.1: Nozzle 1 Mesh Settings

Parameter	**Value**
Face Meshing	both Driver and Driven
Face Sizing	0.0001 m
Face Sizing Growth Rate	1.2
Edge Sizing	0.00005 m
Edge Sizing Growth Rate	1.1

jumps in cell size, which can introduce numerical diffusion or skewness at between fine and coarse regions.

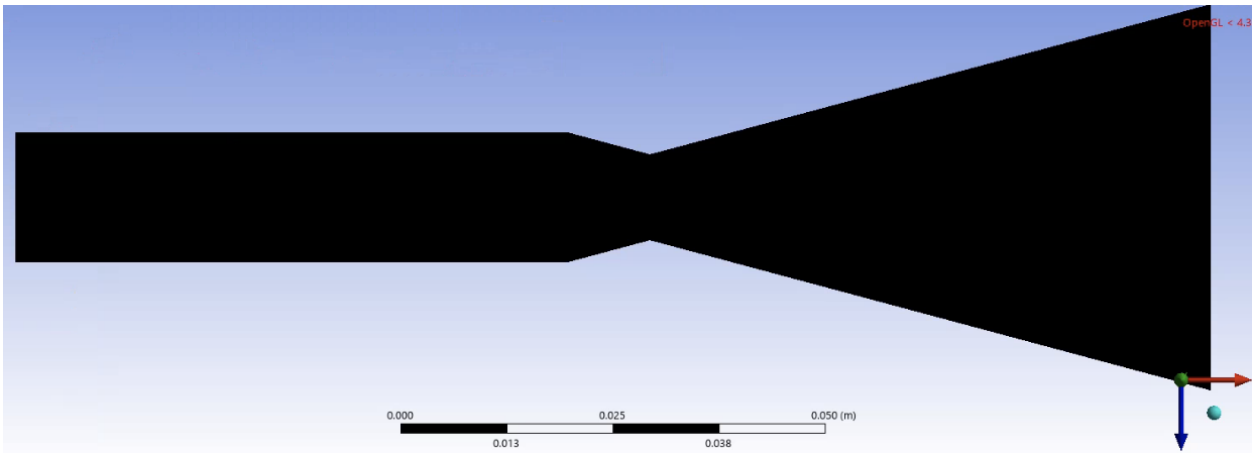


Figure 6.1: Nozzle 1 Mesh: Overview

Next, a Edge Sizing was applied to the wall boundaries, with a target element size of 0.00005 m (0.05 mm) and a growth rate of 1.1. This produces a finer, more gradual mesh immediately adjacent to the walls than the 0.1 mm bulk face size would otherwise provide. The shallower 1.1 growth rate ensures that this refinement near the walls blends smoothly into the coarser interior mesh rather than transitioning abruptly.

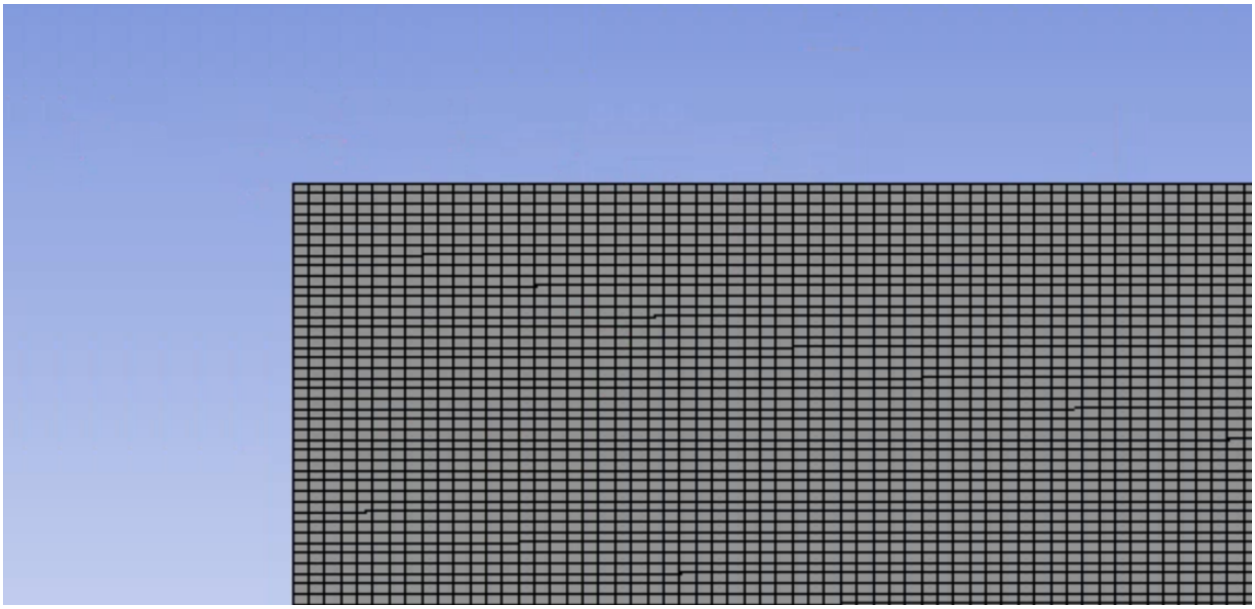


Figure 6.2: Nozzle 1 Mesh: Near-Wall

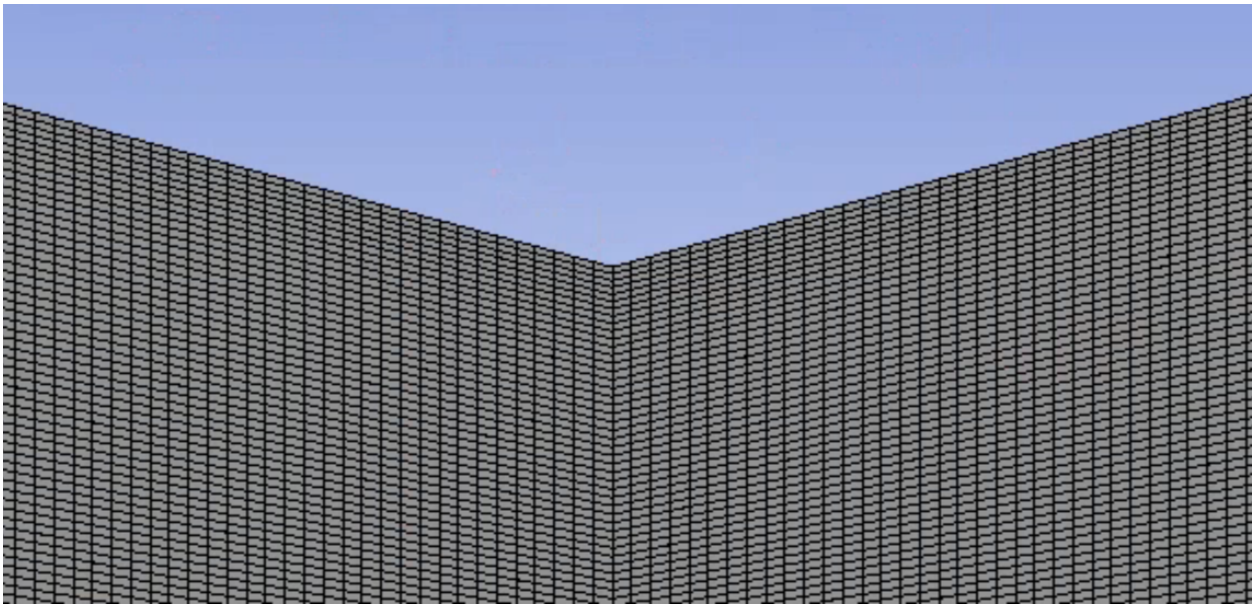


Figure 6.3: Nozzle 1 Mesh: Converging Diverging Wall Transition

The constant-area tube produces a uniform, evenly spaced structured grid, while the converging diverging nozzle section produces a mapped mesh that follows the angled convergent and divergent walls. In both regions, the mesh remains structured and orthogonal, with no visible skewed or collapsed elements at the throat transition.

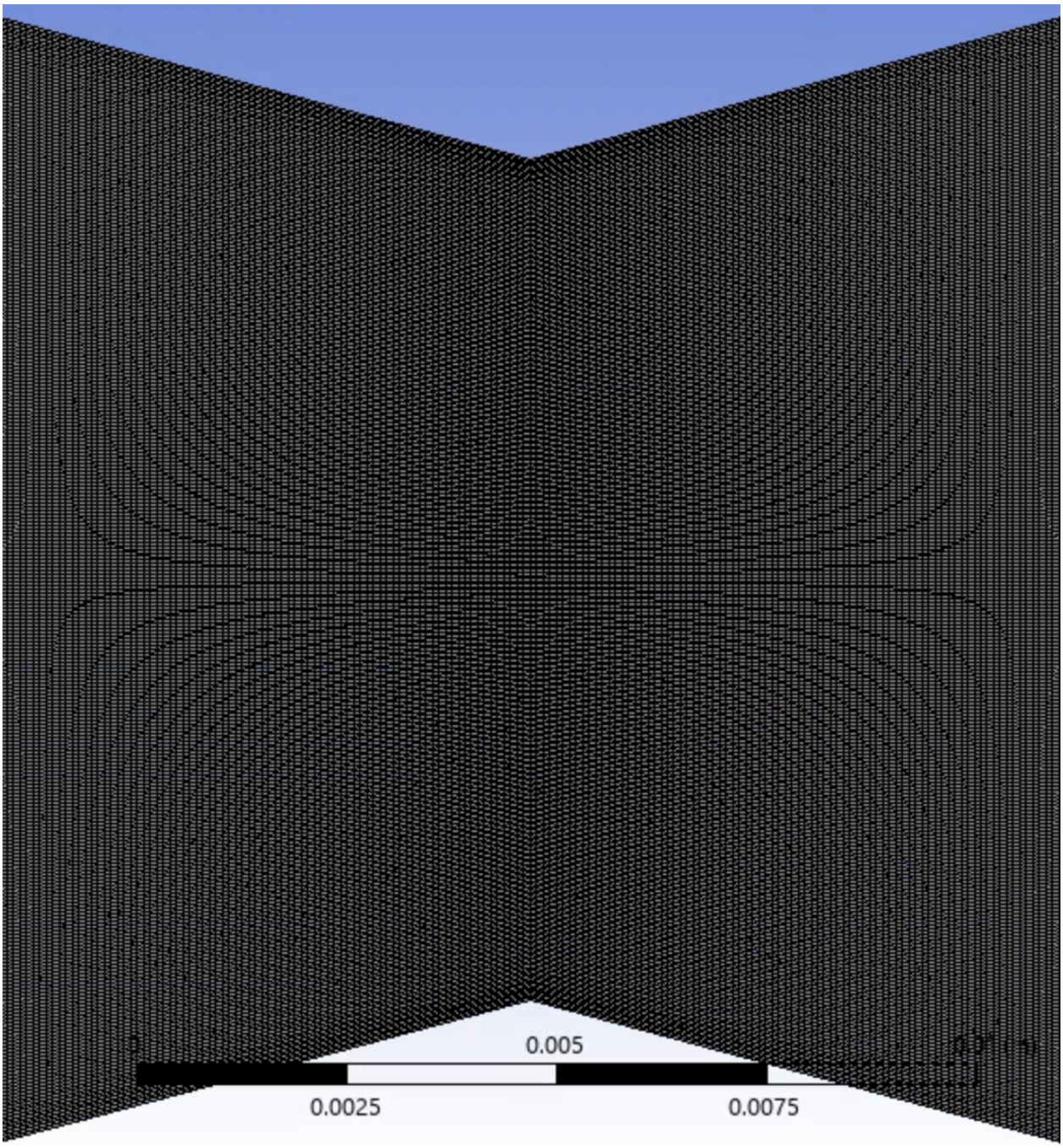


Figure 6.4: Nozzle 1 Mesh: Throat

6.3 Nozzle Configuration 2: Mesh Settings

Configuration 2 follows the same general framework established earlier, but with different absolute sizing values. This reflects the smaller physical domain and steeper wall angle of this geometry.

The edge sizing (0.05 mm, growth rate 1.1) along the walls was kept identical to the

Table 6.2: Nozzle 2 Mesh Settings

Parameter	**Value**
Face Meshing	both Driver and Driven
Face Sizing	0.0009 m
Face Sizing Growth Rate	1.2
Edge Sizing	0.00005 m
Edge Sizing Growth Rate	1.1

baseline, so that the near-wall resolution is treated consistently across both configurations. The bulk face sizing, however, was relaxed from 0.1 mm to 0.9 mm for this geometry. However, the ill effects of this change was minimized as much as possible given the similar order of magnitude and refined edge sizing.

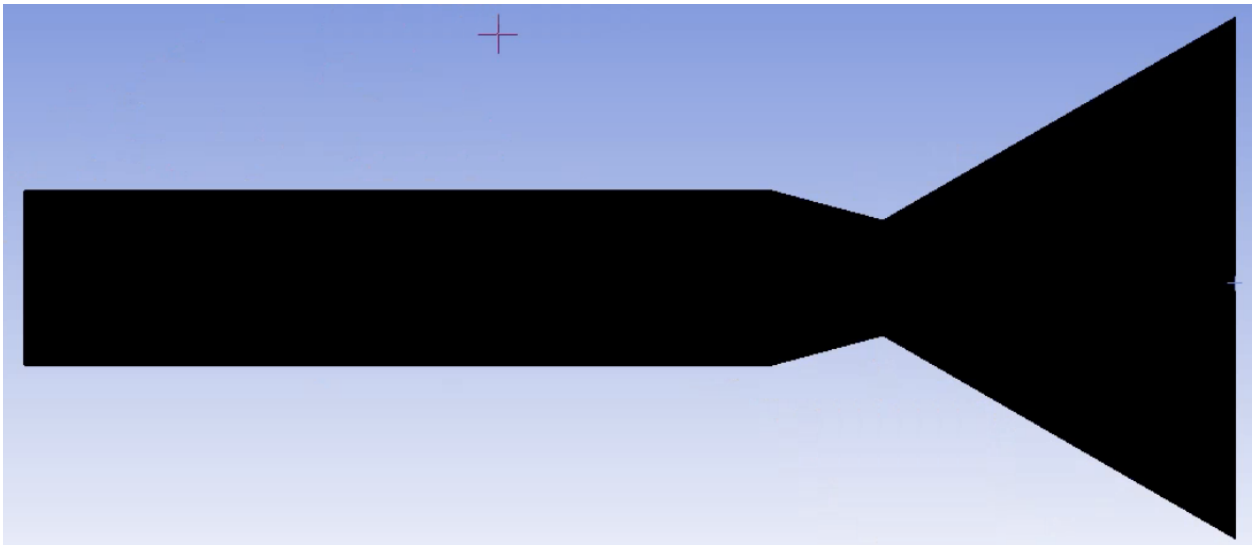


Figure 6.5: Nozzle 2 Mesh: Overview

Given that configuration 2's divergent section spans only 35.37 mm (versus 68.33 mm for the baseline), with the same 65 mm driver length shared by both, retaining the 0.1 mm bulk element size across this shorter domain would push the total cell count for Configuration 2 well above threshold. This change to 0.9 mm was primarily done since the previously chosen 0.1 mm sizing exceeded the cell count permitted on the student version of ANSYS. However, the fine 0.05 mm edge sizing at the walls is unchanged, the physically important gradients are resolved to the same standard in both configurations. This keeps the total cell count for Configuration 2 in a comparable, tractable range despite the smaller domain.

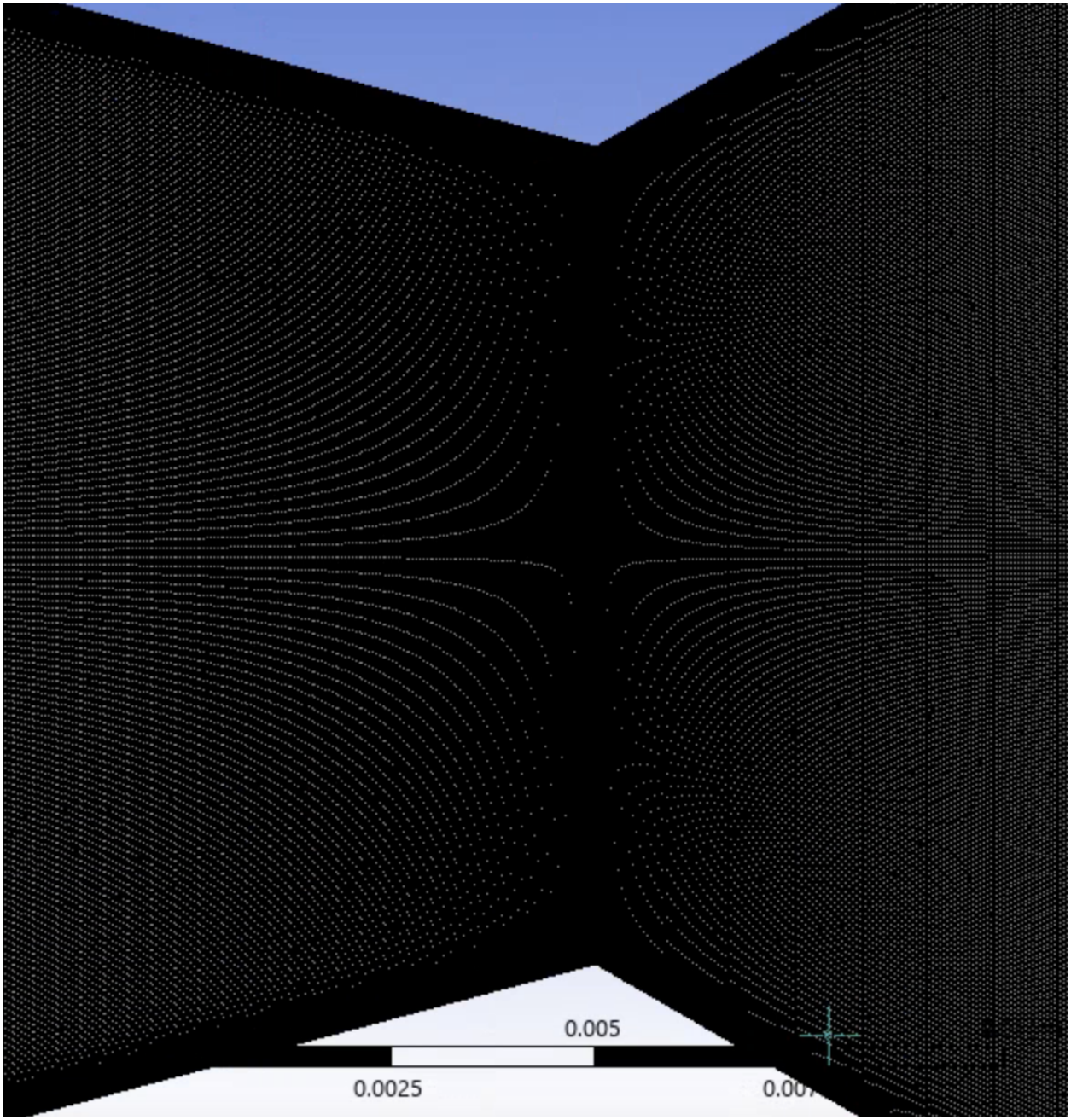


Figure 6.6: Nozzle 2 Mesh: Throat

6.4 Nozzle Configuration 3: Mesh Settings

Configuration 3 uses a mapped face mesh across both the driver and driven regions, a 0.0003 m face sizing with a 1.2 growth rate, and a 0.00009 m edge sizing on the walls with a 1.1 growth rate.

Table 6.3: Nozzle 3 Mesh Settings

Parameter	**Value**
Face Meshing	both Driver and Driven
Face Sizing	0.0003 m
Face Sizing Growth Rate	1.2
Edge Sizing	0.00009 m
Edge Sizing Growth Rate	1.1

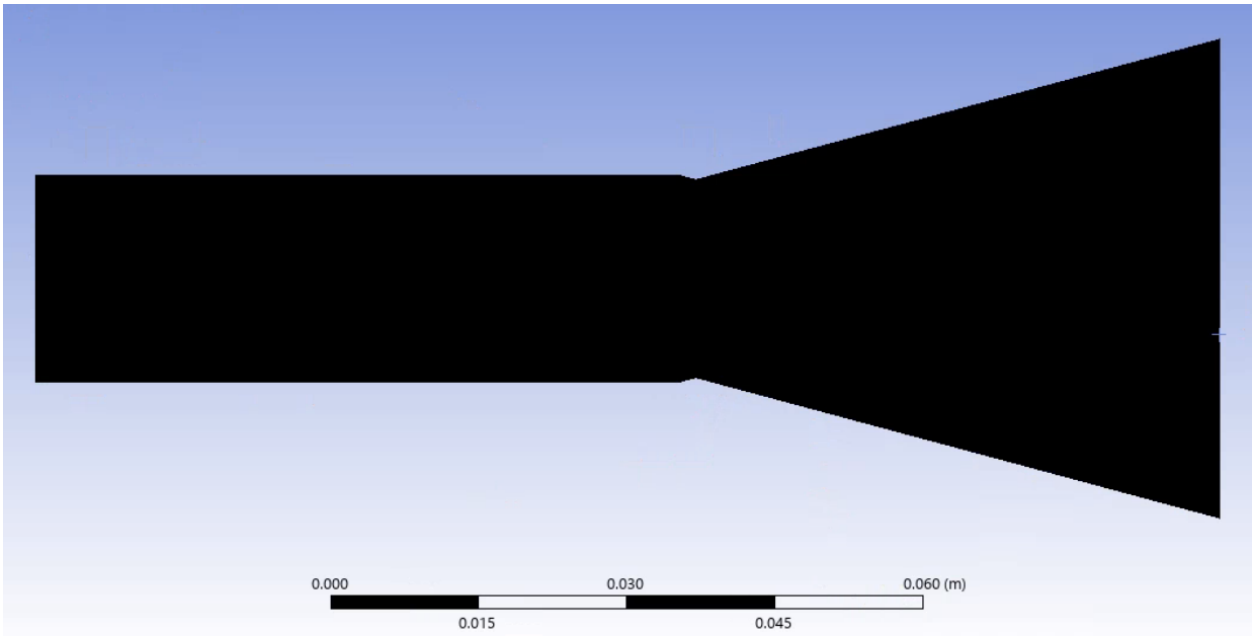


Figure 6.7: Nozzle 3 Mesh: Overview

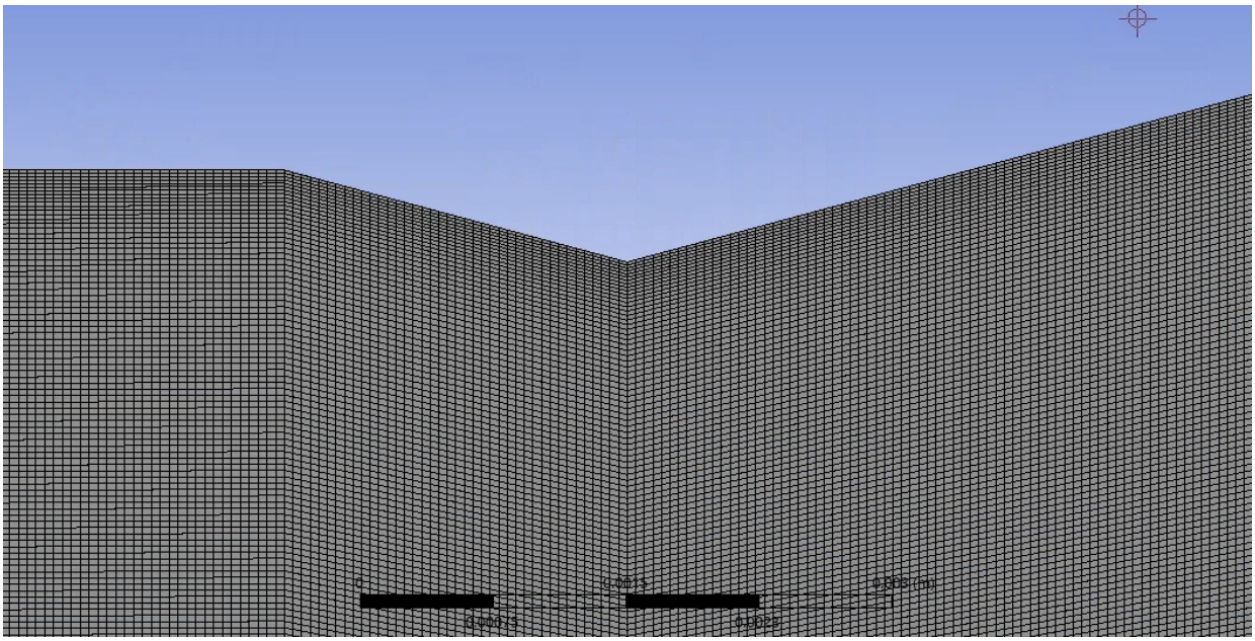


Figure 6.8: Nozzle 3 Mesh: Edge Sizing

This configuration is slightly different compared to the other configurations. Once again, this was done to stay within the bounds of the ANSYS version. However, the smaller divergent angle lets us get away with a slightly lower edge sizing.

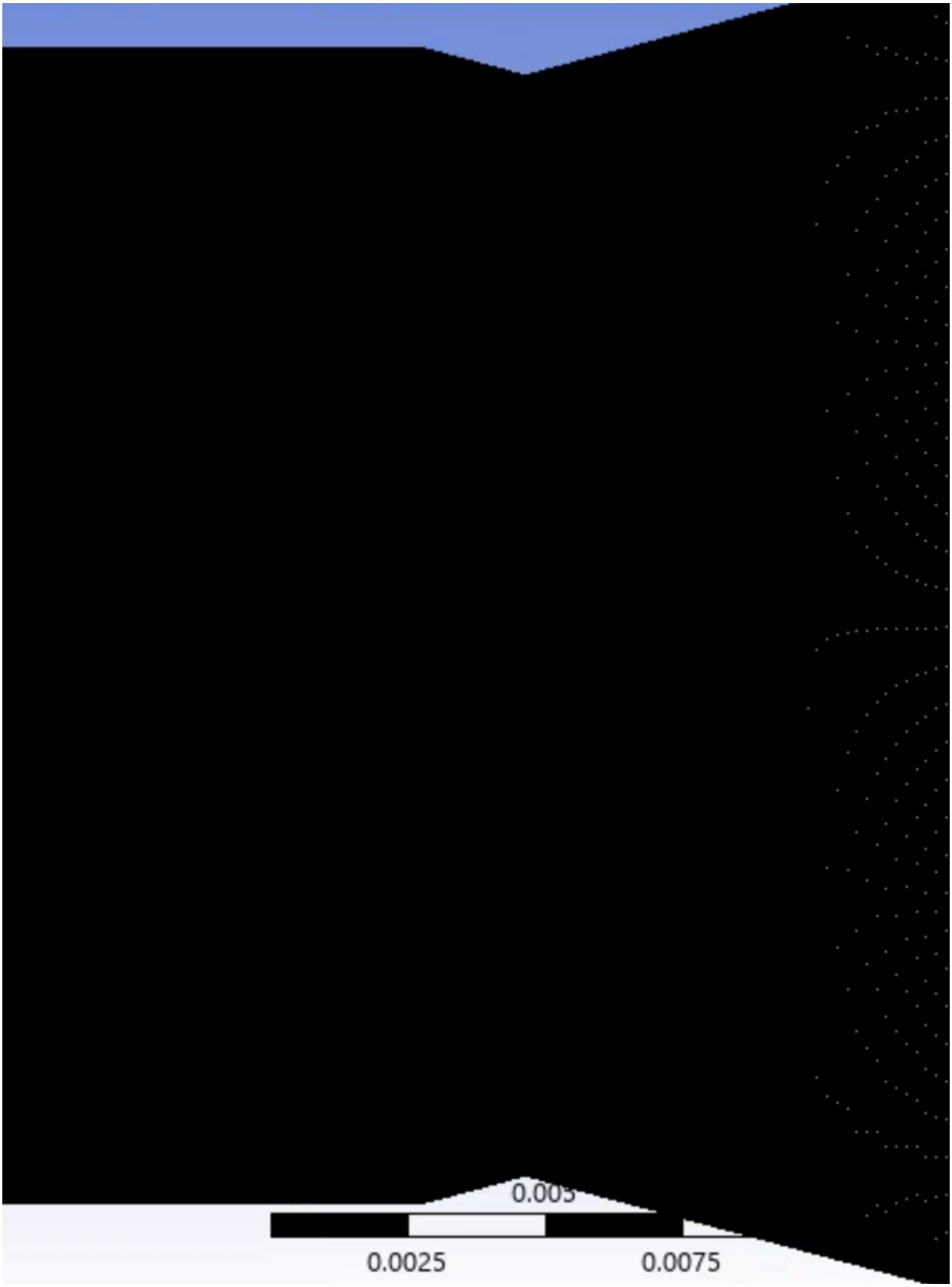


Figure 6.9: Nozzle 3 Mesh: Throat

6.5 Rationale for this Meshing Approach

ANSYS Student

Given that the ANSYS version used for this was a ANSYS Student, there are certain limitations when it comes to mesh refinement and further Fluent analysis. There is a cell count limitation with this version which proves rather hard for further mesh refinement.

Sphere of Influence

As mentioned before, a Sphere of Influence (SOI) control refines the mesh only within a fixed local region, which is centered around a specified point. Since the shock, contact surface, and reflected wave in all traverse the entire axial extent of the domain, a SOI would only benefit the portion of the simulation where the wave passes through that specific region. The majority of the transient is unrefined.

Triangular Meshing Structure

An unstructured mesh would automatically adapt to the divergence angle without requiring the mapped face mesh to be explicitly validated at the throat vertex. However, the cost is worse orthogonal quality and higher skewness than a mapped quadrilateral mesh. Since the domain here is a straightforward, using an unstructured mesh would only be trading away cell quality without gaining any geometric flexibility that is actually needed.

Boundary Layer Inflation Tool

Meshing in ANSYS offers a inflation layer control that grows layers (like prism structures) from the wall. This is a closely related approach and would likely produce a comparable near-wall resolution. But, it was not used here because the edge sizing already achieves the necessary gradation without introducing a separate mesh feature to validate. This keeps the meshing procedure consistent with the approach already established for all the configurations.

Chapter 7

Implementation and Simulation Settings

7.1 Turbulence Model Selection

Chapter 3 validated four viscosity treatments (inviscid, laminar, Spalart-Allmaras, and k–ω) against the same shock tube geometry, and both turbulence models produced reasonable results consistent with turbulent flow models. They produced a more diffused contact surface, a thicker reflected shock, and a delayed effective shock propagation speed relative to the inviscid and laminar cases (Section 3.4). However, k–ω is what was selected as the model for these nozzle configuration studies instead of Spalart-Allmaras for several reasons as mentioned below.

Two Equation Models capture Turbulence Generation along with Transport

As mentioned in previous sections, Spalart-Allmaras is a single equation model that solves a single transport equation for modified eddy viscosity. It was originally developed for external aerodynamic boundary layers on wings and turbomachinery blades. Such flows that are largely attached, mildly loaded, and evolve gradually along a surface. The k–ω model, however, solves two transport equations, one for turbulent kinetic energy k and one for the specific dissipation rate ω. This gives a physically complete representation of how turbulence is generated and dissipated. This matters for PDRE flow. The domain undergoes rapid, cyclic compression and expansion as the shock and reflected wave sweep through it, and a model that explicitly tracks the production of turbulent kinetic energy is better suited to capture the turbulence intensity.

Improved Accuracy for Near-Wall Effects

As noted in Section 2.1, the k–ω model offers improved near-wall performance and removes the need for wall functions. Given the small domain that is being used, the mesh is fine enough to support a refined wall mesh, and hence makes this approach better suited for wall effects. This makes k–ω a better match for the actual mesh being used than a model that benefits from wall functions.

Suitability for Strong Pressure Gradients

Spalart-Allmaras is known to perform less reliably under strong pressure gradients, flow separation, and pronounced streamline curvature (Section 2.1). These are exactly the conditions produced by a reflecting shock in a confined tube, and flow through a converging diverging nozzle. The k–ω model's separate transport of k and ω gives it more flexibility to respond to these strong varying gradients.

7.2 Flux Scheme, Time Integration, and Courant Number

Roe-FDS is a Riemann approximation solver. At each cell face, it linearized the local flow and reconstructs the flux based on the wave structure of the resulting linearized Riemann problem. It is well known for its ability to resolve contact surfaces and shocks with minimal numerical confusions compared to simpler upwind schemes. Since the contact surface trajectory is one of the primary features that will be tracked in the x-t diagrams for the nozzle configurations, a flux scheme that preserves this feature is directly relevant to the quality of the results.

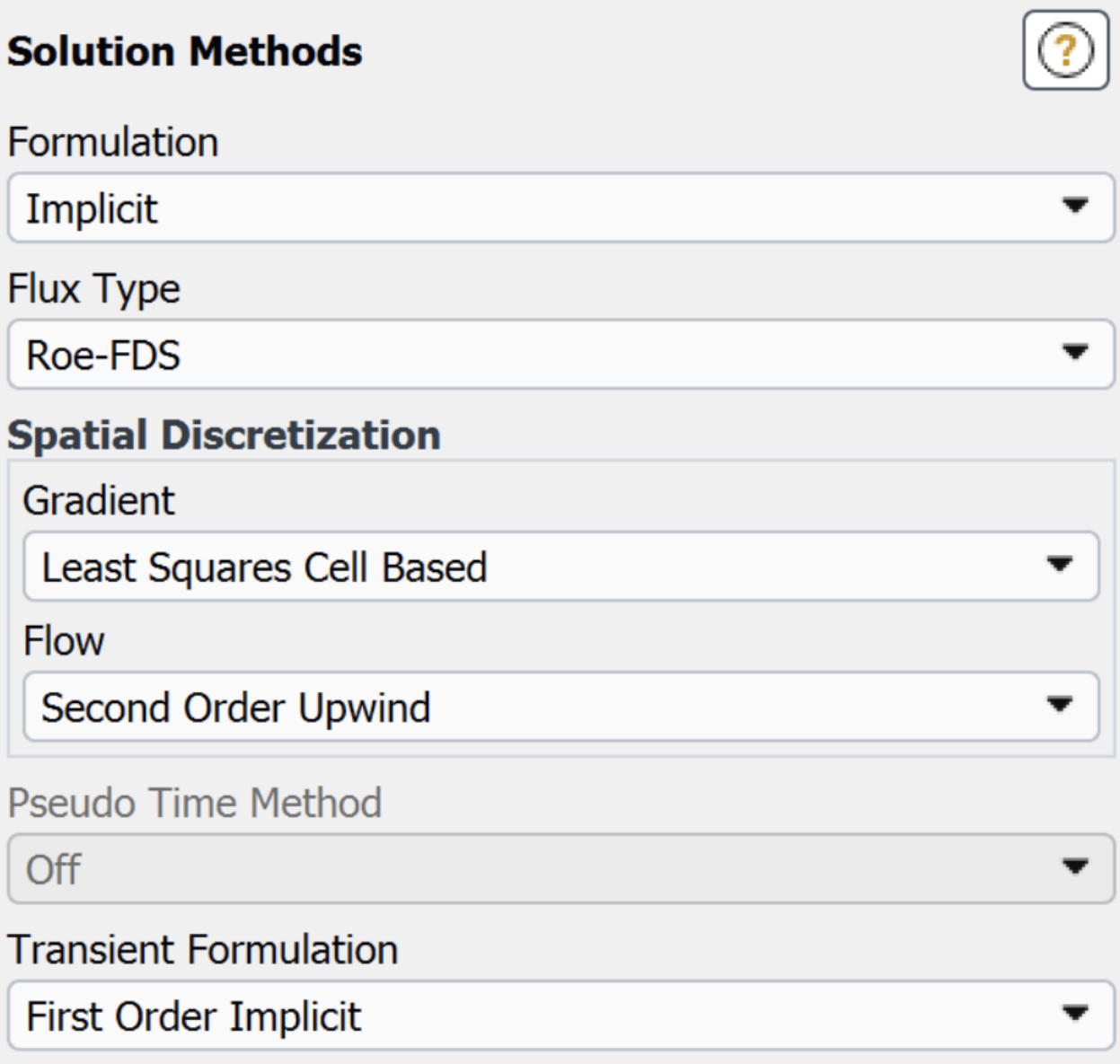


Figure 7.1: Flux Scheme Settings

A Courant number of 1 was used for these simulations to maintain numerical stability. The Courant-Friedrichs-Lewy (CFL) condition, as mentioned in Section 2.2, requires that wave propagation does not travel more than roughly one computational cell per time step for the solution to remain numerically stable and accurate. A Courant number of 1 is a standard and conservative upper bound. It is large enough to avoid unnecessarily small time steps (which would just increase solve time without improving accuracy), while staying within the stability limit for the transient wave physics being resolved.

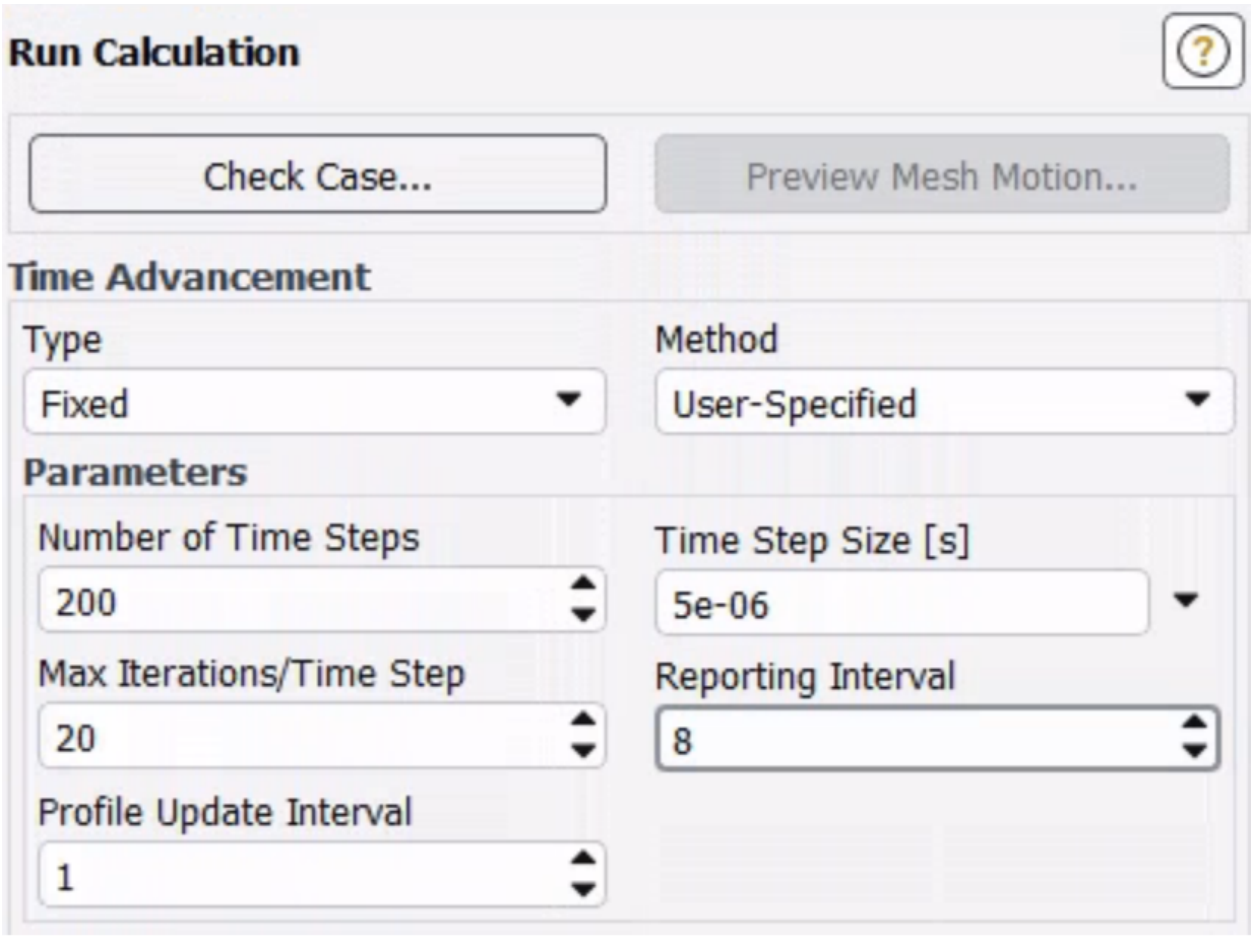


Figure 7.2: Time Advancement Settings

Time integration was fixed to a time step of 0.000005 s over 200 total steps. This corresponds to a total simulation time of 0.001 s (1 ms). This time scale is consistent with the domain of the nozzles contructed and being studied here. A 1 ms window is sufficient to capture shock propagation through the full domain and its subsequent reflection. A fixed time step was used deliberately for two reasons. First, it keeps the discretization identical across all cases, so any differences observed in the resulting x-t diagrams between Configurations 1, 2, and 3 can be attributed to the geometry itself rather than to a difference in time resolution between runs. Second, it maintains consistency with the shock tube simulation performed in Chapter 3. Each time step is limited to a maximum of 20 iterations. 20 iterations per time step was sufficient to bring the residuals for continuity, momentum, and energy down to an acceptable level and towards convergence.

7.3 Boundary Conditions, Initialization, Patching, and Report Files

All the simulations use the same boundary conditions established during the shock tube simulation. No inlet, or outlet present. The goal is to validate wave propagation, reflection,

and area-based acceleration within a closed system, using the same initialization strategy throughout.

All outer boundaries, such as the top and bottom edges of both the constant-area tube and the converging diverging nozzle are set to a Wall boundary condition. This wall will be assigned a no-slip velocity and an adiabatic condition. No-slip is required to resolve the boundary layer physics near the wall that the k-ω model specifically intends to capture. A slip condition at these boundaries would eliminate the velocity gradation that can be observed along the walls. This would defeat the purpose of both the turbulence model selection and the wall mesh refinement strategy. Adiabatic walls are used because the simulation spans a very short time window (0.001 s). Conductive heat loss through the tube and nozzle walls is expected to be negligible in such a short time-span. This assumption would not be made if extended to examine long-term, multi-cycle operations.

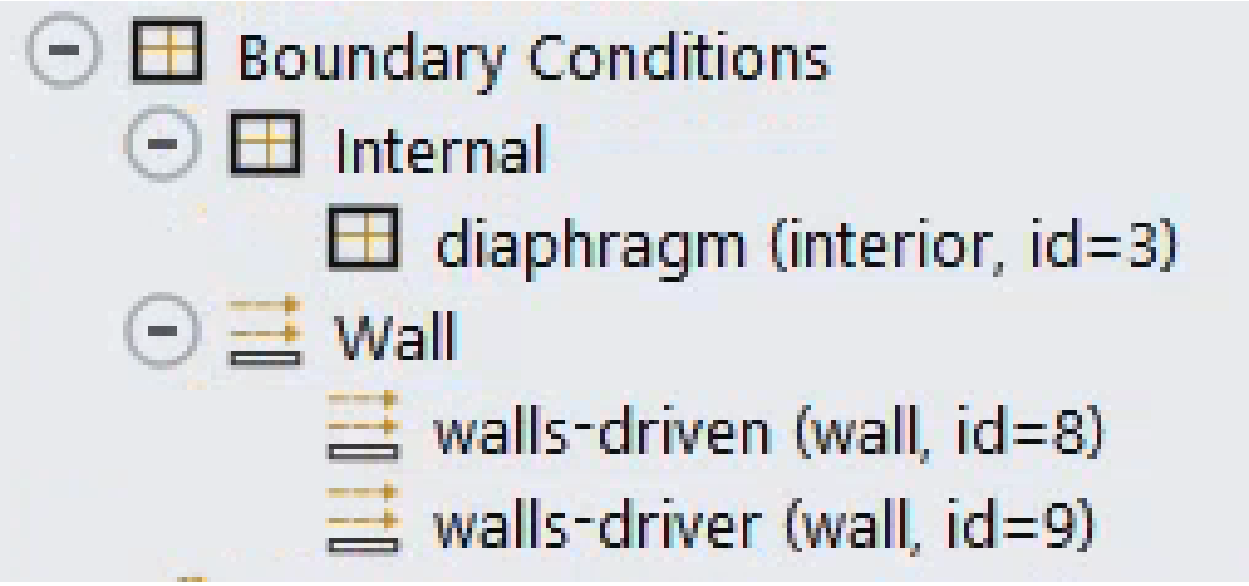


Figure 7.3: Boundary Conditions

The Driver and Driven regions are set to Interior boundary conditions. This allows the solver to treat the two regions as a single continuous fluid domain, with the patched pressure difference between them driving the wave propagation. This avoids the added complexity of modeling diaphragm ruptures, which is not the focus of this project. This approach is a simplification used in classical shock tube analysis and was already validated against theory in Chapter 3.

The domain was initialized at 300 K. The Driver zone was patched to 100,000 Pa and the Driven zone was patched to 10,000 Pa, reproducing the same 10:1 pressure ratio validated against the shock tube. This keeps the driving conditions for the transient wave identical throughout this project, so that differences in behavior can be simply attributed to geometry or turbulence modeling.

A centerline (line/rake surface) was placed along the geometric axis of the nozzle. A pressure report file was added with the control surface as the centerline. This centerline data, exported at every time step, is the raw input that will be used to construct the x-t diagrams.

Chapter 8

CFD Results

For each configuration, the scaled residuals, the vertex-averaged centerline pressure, and a contour of static pressure. The reconstructed centerline pressure x-t diagram used to visualize shock and expansion wave propagation (following the same x-t diagram methodology validated in Chapter 3) are presented in the following chapter (Chapter 9).

8.1 Nozzle Configuration 1: Baseline

8.1.1 Residuals and Centerline Pressure Report

The scaled residuals for Configuration 1 were tracked over around 4,000 iterations of the transient run. As can be seen in Figure 8.1, the y-velocity residual remains the highest of all the quantities, sitting around an order 1-10 for most of it. x-velocity and energy clump between roughly 0.1 and 0.01. The turbulent kinetic energy residual (k) begins very low (around 0.00001), consistent with the low turbulence initial condition, and then steadily rises as the shock and flow develop, reaching roughly 0.01 by the end.

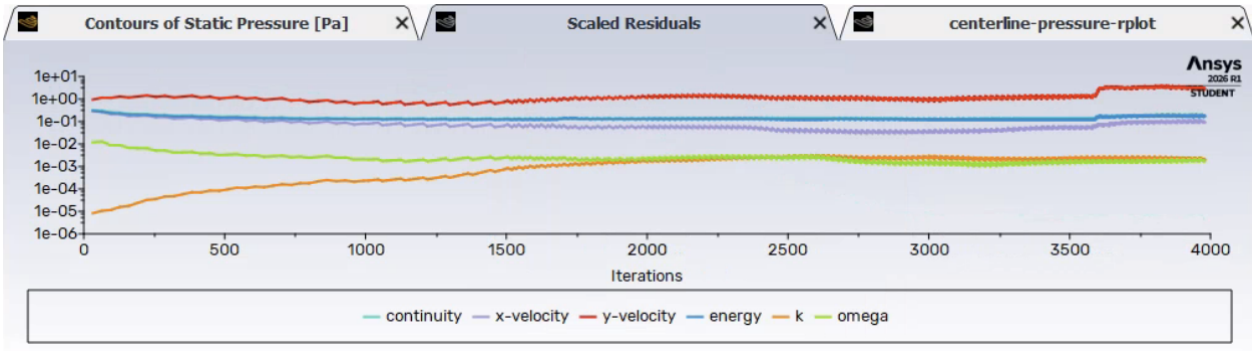


Figure 8.1: Nozzle 1 Simulation Residuals

The specific dissipation rate residual (ω) starts near 0.01 and gradually declines. There is a increase in the y-velocity residual around iteration 3,500. This coincides with continued wave activity late in the transient. This is consistent with new flow structures (the reflected wave reaching further downstream) being introduced late in the simulation rather than indicating

a stability problem, given none of the monitored residuals diverge or oscillate with growing amplitude afterward.

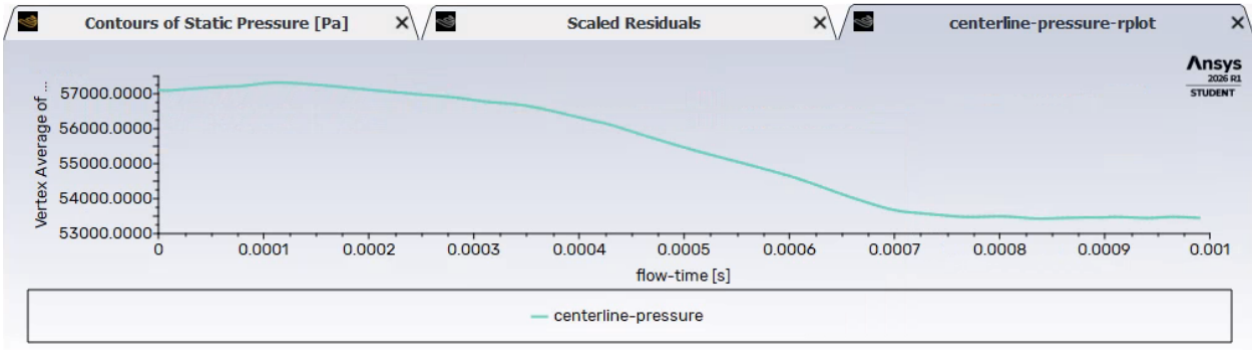


Figure 8.2: Nozzle 1 Centerline Pressure Plot

The vertex-averaged centerline pressure plot for Configuration 1 begins around 57 kPa and remains pretty flat until around t = 0.0004 s. After that it decreases smoothly to approximately 53 kPa by t = 0.0007 s. It then plateaus for the rest of the simulation. This is a very simple and stable trend relative to the other two configurations, which can be seen further below. It has smooth decay without oscillation.

8.1.2 Pressure Contour

The static pressure contour for Configuration 1 (Figure 8.3) provides a spatial snapshot of the wave. The high pressure region (red) occupying the driver and convergent section is the portion the expansion fan has not reached yet. Moving downstream, the continuous color gradient from red through orange and into green marks the expansion fan. The smoothness of this gradient is consistent with an isentropic expansion process, exactly as expected.

Immediately around the throat, a rapid pressure drop dominates the divergent section. This is the signature of the flow accelerating through a constricted area. As the passage narrows to the throat and then expands, the flow undergoes a Prandtl-Meyer like expansion around the throat corner. This converts pressure into velocity and produces the sharp additional pressure drop that is visible.

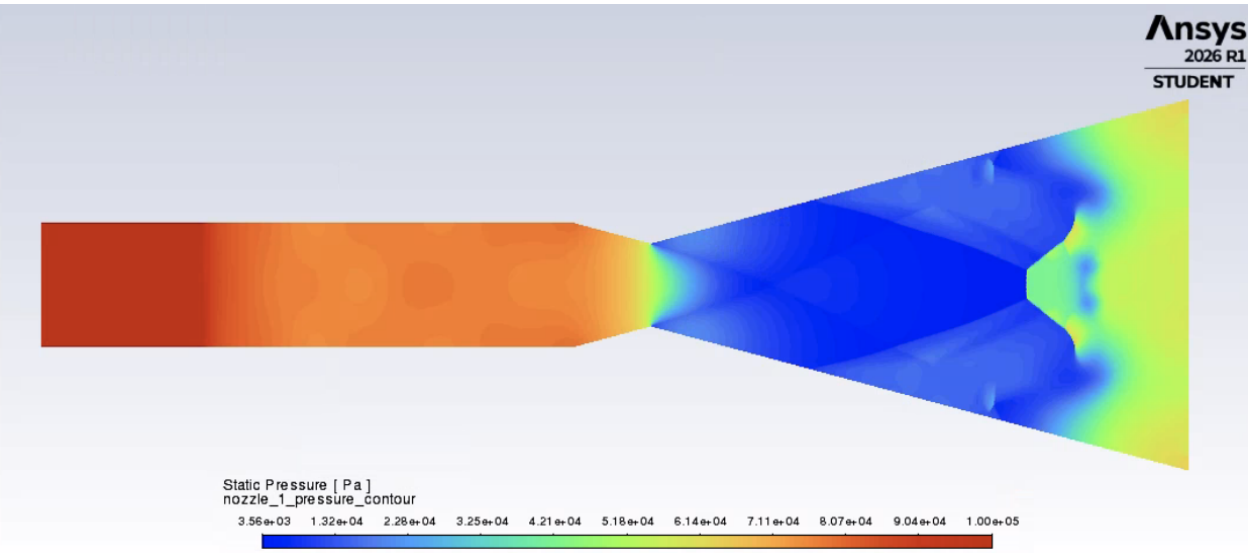


Figure 8.3: Nozzle 1 Pressure Contour

The green and yellow parts breaking up the uniform blue field near the nozzle exit are the signature of the shock structure diffusing. This contour captures the primary expansion fan, with the patchiness at the exit, marking the onset of a diffused wave structure. This is consistent with the moderate (15°) divergence angle keeping the core flow attached and organized while still allowing the anticipated turbulent breakdown near the exit.

8.2 Nozzle Configuration 2: High Divergence

8.2.1 Residuals and Centerline Pressure Report

Configuration 2 has residuals (Figure 8.4) that are broadly similar in magnitude and shape to Configuration 1. Y-velocity remains the highest monitored quantity. x-velocity and energy are closely clustered at lower levels. The turbulent kinetic energy residual rises from a very low starting value (0.0000001) up toward 0.01 as turbulence develops. ω declines gradually. No divergence or sustained oscillation is present. This indicates the solver setup remained stable for this geometry despite the steep divergence angle.

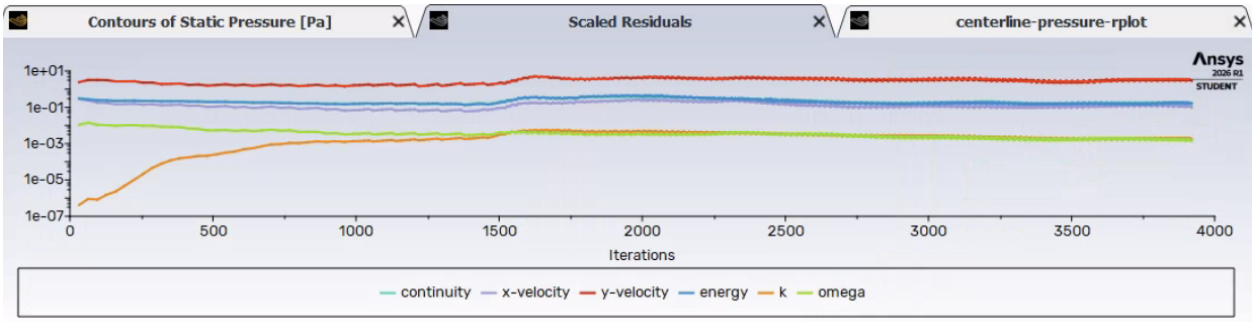


Figure 8.4: Nozzle 2 Simulation Residuals

The centerline pressure plot for this (Figure 8.5) is very different from configuration 1. It starts around 71 kPa, decreases to a minimum (70 kPa) around 0.00025–0.0003 s, and then rises again back to 78 kPa. This dip and subsequent rise behavior is consistent with the prediction made in Section 4.3.2.

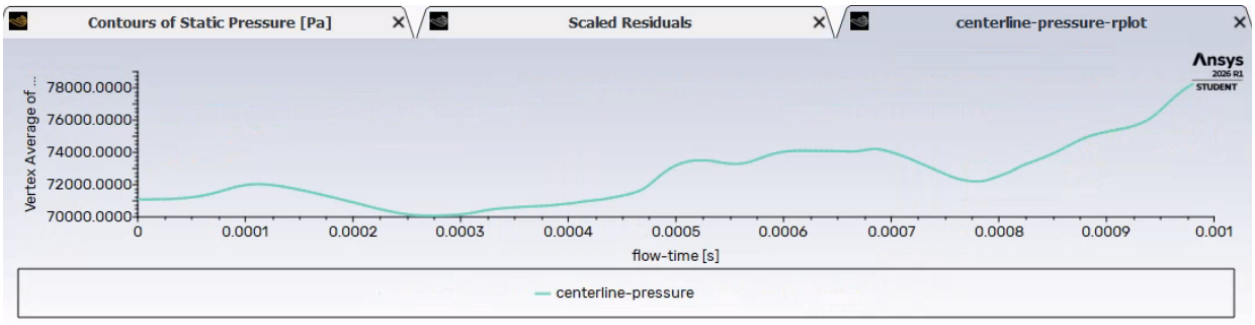


Figure 8.5: Nozzle 2 Centerline Pressure Plot

8.2.2 Pressure Contour

The pressure contour shows the same high pressure in the driver side and expansion fan through the convergent section as the first configuration. But the divergent section is significantly different. The entire divergent section here is filled with a mix of red, green, and blue patches with no single dominant pressure region. Since the divergent section is roughly half the axial length of the baseline's (Section 5.2.1), the initial wave reaches the end wall and reflects back upstream much sooner. By the time this contour is captured, the outgoing expansion and the returning reflected compression wave are already superimposed, instead of occupying separated regions.

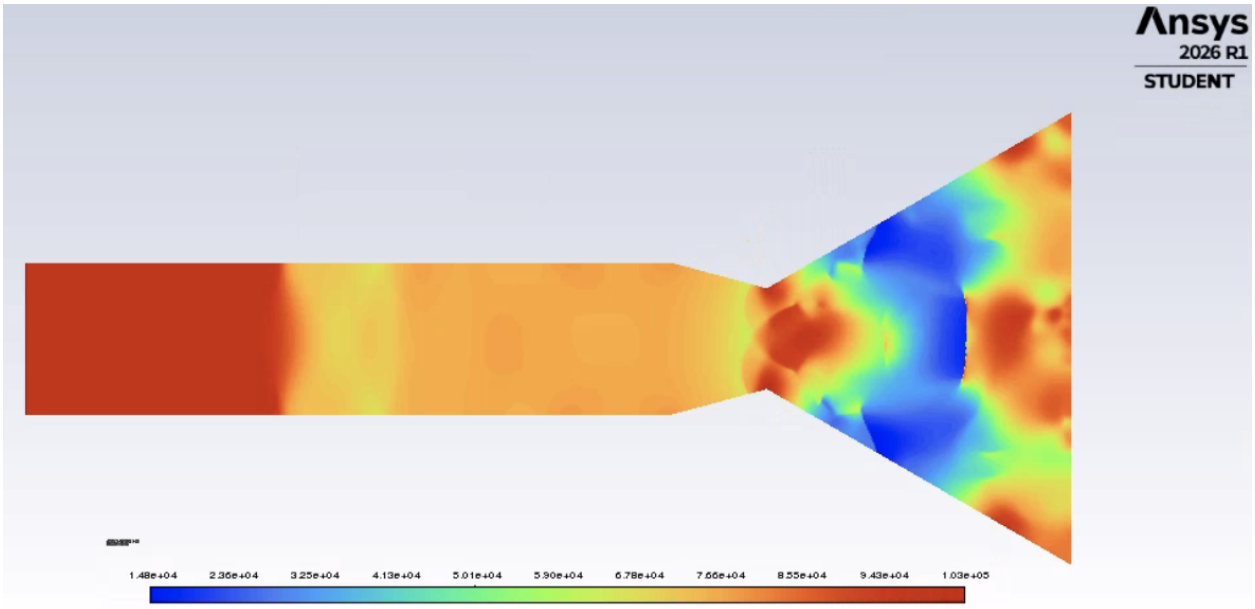


Figure 8.6: Nozzle 2 Pressure Contour

The steeper divergence angle compounds this effect. A more aggressive diverging wall is more prone to flow separation under the high pressure gradient. Rather than a single shock front and a single reflected wave, this configuration's high divergence walls appear to break the wave up into a number of smaller compression and expansion regions that interact.

8.3 Nozzle Configuration 3: High Mass Flow

8.3.1 Residuals and Centerline Pressure Report

These residuals (Figure 8.7) follow the general pattern as the other two configurations. No instability or divergence is observed.

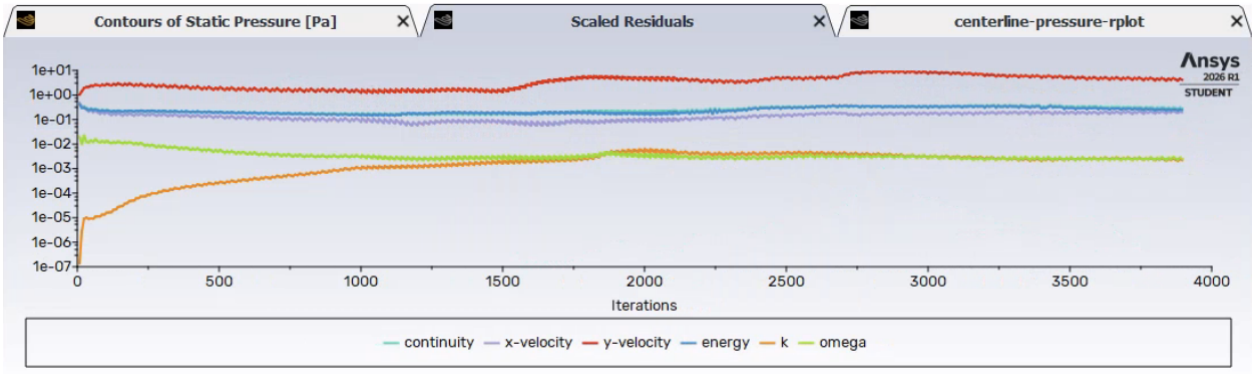


Figure 8.7: Nozzle 3 Simulation Residuals

The centerline pressure plot is very oscillatory when compared to the other configurations. It starts at 59.5 kPa, rises to a small peak of 60.3 kPa at 0.00015 s. It then decreases to 56.5 kPa around and then rises again with more oscillation. Unlike smooth decay in configuration 1 and the single dip and rise in configuration 2, configuration 3 shows multiple local extremes. This is in agreement with a larger throat diameter. More mass flow passing through a wider throat hence the pressure disturbances reflect and interact more.

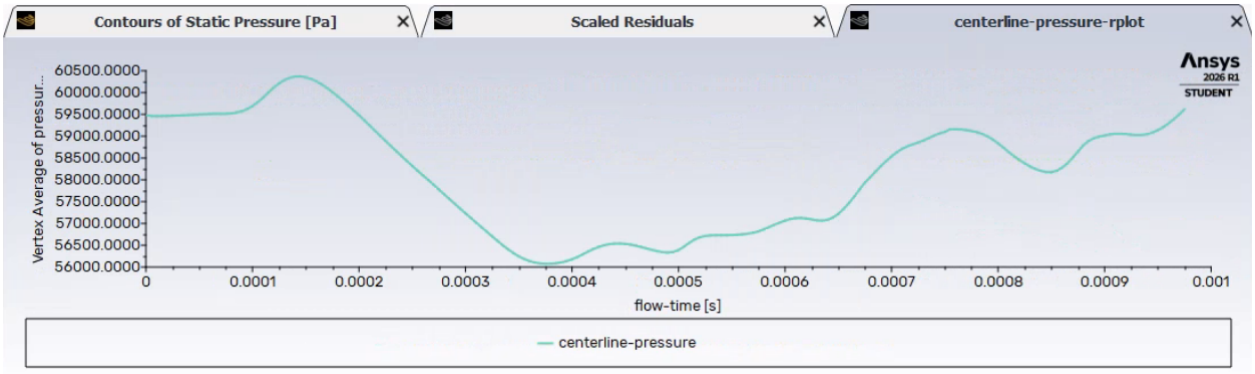


Figure 8.8: Nozzle 3 Centerline Pressure Plot

8.3.2 Pressure Contour

Unlike before, a pronounced low pressure region is present at the throat. This is the clearest and most concentrated Prandtl-Meyer expansion signature of the three configurations. The flow undergoes a more intense expansion right at the throat. This produces a confined low pressure region rather than a gradual pressure drop.

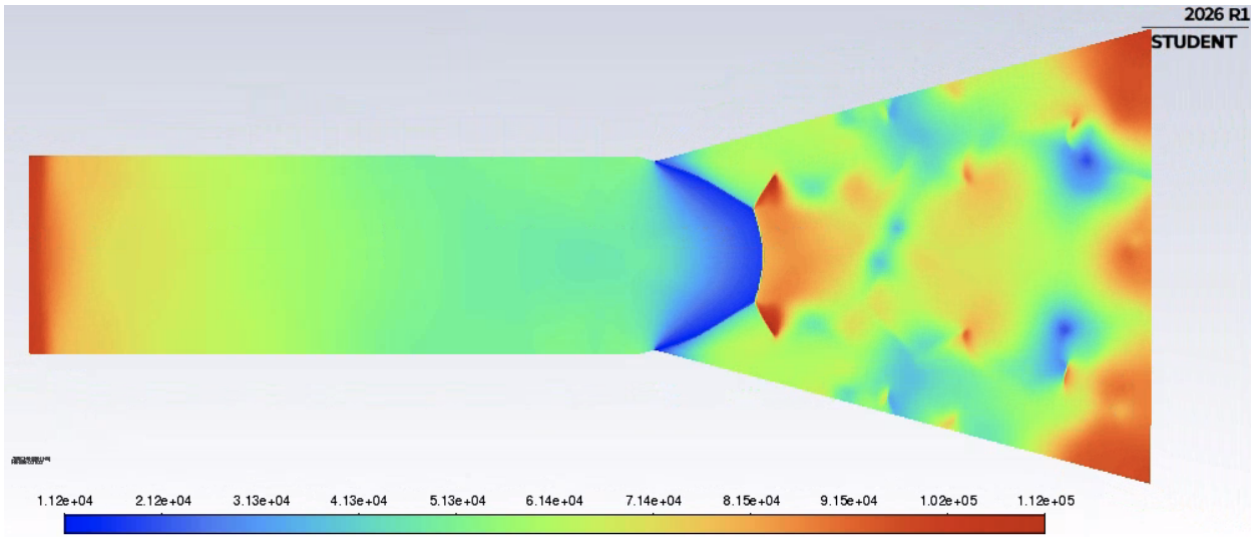


Figure 8.9: Nozzle 3 Pressure Contour

On the driven side, a disorganized pattern of red, green, and blue are visible. It is visibly the most chaotic of all the configurations. This contour suggests that there are multiple overlapping compressions and expansions. This is consistent with the oscillatory centerline pressure plot.

Chapter 9

x-t Diagrams and Discussion

9.1 x-t Diagrams and Benefit of Each Nozzle

Taken together, the three x-t diagrams tell a consistent story about how nozzle geometry shapes the character of the detonation wave as it moves from the driver section to the rest of the nozzle. Each diagram contains the same basic wave structure that was established in the Chapter 3. A MATLAB script was run on the pressure data for each configuration to construct the x-t diagrams (attached in the Appendix)

9.1.1 Nozzle 1

Nozzle 1 produced the cleanest and most shock-tube like x-t structure. The expansion fan on the driver side is a smooth and well defined. Only near the nozzle exit does the diagram begin to show the higher pressure streaks associated with turbulent mixing. So, for most of its length, this configuration behaves close to the idealized, quasi-1D standard. A lower divergence angle keeps the flow attached and the flow organized. However, the mixing effects only show up when and where the theory (Section 4.3.1) predicted they would be smallest.

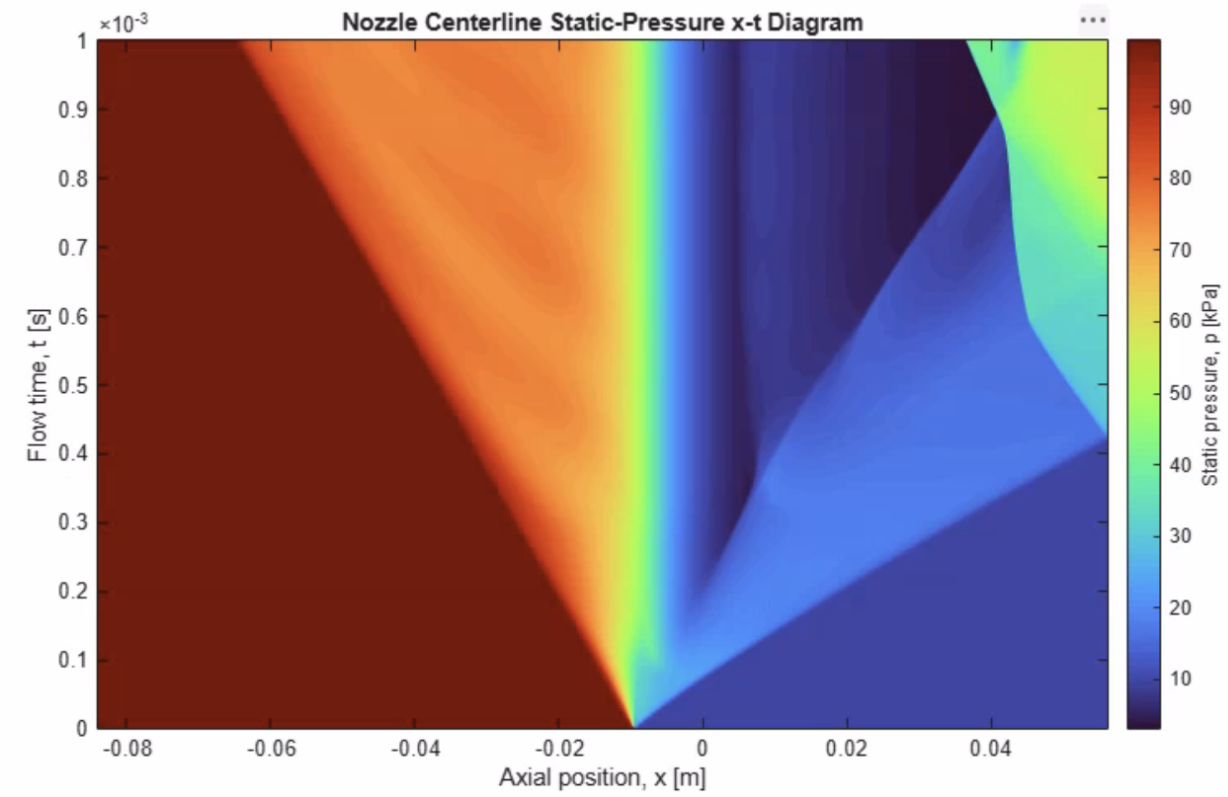


Figure 9.1: Nozzle 1 x-t Diagram

9.1.2 Nozzle 2

Two x-t diagrams were plotted for this nozzle. The first attempt (Figure 9.2) produced a result with a abrupt horizontal discontinuity in the low-pressure (blue) region on the right side of the diagram, where the pressure field appears to jump discontinuously rather than evolving smoothly, as would be expected of a continuously propagating wave. Configuration 2's throat is pinched to a very narrow width over a short axial distance, the mesh cells do not sit exactly on the nominal y = 0 centerline near the throat. Hence, the MATLAB code used to construct the x-t diagram instead must've selected the nearest available cell at each axial point. For such a compact geometry, the nearest cell available can shift.

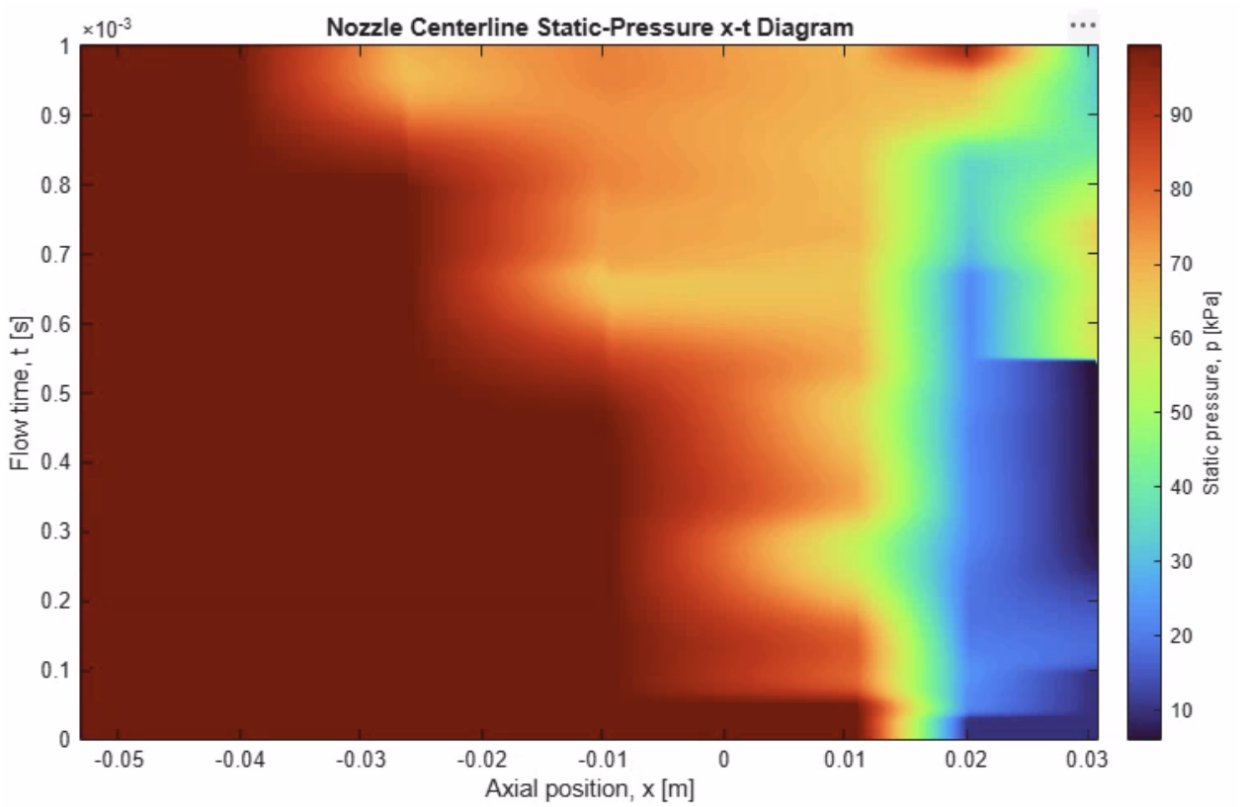


Figure 9.2: Nozzle 2 x-t Diagram: Trial 1

Trial 2 (Figure 9.3) was generated to see if Trial 1 was a one-off. Trial 2 shows a somewhat similar diagram as of Trial 1. This suggests that there is consistency in our simulation and it has less to do with a chance numerical error. This also does show that there might be a chance that the x-t diagram script used might not have worked as well as it did for the first configuration.

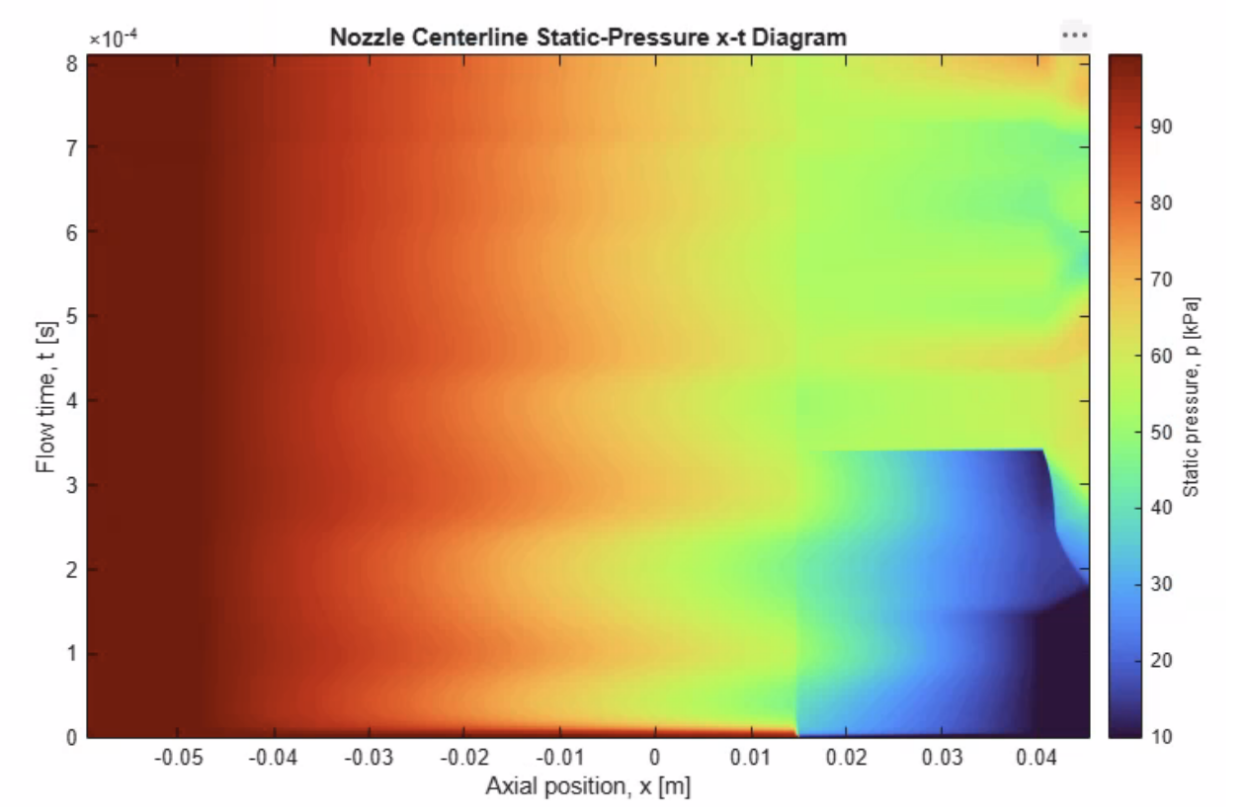


Figure 9.3: Nozzle 2 x-t Diagram: Trial 2

9.1.3 Nozzle 3

Figure 9.4 shows a clearer x-t diagram that what was produced in Configuration 2, however, more turbulent than Configuration 1. Physically, this is consistent with a wider throat. More gas is moving through a larger opening which produces a more energetic and quickly interacting wave structure. This is the turbulent patches of high and low pressure that we see beyond a flow time of 0.3 ms. However, the expansion fan prior to that is sharp and somewhat mirrors the resolution of the first configuration.

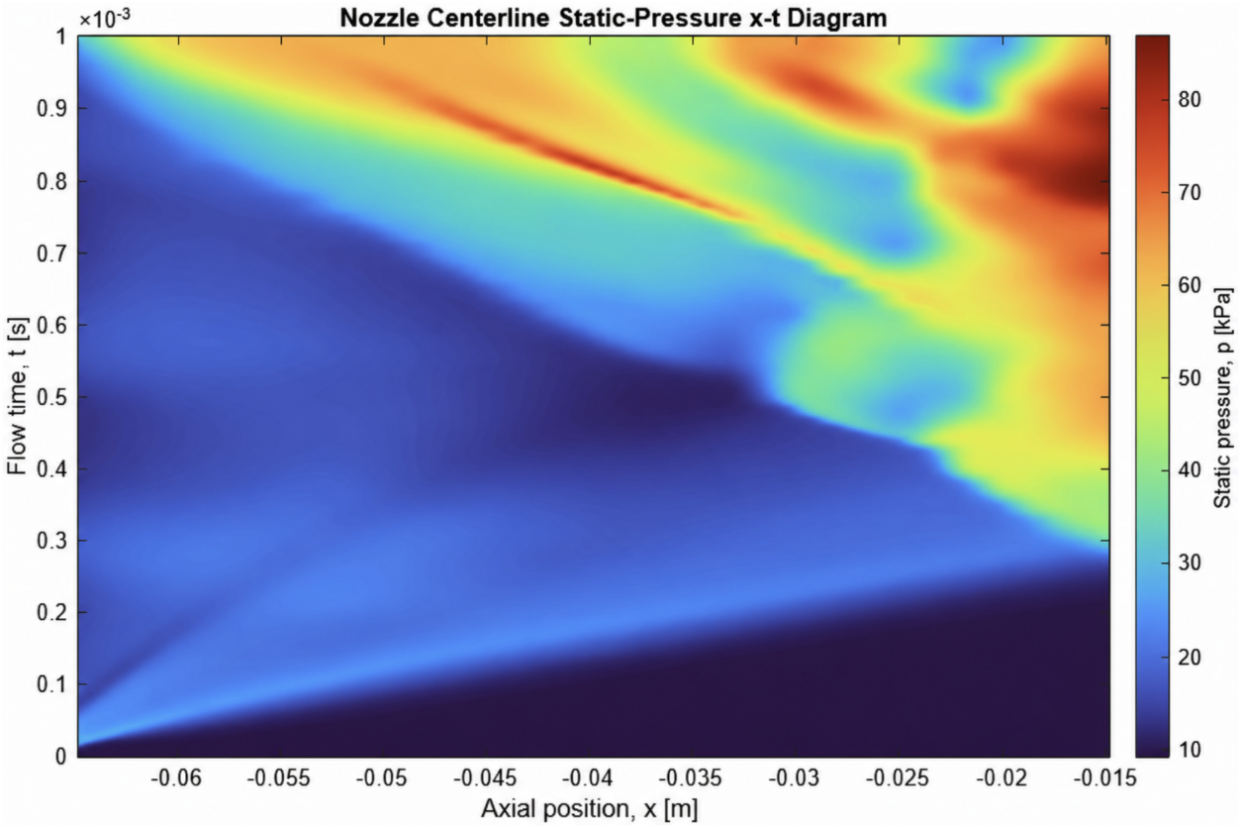


Figure 9.4: Nozzle 3 x-t Diagram

9.2 Discussion and Conclusion

Based on the observations made from the x-t diagrams for each of the nozzle configurations, the first nozzle is a general purpose geometry and is the most stable. As for Configuration 2, the earlier reflection timing suggests this geometry could plausibly be favorable during the early, high-chamber-pressure portion of blowdown, where absolute expansion efficiency matters comparatively less because the chamber pressure itself is still high. Configuration 3 aligns more with the low-pressure portion of a PDRE blowdown. While it was expected that no single one of the three geometries tested here would show CFD behavior that is uniformly deemed the best across the entire simulated window, each shows a different strength tied to a different part of the pressure history of PDRE functioning. These findings from all the simulations run, and x-t diagrams studied in this project is what connects directly back to the motivation of this project that was established in Chapter 1: if a single fixed nozzle geometry cannot be simultaneously well matched for all the stages of a PDRE blowdown, then a nozzle capable of changing its geometry within a cycle has a clear performance advantage behind it.

This is precisely the motivation and the promise of the high-temperature flexible ceramic nozzle concept introduced in Section 1.1. A nozzle built from ultra high-temperature ceramic matrix material can be capable of surviving repeated, rapid thermal and pressure cycling, and would be mechanically suitable enough to change its divergence angle or throat area in

Table 9.1: Comparison of Configuration Quantities

Quantity	**Configuration 1**	**Configuration 2**	**Configuration 3**
Initial reported centerline pressure	∼57 kPa	∼71 kPa	∼59.5 kPa
Minimum reported pressure	∼53 kPa	∼70 kPa	∼56.5 kPa
Maximum reported pressure	∼57 kPa	∼78 kPa	∼60.3 kPa
Pressure behavior	Smooth decay	Dip then rise	Oscillatory
x-t quality	Clear	Discontinuous	Clearer but complex
Numerical observation	Organized wave interactions	Early interactions	Stronger interacting waves

response to the instantaneous chamber conditions. The results of the simulations provide a snapshot/geometry-specific picture of what such a nozzle would be able to do achieve over the course of a single blowdown event. They can be classified as follows:

- Early in blowdown, when chamber pressure is high, the nozzle would ideally behave more like Nozzle Configuration 2. A compact and more divergent passage will minimize the delays to return the system to a state ready for the next detonation cycle. The higher mixing losses introduced by this nozzle would be assumed as a known trade-off considering the faster cycle turnover that it provides.

- As the chamber pressure falls during the middle of blowdown, the nozzle would ideally need to resemble Configuration 1, where the moderate and stable geometry would mimic quasi-1D expansion behavior best.

- Later in blowdown, the chamber pressure falls further. Nozzle Configuration 3 provides a larger throat area and a lower expansion ratio. This resists the overexpansion that a fixed geometry with a high expansion ratio (such as the other two nozzles) would lose performance to as the chamber pressure drops.

A fixed nozzle can only ever be one of these three. But as this project's own results show, any one of these nozzles for an entire cycle means being well matched to only part of that cycle. This means being progressively mismatched for the rest of it. A flexible ceramic nozzle capable of morphing between something resembling Configuration 2's compact, high-divergence profile early in blowdown, Configuration 1's balanced profile through the middle of the cycle, and Configuration 3's larger-throat, lower-expansion-ratio profile late in blowdown would, in principle, track the actually-changing expansion requirement throughout the cycle rather than being locked into a single compromise. This directly addresses the loss, from steady combustion losses due to mismatch is geometry with a rapidly changing chamber condition, that the project set out to investigate. The more promising path suggested by the results is a nozzle that changes shape within a cycle, using the specific geometric behaviors identified here.

Chapter 10

Future Work and Limitations

The x-t diagrams for the second nozzle show discontinuous jumps which might be linked to inconsistent centerline cell sampling as mentioned earlier. Finding a better centerline-extraction method can help with better understanding of Nozzle 2. Additionally, the simulations did not consider reactive detonation chemistry. All cases used a simplified patched pressure ratio. So ignition and heat releases are not modeled. Introduce finite-rate chemistry or a more detailed detonation initiation model in place of the CJ approximation, to capture ignition and DDT effects. In the future, running an actual moving-boundary case that changes between the three tested geometries within a single blowdown cycle can help better test the concept being championed by this project. This project focussed on a short window and small scale. Each case covers only 0.001 s on a millimeter-scale domain. This is not sufficent enough to warrant a gospel-like conclusion. Longer duration simulations on larger scale models would provide further insight to the claims made here. Furthermore, extending the study to multiple cycles can would be a good choice.

Appendix A

MATLAB Code for x-t Diagram Construction

Table of Contents

NOZZLE STATIC-PRESSURE X-T DIAGRAM 1
========== 1
========== 2
========== 3
========== 3
Read node coordinates 3
Read face-node connectivity 4
Read number of nodes per face 4
Read adjacent-cell connectivity 4
Determine number of cells 5
========== 5
========== 6
========== 6
Add contributions from face-cell side 0 7
Add contributions from face-cell side 1 7
========== 8
========== 9
========== 10
========== 11
========== 12
========== 13
========== 13
========== 14
========== 15
========== 16

NOZZLE STATIC-PRESSURE X-T DIAGRAM

Reads: FFF-Setup-Output.cas.h5 FFF-1-00001.dat.h5 FFF-1-00002.dat.h5 ...

Produces: nozzle_xt_results.mat nozzle_xt_diagram.png nozzle_xt_diagram.fig

The script: 1. Reads the Fluent mesh 2. Reconstructs cell centers 3. Detects the nozzle centerline 4. Reads static pressure from every time-step file 5. Creates an x-t diagram

```
clear;
clc;
close all;
```

==========

==========

USER SETTINGS

==========

```
dataFolder = ...
    "C:\Users\016785314\AppData\Local\Temp\WB_016785314_8452_2\" + ...
    "wbnew_files\dp0\FFF\Fluent";
```

```
caseFileName = "FFF-Setup-Output.cas.h5";

dataFilePattern = "FFF-1-*.dat.h5";

pressureDataset = "/results/1/phase-1/cells/SV_P/1";

% Pressure conversion:
%   1   = Pa
%   1e3 = kPa
%   1e6 = MPa
pressureDivisor = 1e3;
pressureUnit = "kPa";

% Leave empty for automatic color limits.
% Example:
% pressureColorLimits = [10 100];
pressureColorLimits = [];

numberOfContourLevels = 150;

saveOutputs = true;
```

CHECK FILES

```
caseFile = fullfile(dataFolder, caseFileName);

if ~isfolder(dataFolder)
    error("The specified folder does not exist:%s%s", ...
        newline, dataFolder);
end

if ~isfile(caseFile)
    error("The Fluent case file was not found:%s%s", ...
        newline, caseFile);
end

dataFiles = dir(fullfile(dataFolder, dataFilePattern));

if isempty(dataFiles)
    error("No data files matching %s were found in:%s%s", ...
        dataFilePattern, newline, dataFolder);
end

fprintf("Case file found:\n%s\n\n", caseFile);
fprintf("Number of transient data files found: %d\n\n", ...
    numel(dataFiles));
```

```
Error using make_nozzle_xt_diagram (line 62)
The specified folder does not exist:
C:\Users\016785314\AppData\Local\Temp\WB_016785314_8452_2\wbnew_files\dp0\FFF
\Fluent
```

SORT DATA FILES BY TIME-STEP NUMBER

```
nFiles = numel(dataFiles);
fileStepNumber = nan(nFiles,1);

for k = 1:nFiles

    token = regexp( ...
        dataFiles(k).name, ...
        '(\d+)\.dat\.h5$', ...
        'tokens', ...
        'once');

    if isempty(token)
        error("Could not identify the time-step number in file:%s%s", ...
            newline, dataFiles(k).name);
    end

    fileStepNumber(k) = str2double(token{1});
end

[fileStepNumber, fileOrder] = sort(fileStepNumber);
dataFiles = dataFiles(fileOrder);

fprintf("First data file: %s\n", dataFiles(1).name);
fprintf("Last data file:  %s\n\n", dataFiles(end).name);
```

READ MESH INFORMATION

```
fprintf("Reading mesh from Fluent case file...\n");
```

Read node coordinates

```
nodeCoordinates = h5read( ...
    caseFile, ...
    "/meshes/1/nodes/coords/4");
```

```
nodeCoordinates = double(nodeCoordinates);

% Fluent may store coordinates as:
%
%   2 x numberOfNodes
%
% instead of:
%
%   numberOfNodes x 2
%
% Convert to numberOfNodes x 2 if necessary.

if size(nodeCoordinates,1) == 2 && size(nodeCoordinates,2) > 2
    nodeCoordinates = nodeCoordinates.';
end

if size(nodeCoordinates,2) ~= 2
    error( ...
        "Unexpected node-coordinate dimensions: %d x %d", ...
        size(nodeCoordinates,1), ...
        size(nodeCoordinates,2));
end
```

Read face-node connectivity

```
faceNodeIDs = h5read( ...
    caseFile, ...
    "/meshes/1/faces/nodes/1/nodes");

faceNodeIDs = double(faceNodeIDs);

% Make sure the array is arranged as:
% numberOfFaces x 2

if size(faceNodeIDs,1) == 2 && size(faceNodeIDs,2) > 2
    faceNodeIDs = faceNodeIDs.';
end
```

Read number of nodes per face

```
nodesPerFace = h5read( ...
    caseFile, ...
    "/meshes/1/faces/nodes/1/nnodes");

nodesPerFace = double(nodesPerFace(:));
```

Read adjacent-cell connectivity

```
faceCell0 = h5read( ...
    caseFile, ...
    "/meshes/1/faces/c0/1");
```

```
faceCell0 = double(faceCell0(:));

faceCell1 = h5read( ...
    caseFile, ...
    "/meshes/1/faces/c1/1");

faceCell1 = double(faceCell1(:));
```

Determine number of cells

```
cellZoneMaximum = h5read( ...
    caseFile, ...
    "/meshes/1/cells/zoneTopology/maxId");

nCells = double(max(cellZoneMaximum(:)));

nNodes = size(nodeCoordinates,1);
nFaces = numel(nodesPerFace);

fprintf("Number of mesh nodes: %d\n", nNodes);
fprintf("Number of mesh faces: %d\n", nFaces);
fprintf("Number of mesh cells: %d\n\n", nCells);
```

==========

==========

VERIFY FACE CONNECTIVITY

==========

```
if any(nodesPerFace ~= 2)
    error([ ...
        "This script expects a two-dimensional mesh with exactly " ...
        "two nodes per face."]);
end

% Convert face-node array into numberOfFaces x 2 if it was read as a vector.

if isvector(faceNodeIDs)

    faceNodeIDs = faceNodeIDs(:);

    if numel(faceNodeIDs) ~= 2*nFaces
        error([ ...
            "Unexpected face-node array length. Expected %d values " ...
            "but found %d."], ...
            2*nFaces, numel(faceNodeIDs));
    end

    faceNodeIDs = reshape(faceNodeIDs, 2, []).';

elseif size(faceNodeIDs,1) ~= nFaces || size(faceNodeIDs,2) ~= 2
```

```
    error( ...
        "Unexpected face-node dimensions: %d x %d", ...
        size(faceNodeIDs,1), ...
        size(faceNodeIDs,2));
end

node1 = faceNodeIDs(:,1);
node2 = faceNodeIDs(:,2);

% Check node-ID range.

if any(node1 < 1) || any(node1 > nNodes) || ...
   any(node2 < 1) || any(node2 > nNodes)

    error([ ...
        "One or more face-node IDs fall outside the valid node range " ...
        "of 1 to %d."], nNodes);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

COMPUTE FACE MIDPOINTS

```
faceMidpointX = ...
    0.5 .* ( ...
    nodeCoordinates(node1,1) + ...
    nodeCoordinates(node2,1));

faceMidpointY = ...
    0.5 .* ( ...
    nodeCoordinates(node1,2) + ...
    nodeCoordinates(node2,2));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

RECONSTRUCT CELL CENTERS

```
fprintf("Reconstructing cell-center coordinates...\n");

% Some Fluent HDF5 files store cell IDs as zero-based indices.
% MATLAB requires one-based indices.
%
% Detect and correct this automatically.

if min(faceCell0) == 0
    faceCell0 = faceCell0 + 1;
end
```

```
if min(faceCell1) == 0
    faceCell1 = faceCell1 + 1;
end

% Verify cell-ID ranges.

if any(faceCell0 < 1) || any(faceCell0 > nCells)
    error("faceCell0 contains invalid cell IDs.");
end

if any(faceCell1 < 1) || any(faceCell1 > nCells)
    error("faceCell1 contains invalid cell IDs.");
end
```

Add contributions from face-cell side 0

```
cellXSum = accumarray( ...
    faceCell0, ...
    faceMidpointX, ...
    [nCells,1], ...
    @sum, ...
    0);

cellYSum = accumarray( ...
    faceCell0, ...
    faceMidpointY, ...
    [nCells,1], ...
    @sum, ...
    0);

cellFaceCount = accumarray( ...
    faceCell0, ...
    1, ...
    [nCells,1], ...
    @sum, ...
    0);
```

Add contributions from face-cell side 1

```
nInternalFaces = numel(faceCell1);

if nInternalFaces > nFaces
    error("The faceCell1 array is longer than the total face count.");
end

internalFaceX = faceMidpointX(1:nInternalFaces);
internalFaceY = faceMidpointY(1:nInternalFaces);

cellXSum = cellXSum + accumarray( ...
    faceCell1, ...
    internalFaceX, ...
    [nCells,1], ...
```

```
    @sum, ...
    0);

cellYSum = cellYSum + accumarray( ...
    faceCell1, ...
    internalFaceY, ...
    [nCells,1], ...
    @sum, ...
    0);

cellFaceCount = cellFaceCount + accumarray( ...
    faceCell1, ...
    1, ...
    [nCells,1], ...
    @sum, ...
    0);

if any(cellFaceCount == 0)
    error("One or more cells were not connected to any faces.");
end

cellCenterX = cellXSum ./ cellFaceCount;
cellCenterY = cellYSum ./ cellFaceCount;

fprintf("Cell centers reconstructed successfully.\n");
fprintf("Minimum faces per cell: %d\n", min(cellFaceCount));
fprintf("Maximum faces per cell: %d\n\n", max(cellFaceCount));
```

==========

==========

DETECT NOZZLE CENTERLINE

==========

```
meshYMinimum = min(cellCenterY);
meshYMaximum = max(cellCenterY);

estimatedCenterlineY = ...
    0.5 .* (meshYMinimum + meshYMaximum);

fprintf("Mesh minimum y: %.9g m\n", meshYMinimum);
fprintf("Mesh maximum y: %.9g m\n", meshYMaximum);
fprintf("Estimated centerline y: %.9g m\n\n", ...
    estimatedCenterlineY);

distanceFromCenterline = abs( ...
    cellCenterY - estimatedCenterlineY);

% Initial tight tolerance.

centerlineTolerance = 1e-5;
```

```
centerlineCellIDs = find( ...
    distanceFromCenterline <= centerlineTolerance);

% If too few cells are found, use the nearest row adaptively.

if numel(centerlineCellIDs) < 20

    sortedDistances = sort(distanceFromCenterline);

    initialCount = max(100, round(0.005*nCells));

    adaptiveTolerance = sortedDistances(initialCount);

    centerlineCellIDs = find( ...
        distanceFromCenterline <= adaptiveTolerance);

    centerlineTolerance = adaptiveTolerance;
end

centerlineX = cellCenterX(centerlineCellIDs);
centerlineY = cellCenterY(centerlineCellIDs);

[centerlineX, centerlineOrder] = sort(centerlineX);

centerlineCellIDs = centerlineCellIDs(centerlineOrder);
centerlineY = centerlineY(centerlineOrder);

fprintf("Centerline tolerance: %.9g m\n", centerlineTolerance);
fprintf("Number of selected centerline cells: %d\n", ...
    numel(centerlineCellIDs));
fprintf("Centerline x-range: %.9g to %.9g m\n\n", ...
    min(centerlineX), max(centerlineX));
```

==========

==========

CENTERLINE VERIFICATION PLOT

==========

```
figure( ...
    "Name", "Detected nozzle centerline", ...
    "Color", "w");

maximumBackgroundPoints = 40000;

if nCells > maximumBackgroundPoints

    backgroundIndex = round( ...
        linspace(1, nCells, maximumBackgroundPoints));

else
```

```
    backgroundIndex = 1:nCells;
end

scatter( ...
    cellCenterX(backgroundIndex), ...
    cellCenterY(backgroundIndex), ...
    3, ...
    ".");

hold on;

plot( ...
    centerlineX, ...
    centerlineY, ...
    "r-", ...
    "LineWidth", 1.5);

hold off;

xlabel("Axial coordinate, x [m]");
ylabel("Transverse coordinate, y [m]");
title("Automatically Detected Nozzle Centerline");

legend( ...
    "Mesh-cell sample", ...
    "Selected centerline", ...
    "Location", "best");

axis equal;
grid on;
box on;
```

==========

==========

READ STATIC PRESSURE FROM ALL FILES

==========

```
nCenterlineCells = numel(centerlineCellIDs);

pressureXT = nan(nFiles, nCenterlineCells);
flowTime = nan(nFiles,1);
fluentTimeStep = nan(nFiles,1);

fprintf("Reading transient pressure files...\n\n");

for k = 1:nFiles

    currentFile = fullfile( ...
        dataFiles(k).folder, ...
        dataFiles(k).name);
```

```
    fprintf("Reading file %d of %d: %s\n", ...
        k, nFiles, dataFiles(k).name);

    pressureAllCells = h5read( ...
        currentFile, ...
        pressureDataset);

    pressureAllCells = double(pressureAllCells(:));

    if numel(pressureAllCells) ~= nCells
        error([ ...
            "Pressure-array size mismatch in file %s.\n" ...
            "Expected %d values but found %d."], ...
            dataFiles(k).name, ...
            nCells, ...
            numel(pressureAllCells));
    end

    pressureXT(k,:) = ...
        pressureAllCells(centerlineCellIDs).' ./ pressureDivisor;

    [flowTime(k), fluentTimeStep(k)] = ...
        readFluentTimeInformation(currentFile);
end

fprintf("\nAll pressure files were read successfully.\n\n");
```

==========

==========

HANDLE TIME VALUES

==========

```
if any(isnan(flowTime))

    warning([ ...
        "Flow time could not be read from one or more files. " ...
        "File-step number will be used instead."]);

    timeCoordinate = fileStepNumber;
    timeAxisLabel = "Time-step number";

else

    timeCoordinate = flowTime;
    timeAxisLabel = "Flow time, t [s]";
end

[timeCoordinate, timeOrder] = sort(timeCoordinate);

pressureXT = pressureXT(timeOrder,:);
flowTime = flowTime(timeOrder);
```

```
fluentTimeStep = fluentTimeStep(timeOrder);
fileStepNumber = fileStepNumber(timeOrder);
dataFiles = dataFiles(timeOrder);
```

==========

==========

COMBINE DUPLICATE X LOCATIONS

==========

```
roundingTolerance = 1e-10;

roundedX = ...
    round(centerlineX ./ roundingTolerance) .* roundingTolerance;

[uniqueX, ~, xGroup] = unique(roundedX, "stable");

if numel(uniqueX) < numel(centerlineX)

    fprintf( ...
        "Combining %d cells into %d unique x-locations.\n\n", ...
        numel(centerlineX), numel(uniqueX));

    averagedPressureXT = nan(nFiles, numel(uniqueX));
    averagedCenterlineY = nan(numel(uniqueX),1);
    representativeCellIDs = nan(numel(uniqueX),1);

    for j = 1:numel(uniqueX)

        members = (xGroup == j);

        averagedPressureXT(:,j) = ...
            mean(pressureXT(:,members), 2, "omitnan");

        averagedCenterlineY(j) = ...
            mean(centerlineY(members), "omitnan");

        firstMember = find(members, 1, "first");

        representativeCellIDs(j) = ...
            centerlineCellIDs(firstMember);
    end

    centerlineX = uniqueX;
    centerlineY = averagedCenterlineY;
    centerlineCellIDs = representativeCellIDs;
    pressureXT = averagedPressureXT;
end
```

CREATE X-T DIAGRAM WITH IMAGESC

```
figure( ...
    "Name", "Static-pressure x-t diagram", ...
    "Color", "w");

imagesc( ...
    centerlineX, ...
    timeCoordinate, ...
    pressureXT);

set(gca, "YDir", "normal");

xlabel("Axial position, x [m]");
ylabel(timeAxisLabel);
title("Nozzle Centerline Static-Pressure x-t Diagram");

colorbarHandle = colorbar;

colorbarHandle.Label.String = ...
    "Static pressure, p [" + pressureUnit + "]";

colormap(turbo);

if ~isempty(pressureColorLimits)
    clim(pressureColorLimits);
end

axis tight;
box on;
```

CREATE CONTOUR X-T DIAGRAM

```
figureXT = figure( ...
    "Name", "Static-pressure x-t contours", ...
    "Color", "w");

contourf( ...
    centerlineX, ...
    timeCoordinate, ...
    pressureXT, ...
    numberOfContourLevels, ...
    "LineColor", "none");
```

```
xlabel("Axial position, x [m]");
ylabel(timeAxisLabel);
title("Nozzle Centerline Static-Pressure x-t Diagram");

colorbarHandle = colorbar;

colorbarHandle.Label.String = ...
    "Static pressure, p [" + pressureUnit + "]";

colormap(turbo);

if ~isempty(pressureColorLimits)
    clim(pressureColorLimits);
end

axis tight;
box on;
```

PLOT SELECTED PRESSURE PROFILES

```
firstIndex = 1;
middleIndex = round(nFiles/2);
lastIndex = nFiles;

figure( ...
    "Name", "Selected pressure profiles", ...
    "Color", "w");

plot( ...
    centerlineX, ...
    pressureXT(firstIndex,:), ...
    "LineWidth", 1.3);

hold on;

plot( ...
    centerlineX, ...
    pressureXT(middleIndex,:), ...
    "LineWidth", 1.3);

plot( ...
    centerlineX, ...
    pressureXT(lastIndex,:), ...
    "LineWidth", 1.3);

hold off;
```

```
xlabel("Axial position, x [m]");

ylabel( ...
    "Static pressure, p [" + pressureUnit + "]");

title("Selected Centerline Pressure Profiles");

legend( ...
    createTimeLabel( ...
    timeCoordinate(firstIndex), timeAxisLabel), ...
    createTimeLabel( ...
    timeCoordinate(middleIndex), timeAxisLabel), ...
    createTimeLabel( ...
    timeCoordinate(lastIndex), timeAxisLabel), ...
    "Location", "best");

grid on;
box on;
```

SAVE OUTPUTS

```
if saveOutputs

    matlabOutputFile = fullfile( ...
        dataFolder, ...
        "nozzle_xt_results.mat");

    imageOutputFile = fullfile( ...
        dataFolder, ...
        "nozzle_xt_diagram.png");

    figureOutputFile = fullfile( ...
        dataFolder, ...
        "nozzle_xt_diagram.fig");

    save( ...
        matlabOutputFile, ...
        "centerlineX", ...
        "centerlineY", ...
        "centerlineCellIDs", ...
        "estimatedCenterlineY", ...
        "centerlineTolerance", ...
        "timeCoordinate", ...
        "flowTime", ...
        "fluentTimeStep", ...
        "fileStepNumber", ...
        "pressureXT", ...
        "pressureUnit", ...
```

```
        "pressureDivisor", ...
        "-v7.3");

    exportgraphics( ...
        figureXT, ...
        imageOutputFile, ...
        "Resolution", 300);

    savefig( ...
        figureXT, ...
        figureOutputFile);

    fprintf("Results saved successfully:\n");
    fprintf("%s\n", matlabOutputFile);
    fprintf("%s\n", imageOutputFile);
    fprintf("%s\n\n", figureOutputFile);
end

fprintf("X-t diagram generation complete.\n");
```

LOCAL FUNCTIONS

```
function [flowTime, timeStep] = ...
    readFluentTimeInformation(h5File)

    flowTime = NaN;
    timeStep = NaN;

    try

        rawMetadata = h5read( ...
            h5File, ...
            "/settings/Data Variables");

        metadataText = ...
            convertHDF5TextToString(rawMetadata);

        flowTimeToken = regexp( ...
            metadataText, ...
            '\(flow-time\s+([-+0-9.eEdD]+)\)', ...
            'tokens', ...
            'once');

        if ~isempty(flowTimeToken)

            numberText = string(flowTimeToken{1});

            numberText = replace( ...
```

```
                numberText, ...
                ["D","d"], ...
                ["E","e"]);

            flowTime = str2double(numberText);
        end

        timeStepToken = regexp( ...
            metadataText, ...
            '\(time-step\s+([-+0-9.eEdD]+)\)', ...
            'tokens', ...
            'once');

        if ~isempty(timeStepToken)

            numberText = string(timeStepToken{1});

            numberText = replace( ...
                numberText, ...
                ["D","d"], ...
                ["E","e"]);

            timeStep = str2double(numberText);
        end

    catch ME

        warning( ...
            "Could not read Fluent time metadata from %s:\n%s", ...
            h5File, ...
            ME.message);
    end
end

function textOutput = ...
    convertHDF5TextToString(rawInput)

    if iscell(rawInput)
        rawInput = rawInput{1};
    end

    if isstring(rawInput)

        textOutput = rawInput(1);

    elseif ischar(rawInput)

        textOutput = string(rawInput);

    elseif isnumeric(rawInput)

        textOutput = string(char(rawInput(:).'));

    else
```

```
        try
            textOutput = string(char(rawInput(:).'));
        catch
            textOutput = string(rawInput);
        end
    end
end

function labelText = ...
    createTimeLabel(timeValue, timeAxisLabel)

    if contains(timeAxisLabel, "[s]")

        labelText = sprintf( ...
            "t = %.6g s", ...
            timeValue);

    else

        labelText = sprintf( ...
            "Step = %.0f", ...
            timeValue);
    end
end
```

Published with MATLAB® R2026a

References

[1] Hinkey, J. B., et al., "Pulse Detonation Rocket Engine Research at NASA Marshall," *NASA Technical Report*, 2019.

[2] Heiser, W. H., and Pratt, D. T., *Hypersonic Airbreathing Propulsion*, AIAA, 1994.

[3] Kailasanath, K., "Review of Propulsion Applications of Detonation Waves," *AIAA Journal*, 2000.

[4] Pandey, K. M., et al., "Review on Recent Advances in Pulse Detonation Engines," *Journal of Combustion*, 2016.

[5] Roy, G., et al., "Pulse Detonation Propulsion: Challenges and Opportunities," *Progress in Aerospace Sciences*, 2004.

[6] Schauer, F., et al., "Numerical Rocket Modeling of Pulse Detonation Engine Gasdynamics," *NASA Technical Report*, 2020.

[7] Mital, S., et al., "Propulsion Tests on Ultra-High-Temperature Ceramic Matrix Composites," *Journal of the American Ceramic Society*, 2022.

[8] NASA, "Ultra-High-Temperature Ceramic Matrix Composites for Rocket Applications," *NASA TM*, 2023.

[9] Schauer, F., et al., "Pressure and Thrust Measurements of a High-Frequency Pulsed Detonation Tube," *Journal of Propulsion and Power*, 2018.

[10] Li, J., et al., "Frequency Scaling in Rocket-Type Pulse Detonation Engines," *Aerospace Science and Technology*, 2022.

[11] Frolov, S., et al., "Parametric Studies of PDRE Operation," *Combustion Science and Technology*, 2021.

[12] Zhang, Y., et al., "Recent Developments in Pulse Detonation Rocket Engines," *Acta Astronautica*, 2023.

[13] Kim, H., et al., "Multi-Objective Optimization of Rocket-Type Pulse Detonation Engine Nozzles," *Aerospace*, 2021.

[14] ANSYS Innovation Space, "Basics of Turbulent Flows – Lesson 5: Closure of RANS Equations," Handout, ANSYS Innovation Space, September 2020. URL: https://innovationspace.ansys.com/courses/wp-content/uploads/2020/09/Basics-of-Turbulent-Flows-Lesson-5-Handout.pdf.

[15] Boussinesq, J., *Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes à grande section*, Gauthier-Villars et fils, Paris, France, 1897. URL: https://archive.org/details/thbeoriedelbeco01bousrich.

[16] Prandtl, L., "Bericht über Untersuchungen zur ausgebildeten Turbulenz," *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, Vol. 5, No. 2, 1925, pp. 136–139. https://doi.org/10.1002/zamm.19250050212.

[17] Launder, B. E., and Spalding, D. B., *Lectures in Mathematical Models of Turbulence*, Academic Press, London and New York, 1972.

[18] Wilcox, D. C., "Formulation of the k–ω Turbulence Model Revisited," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2823–2838. https://doi.org/10.2514/1.36541.

[19] Menter, F. R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605. https://doi.org/10.2514/3.12149.

[20] Hanjalić, K., and Launder, B. E., *Modelling Turbulence in Engineering and the Environment: Second-Moment Routes to Closure*, Cambridge University Press, Cambridge, UK, 2011.

[21] Nagarkar, B., *Computational Analysis of Pulse Detonation Engine: Effects of Converging and Diverging Tube Geometries*, Master's Thesis, Department of Aerospace Engineering, San José State University, San José, CA, 2018.

[22] Bermudez De La Hoz, J., Rudy, W., and Teodorczyk, A., *Numerical Simulation of the Transition to Detonation in a Hydrogen–Air Mixture Due to Shock Wave Focusing on a 90-Degree Wedge*, Vol. 18, No. 3, 2025, Article 619. https://doi.org/10.3390/en18030619

[23] Morris, C. I., *Numerical Modeling of Pulse Detonation Rocket Engine Gasdynamics and Performance*, Journal of Propulsion and Power.