Computational Fluid Dynamics for Sonic Black Hole Research

Zachary Berkson-Korenberg

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Science

Department of Physics
McGill University
Montréal, Quebec, Canada
December 2024

Abstract

This thesis presents a comprehensive study of the theoretical and computational modeling of fluid dynamics in the context of sonic black holes, with a focus on the formation of acoustic horizons in nanoscale converging-diverging nozzles. A custom Python simulation framework built around computational fluid dynamics library OpenFOAM was developed to replicate laboratory conditions and explore the formation of acoustic horizons. The framework was rigorously tested against known analytical solutions, including Poiseuille flow, ensuring accuracy and validity.

Simulations of helium-4 flow at 77 K were performed with a variety of nozzle geometries, including those used in previous high-pressure silicon nitride nanopore experiments. Remarkably, agreement with experimental geometries was found by matching only the measured mass flow profiles with those simulated, achieved without requiring specific assumptions about nozzle geometry. These findings confirmed that Mach 1 was reached under laboratory conditions—suggesting the formation of a sonic black hole—and enabled precise calculations of Unruh-Hawking radiation emission rates. The work further investigated the influence of nozzle geometry by employing a simplified model, targeting specific geometric parameters such as inner radius and opening angle, and found good mass flow agreement with substantial variability in emission rates, showing the necessity of complete geometric profiles for accurately predicting phonon emissions. The study explores the critical role of nozzle geometry in fluid behaviour, offering tools to estimate experimental outcomes with sufficient accuracy, even in regimes where full geometric data is unavailable.

Abrégé

Cette thèse examine la modélisation théorique et numérique de la dynamique des fluides liée aux trous noirs soniques en se concentrant sur la formation d'horizons acoustiques dans des buses convergentes-divergentes nanométriques. Un cadre de simulation Python, basé sur OpenFOAM, a été conçu pour reproduire les conditions expérimentales et étudier ces phénomènes. Validé avec des solutions analytiques comme l'écoulement de Poiseuille, il garantit précision et fiabilité.

Des simulations de l'écoulement d'hélium-4 à 77 K ont été réalisées sur une variété de géométries de buses, y compris celles utilisées dans des expériences antérieures avec des nanopores en nitrure de silicium sous haute pression. De manière remarquable, une concordance avec les géométries expérimentales a été obtenue en ajustant les profils de débit massique mesurés à ceux simulés, et ce, sans nécessiter d'hypothèses spécifiques sur la géométrie des buses. Ces résultats ont confirmé la formation d'un trou noir sonique dans des conditions de laboratoire et ont permis des calculs précis des taux d'émission de rayonnement d'Unruh-Hawking. Le travail a également exploré l'influence de la géométrie des buses à l'aide d'un modèle simplifié, en se concentrant sur des paramètres géométriques spécifiques tels que le rayon intérieur et l'angle d'ouverture. Une forte concordance des débits massiques a été observée, accompagnée d'une grande variabilité dans les taux d'émission, démontrant ainsi la nécessité de profils géométriques complets pour prédire avec précision les émissions de phonons.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor, Professor Guillaume Gervais, for his support and guidance throughout this project. Our countless discussions were invaluable, and his passion for science has been a constant source of inspiration. I am excited to continue working under his mentorship during my PhD.

This project would not have been possible without the foundational work of Matei Petrescu, who generously assisted with simulations and calculations. The experimental groundwork laid by Sam Neale nearly a decade ago formed the backbone of this research. His meticulous documentation allowed me to build upon his findings with confidence. I am also deeply indebted to Alex Maloney, whose teaching of general relativity profoundly shaped my understanding and equipped me to tackle challenging concepts in this field.

I also wish to thank Nick Wicklund, Dawson Willerton, and Renée Goodman, who have continued Sam Neale's work, for their assistance in understanding the experimental aspects of this project and for their help with calculations. To my lab colleagues—Frederik Poitevin, Sujatha Vijayakrishnan, Frédéric Boivin, Oulin Yu, and Talia Martz-Oberlander—thank you for the moral support and stimulating discussions which made the challenging moments manageable.

Finally, I am deeply grateful to my friends, my family, and my fiancée, Laurence Charlebois, for their unwavering support and dedication. Their encouragement has been a pillar of strength throughout this journey.

Dedication

This thesis is dedicated to my mother, Joy Berkson, for her lifelong love, encouragement, and belief in me. I would not have reached this milestone without her, and I am forever grateful.

Contents

List of Abbreviations

CD Converging-Diverging.

CFD Computational Fluid Dynamics.

CV Control Volume.

FDM Finite Difference Method.

FOAM Field Operation and Manipulation.

FVM Finite Volume Method.

GNU GNU's Not Unix!.

GPL General Public License.

NS Navier-Stokes.

ODE Ordinary Differential Equation.

OF OpenFOAM.

PDE Partial Differential Equation.

PISO Pressure-Implicit with Splitting of Operators.

SIMPLE Semi-Implicit Method for Pressure-Linked Equations.

TEM Transmission Electron Microscope.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Gravity, a force that has shaped human understanding of the universe for centuries, remains a profound mystery at the intersection of general relativity (GR) and quantum mechanics. While Albert Einstein's general theory of relativity provides an elegant framework for understanding gravity on large scales, it remains incompatible with the principles of quantum mechanics, which governs physics at the subatomic scale. This incongruity is most evident in the study of black holes — astrophysical phenomena predicted by GR where spacetime curvature becomes so strong that an event horizon forms where no matter or electromagnetic energy can escape, leading to singularities where quantum effects are thought to dominate. The singularity, characterized by a finite mass with zero volume and thus infinite density, is commonly perceived as an incompleteness of Einstein's theory [1]. This conflict is a central challenge in modern physics, with the hope that a widely-sought unified theory of quantum gravity will somehow resolve it.

Theoretical research into astrophysical black holes has been remarkably fruitful, transforming them from mathematical oddities—thought impossible by even Einstein—to a central pillar of modern astrophysics. Today, black holes captivate not only scientists but also the public imagination, and is an iconic symbol of cosmic mystery and the frontier of human understanding. Their allure far surpasses academia, influencing popular culture through books, movies, and television. Films like *The Black Hole* and *Interstellar* dramatize their extreme physics and visual majesty, while documentary series like *Cosmos* bring their complex science to broader audiences. The image of a black hole, captured for the first time by the Event Horizon Telescope in 2019, became an international sensation. Black holes are the ultimate enigma: unfathomably massive objects with which the widely accepted laws of physics break down.

Astrophysical black holes, however, present a daunting experimental challenge. Their immense distances and extreme conditions make direct observations of quantum effects impossible with current or even foreseeable future technologies. To address this, researchers have turned to analogue systems on Earth, which replicate key features of astrophysical black holes in accessible laboratory settings. Among these analogues, the acoustic black hole—where fluid flow surpasses the local speed of sound, creating an event horizon for phonons—offers a unique opportunity to explore theoretically predicted phenomena, such as analogous Hawking radiation. These systems allow for controlled studies of curved spacetime, pioneering the experimental (in)validation of previously un(dis)provable theoretical predictions.

1.2 Black Hole in a Lab

A common approach for sonic black hole formation that is central to this work is the use of converging-diverging nozzles, also known as *de Laval* nozzles. As will be shown in the following section, these nozzles are capable of accelerating fluid flow to supersonic speeds, with the acoustic horizon forming at the throat of the nozzle—the point of smallest cross-sectional area between the converging and diverging sections. While the converging-diverging nozzle is not the only method for creating a sonic black hole in a laboratory setting, it remains a particularly effective medium and is the primary focus of this study.

1.2.1 Experimental Shortcomings

Designing an experiment to detect phonon emissions from such an acoustic horizon presents several challenges, particularly regarding the precise characterization of fluid flow, as we will see that the emission rate of phonons is highly sensitive to spatial velocity variations. Two primary challenges arise from this relationship. Firstly, phonons traveling through a fluid medium experience Doppler shifts given by[1]

$$\omega_{obs} = \frac{1}{2}\left(1 - \frac{u}{c_s}\right)\omega_{emitted}, \tag{1.1}$$

where u is fluid velocity at the point of detection, c_s is the local speed of sound, $\omega_{emitted}$ and ω_{obs} are the emitted and observed frequencies, respectively.

Resolving this Doppler shift requires detailed knowledge of the velocity profile at multiple points within the nozzle. However, simply knowing the geometric profile of the nozzle is insufficient, as the vast majority of solutions to the governing fluid equations are non-analytical [2]. Experimental techniques typically lack the spatial resolution required to directly measure velocity gradients and other critical flow characteristics. Consequently, computational modeling is required to complement experimental design by predicting velocity fields at detection points or even identifying regions where phonon emissions are strongest. Detection methods for emitted phonons include advanced technologies such as coupled piezoelectric detectors sensitive to acoustic signals, and laser-based systems capable of measuring density

---

[1] The Doppler effect is given by $\omega = \omega_0\left(\frac{c_s+v_d}{c_s-v_s}\right)$, where v_d and v_s are the velocities of the detector and source relative to the medium. If we define the medium velocity as the fluid velocity u, then a stationary detector has relative velocity $v_d = -u$ and a stationary source at the nozzle throat has relative velocity $v_s = -c_s$. Substituting these values yields equation (1.1) which is defined for $u \leq c_s$.

variations in fluid [3]. While all accelerating bodies radiate[2], a phonon emitted downstream cannot classically propagate back through the nozzle. Thus, these systems could, in principle, detect emitted phonons at both the inlet and outlet of the nozzle, providing a complete description of phonon emission. Isolating phonon emission spectra from background noise would require some sort of Fourier analysis and must account for Doppler shifts at both detection points, which are only classically defined for subsonic flow.

Secondly, we will see that the phonon emission rate is related to the spatial gradient of velocity, *via* the emission spectrum, meaning stronger accelerations over shorter distances result in significantly higher emission rates. For now, we will assume that the blackbody emission spectrum has an inverse dependence on length without further derivation. Therefore, a nozzle with radius $R_{in} = 100$ nm would exhibit a phonon temperature that is 10^5 times larger than that of a $R_{in} = 1$ cm nozzle. The emission rates for the larger nozzle, however, are only just within the lower detection limits of current technologies [3].

Previous experimental investigations conducted by our group focused on silicon nitride nanonozzles, where high-pressure flow measurements with helium-4 gas at 77 K were performed [4]. These experiments aimed to explore the viability of sonic black hole formation in the geometry produced by Transmission Electron Microscope (TEM) drilled nanopores. Although definitive evidence of a sonic black hole could not be obtained, the team made preliminary estimates of the expected phonon emission rates under these conditions. The work provided insights into the potential detectability of acoustic radiation but also highlighted the limitations of purely experimental approaches in resolving such phenomena.

---

[2] While this statement may not be universally true in classical mechanics, the phenomenon is a key prediction of quantum field theory, known as the *Unruh Effect*. Similarly, classical field theory predicts radiation from an accelerating *charged* particle, as described by the *Larmor formula*.

1.3 Research Objectives

This work builds on a project initiated by Matei Petrescu [5] which explored the viability of computational methods to enhance the understanding of experimental flow measurements, such as those conducted previously by Sam Neale [4] on helium gas flow in silicon nitride nanonozzles. Petrescu's computational efforts focused on simulating helium gas flow in the micron range and validating isentropic relations in the mesoscopic regime. In contrast, Neale's experimental work investigated high-pressure helium gas flow at the nanometer scale, providing foundational experimental data for studying transonic nanoscale flow systems.

The primary goal is to advance the study of sonic black hole research by providing a robust computational framework capable of accurately replicating laboratory conditions and performing detailed analyses of acoustic horizon formation and the associated phonon emissions in converging-diverging nozzles. While previous computational efforts primarily addressed flow dynamics in the mesoscopic regime, this work focuses on nanoscale geometries where the steep velocity gradients necessary for phonon detection are most pronounced. This framework seeks to bridge the gap between theoretical predictions and experimental observations, overcoming key limitations of earlier approaches by carefully engineering simulations to resolve missing experimental details. Ultimately, this effort provides a powerful tool for refining experimental designs and advancing our understanding of sonic black holes in nanoscale converging-diverging nozzle systems.

Chapter 2 lays the mathematical groundwork necessary for understanding fluid dynamics and the principles of sonic black holes in converging-diverging nozzles. Chapter 3 introduces fundamental concepts of Computational Fluid Dynamics (CFD). In Chapter 4, the framework employed in this study is described in detail, along with its application to replicate laboratory experiments involving high-speed helium gas through silicon nitride nanonozzles. Finally, Chapter 5 presents the results of these simulations, comparing them to previous experimental measurements and exploring the insights they provide into the formation of acoustic horizons and associated phonon emissions.

Chapter 2

Background

2.1 Fluid Dynamics

Fluid dynamics is a branch of physics that studies the behaviour of liquids or gases, and is governed by sets of partial differential equations describing specific non-equilibrium conditions based on Newton's second law.

2.1.1 Euler's Equations

Euler's equations are one of the fundamental equation sets of fluid dynamics governing adiabatic and inviscid flow, which connect velocity, pressure, and density as functions of space and time. Euler's equations are given by [2]

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

$$\frac{D\vec{u}}{Dt} \equiv \frac{\partial \vec{u}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla}\right) \vec{u} = -\frac{1}{\rho} \vec{\nabla} P, \tag{2.2}$$

$$\frac{De}{Dt} \equiv \frac{\partial e}{\partial t} + \vec{u} \cdot \vec{\nabla} e = -\frac{P}{\rho} \vec{\nabla} \cdot \vec{u}, \tag{2.3}$$

where $\vec{u}$ is velocity, ρ is fluid density, P is pressure, e is specific internal energy, and where the material derivative operator is defined as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{u} \cdot \vec{\nabla}. \tag{2.4}$$

Equation (2.2) is often referred to as the momentum conservation equation, but it is expressed in terms of velocity. Consider the following equation in tensor[1] notation,

$$\frac{\partial}{\partial t}(\rho u_i) = \rho \frac{\partial u_i}{\partial t} + \frac{\partial \rho}{\partial t} u_i. \tag{2.5}$$

With the same notation, we can write (2.1, 2.2) in the form

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\frac{\partial (\rho u_j)}{\partial x_j}, && (2.6) \\ \frac{\partial u_i}{\partial t} &= -u_j \frac{\partial u_i}{\partial x_j} - \frac{1}{\rho}\frac{\partial P}{\partial x_i}. && (2.7) \end{aligned}$$

Substituting into (2.5), we obtain

$$\begin{aligned} \frac{\partial}{\partial t}(\rho u_i) &= -\rho u_j \frac{\partial u_i}{\partial u_j} - \frac{\partial P}{\partial x_i} - u_i \frac{\partial (\rho u_j)}{\partial x_j}, && (2.8) \\ &= -\frac{\partial P}{\partial x_i} - \frac{\partial}{\partial x_j}(\rho u_i u_j), && (2.9) \\ &= -\frac{\partial \Pi_{ij}}{\partial x_j}, && (2.10) \end{aligned}$$

where Π_{ij} is known as the *momentum flux density* tensor, given by

$$\Pi_{ij} \equiv P\delta_{ij} + \rho u_i u_j. \tag{2.11}$$

The derivation is immediately obvious if you consider $\frac{\partial P}{\partial x_i} = \delta_{ij}\frac{\partial P}{\partial x_j}$, where δ_{ij} is the Kronecker delta tensor. This gives the conservation form of (2.1, 2.2),

---

[1] A *tensor* is an object that transforms like a tensor.

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}) &= 0, && (2.12)\\
\frac{\partial (\rho \vec{u})}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u} \otimes \vec{u}) &= -\vec{\nabla} P, && (2.13)
\end{aligned}
$$

where $\otimes$ is the outer product given by $(\vec{u} \otimes \vec{u})_{ij} = u_i u_j$, producing a rank-2 tensor. For an incompressible fluid, where density ρ is constant with time and is spatially uniform, (2.1, 2.2) and (2.12, 2.13) are written as

$$
\begin{aligned}
\vec{\nabla} \cdot \vec{u} &= 0, && (2.14)\\
\frac{D\vec{u}}{Dt} &= -\vec{\nabla} p_k, && (2.15)
\end{aligned}
$$

where kinematic pressure is defined as $p_k \equiv P/\rho$.

2.1.2 Navier-Stokes

2.1.2.1 Construction of the Navier-Stokes Equations

The Navier-Stokes equations are a set of partial differential equations which describe the motion of viscous fluids and are derived from Euler's equations. To derive the Navier-Stokes equations, the effects of energy dissipation become important. The momentum flux density tensor given by (2.11) represents a completely reversible transfer of momentum due to simple mechanical transport of particles and pressure forces acting on a fluid. Viscosity, which is an internal friction, causes irreversible momentum transfer from points where velocity is large to those where velocity is small. To consolidate friction, we add a term which accounts for the irreversible viscous transfer of momentum, σ'_{ij}, *i.e.* the *viscous* stress tensor

$$\Pi_{ij} = P\delta_{ij} + \rho u_i u_j - \sigma'_{ij} = -\sigma_{ij} + \rho u_i u_j, \qquad (2.16)$$

$$\sigma_{ij} = -P\delta_{ij} + \sigma'_{ij}, \qquad (2.17)$$

where σ_{ij} is known as the stress tensor. We can then write Euler's momentum equation in conservative form (2.13) as

$$\frac{\partial}{\partial t}\left(\rho\vec{u}\right) + \vec{\nabla}\cdot\left(\rho\vec{u}\otimes\vec{u}\right) = \vec{\nabla}\cdot\overset{\leftrightarrow}{\boldsymbol{\sigma}}. \tag{2.18}$$

For small velocity gradients, we may suppose that momentum transfer due to viscosity depends only on the first order velocity derivatives[2], *i.e.* $\partial u_i/\partial x_j$. Furthermore, we can assume that σ'_{ij} is a linear combination of these velocity gradients. Since σ'_{ij} must vanish for constant velocity, we may assume that there are no terms present in the tensor that are independent of velocity derivatives. Similarly, σ_{ij} must vanish for uniform rotations with angular velocity $\vec{\Omega}$, as no internal friction can occur for this type of motion [2]. If we consider $\vec{u} = \vec{\Omega}\times\vec{r}$, then the sums

$$\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \tag{2.19}$$

are linear combinations of first order velocity derivatives and drop out when $\vec{u} = \vec{\Omega}\times\vec{r}$ is constant. Therefore, σ'_{ij} must contain only these symmetric combinations of $\partial u_i/\partial x_j$. The most general rank-2 tensor that satisfies the above conditions is [2]

$$\sigma'_{ij} = \eta\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_l}{\partial x_l}\right) + \zeta\delta_{ij}\frac{\partial u_l}{\partial x_l}, \tag{2.20}$$

with coefficients η and ζ independent of velocity. Einstein summation convention is used with summation index l. For isotropic fluids, these coefficients are positive scalar values, called the *coefficients of viscosity*, where η is the *shear* viscosity and ζ has many names, such as the *second*, *bulk* or even *odd* viscosity. η and ζ are dynamic viscosities with SI units $\mathrm{Pa}\cdot\mathrm{s}$. In vector form, (2.20) is written as

$$\overset{\leftrightarrow}{\boldsymbol{\sigma}}' = \eta\left(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T\right) + \left(\zeta - \frac{2}{3}\eta\right)\left(\vec{\nabla}\cdot\vec{u}\right)\overset{\leftrightarrow}{\boldsymbol{I}}, \tag{2.21}$$

---

[2] This approximation is valid when the stress in a fluid can be expressed as a local function of the velocity gradient, which holds for gases in the continuum limit. However, in flows with small characteristic length scales relative to the molecular mean-free path, large velocity gradients, or non-Newtonian fluids, higher-order corrections may be required.

and the stress tensor (2.17) becomes

$$\begin{aligned}
\overleftrightarrow{\boldsymbol{\sigma}} &= -P\overleftrightarrow{\boldsymbol{I}} + \eta\left(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T\right) + \left(\zeta - \frac{2}{3}\eta\right)\left(\vec{\nabla}\cdot\vec{u}\right)\overleftrightarrow{\boldsymbol{I}}, && (2.22)\\
&= \left(P - \left(\zeta - \frac{2}{3}\eta\right)\left(\vec{\nabla}\cdot\vec{u}\right)\right)\overleftrightarrow{\boldsymbol{I}} + \eta\left(\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T\right), && (2.23)
\end{aligned}$$

which is true for Newtonian fluids [2]. Combining (2.12), (2.18), and (2.23) with an external body force[3] $\vec{f}$, we arrive at the Navier-Stokes equations that describe viscous fluids:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla}\cdot(\rho\vec{u}) = 0, \tag{2.24}$$

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\vec{\nabla})\vec{u}\right) = -\vec{\nabla}P + \vec{\nabla}\cdot\left\{\eta\left[\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T - \frac{2}{3}\left(\vec{\nabla}\cdot\vec{u}\right)\overleftrightarrow{\boldsymbol{I}}\right]\right\} + \vec{\nabla}\left[\zeta\left(\vec{\nabla}\cdot\vec{u}\right)\right] + \rho\vec{f}. \tag{2.25}$$

In conservative form, (2.25) is written as

$$\frac{\partial}{\partial t}(\rho\vec{u}) + \vec{\nabla}\cdot\left(\rho\vec{u}\otimes\vec{u} + \left[P - \zeta\left(\vec{\nabla}\cdot\vec{u}\right)\right]\overleftrightarrow{\boldsymbol{I}} - \eta\left[\vec{\nabla}\vec{u} + \vec{\nabla}\vec{u}^T - \frac{2}{3}\left(\vec{\nabla}\cdot\vec{u}\right)\overleftrightarrow{\boldsymbol{I}}\right]\right) = \rho\vec{f}. \tag{2.26}$$

Defining *shear kinematic* viscosity $\nu \equiv \eta/\rho$ and *bulk kinematic* viscosity $\xi \equiv \zeta/\rho$ with units m^2/s, and assuming that both viscosities are uniform, (2.25) can be written as

$$\frac{D\vec{u}}{Dt} = \frac{\partial \vec{u}}{\partial t} + \left(\vec{u}\cdot\vec{\nabla}\right)\vec{u} = -\frac{1}{\rho}\vec{\nabla}P + \nu\vec{\nabla}^2\vec{u} + \left(\frac{1}{3}\nu + \xi\right)\vec{\nabla}\left(\vec{\nabla}\cdot\vec{u}\right) + \vec{f}, \tag{2.27}$$

which is the simplest form of the compressible Navier-Stokes momentum equation that can be written. For isothermal fluids, low Mach number flows, ideal gases with constant properties, and laminar flow, the assumption of uniform kinematic viscosities provide a reasonable approximation to make the mathematical treatment of the Navier-Stokes equations more tractable.

---

[3] Most often just gravity for neutral fluids, with SI units $\mathrm{N/kg} = \mathrm{m/s}^2$

2.1.2.2 Reynolds Number

The Reynolds number is a dimensionless quantity used most often to predict the turbulent regime of a well described fluid, and is defined as [6]

$$\mathrm{Re} \equiv \frac{\text{Inertial force}}{\text{Viscous force}} \propto \frac{\rho|\vec{u}|\partial_x|\vec{u}|}{\eta\partial_x^2|\vec{u}|} = \frac{\rho|\vec{u}|D}{\eta} = \frac{|\vec{u}|D}{\nu}, \tag{2.28}$$

where D is the characteristic length of the fluid, typically the smallest of constriction length and diameter. Viscous friction is negligible when $\mathrm{Re} \ll 1$. When $\mathrm{Re} < 2000$, fluid flow is called *laminar*. Flow will always be turbulent when $\mathrm{Re} \gtrsim 4000$. Between $2000 < \mathrm{Re} < 4000$ flow is unstable, having characteristics of both laminar and turbulent regimes, where small disturbances can grow leading to intermittent bursts of turbulence.

In this work, we consider helium-4 at $T = 77\ K$, which has kinematic viscosity between $\nu = 1.33 \times 10^{-5}\ \mathrm{m^2/s}$ at 1 atm and $\nu = 1.82 \times 10^{-7}\ \mathrm{m^2/s}$ at 100 atm with a speed of sound ranging from 517.28 m/s to 615.90 m/s [7]. With a characteristic length of $D \approx 100$ nm, the Reynolds number (2.28) is between $\mathrm{Re} = 4$ and $\mathrm{Re} = 340$, well within the laminar flow condition of $\mathrm{Re} \ll 2000$.

2.1.2.3 Laminar Incompressible Flow

For incompressible fluids, the fluid density remains constant and uniform, significantly simplifying fluid problems and analysis. The incompressible assumption is valid for many practical fluid flow problems, especially when dealing with liquids or gases at low Mach numbers, making mathematical treatment even easier than equation (2.27). Continuity equation (2.14) for an incompressible fluid can be written as

$$\vec{\nabla} \cdot \vec{u} = \frac{\partial u_l}{\partial x_l} = 0. \tag{2.29}$$

As such, the diagonal terms of (2.20) vanish, leaving

$$\sigma'_{ij} = \eta\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{2.30}$$

or in vector form

$$\overleftrightarrow{\boldsymbol{\sigma}}' = \eta \left(\vec{\nabla} \vec{u} + \vec{\nabla} \vec{u}^T \right). \tag{2.31}$$

Furthermore, we can remove ρ from the derivatives in (2.18), leaving

$$\frac{\partial \vec{u}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla} \right) \vec{u} = \frac{1}{\rho} \left(\vec{\nabla} \cdot \overleftrightarrow{\boldsymbol{\sigma}} \right). \tag{2.32}$$

Following the same procedure to arrive at (2.27) with (2.32), we arrive at

$$\frac{\partial \vec{u}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla} \right) \vec{u} = -\frac{1}{\rho} \vec{\nabla} P + \nu \vec{\nabla}^2 \vec{u} + \vec{f}. \tag{2.33}$$

Using kinematic pressure p_k and neglecting external forces, we can rewrite (2.33) as

$$\frac{D\vec{u}}{Dt} = -\vec{\nabla} p_k + \nu \vec{\nabla}^2 \vec{u}, \tag{2.34}$$

which is the simplest form of the incompressible Navier-Stokes momentum equation.

2.1.3 Poiseuille Approximation

Laminar incompressible flows are one of the few fluid dynamic problems for which analytical solutions exist. Consider a fluid travelling along the central axis of a cylindrical pipe with length L, radius R, and diameter D, driven by a pressure gradient between the two ends. Assuming a steady-state[4] incompressible laminar flow, (2.34) becomes

$$\nu \vec{\nabla}^2 \vec{u} = \vec{\nabla} p_k. \tag{2.35}$$

Taking z as the central axis, we can exploit cylindrical symmetries to simplify $\vec{u} = (0, 0, u_z(r))$. Furthermore, mass conservation (2.14) with the previous symmetry assumption asserts that the flow is fully developed (*i.e.* that $\partial u_z / \partial z = 0$). If we assume that the pressure gradient is uniform along the central axis, then $\vec{\nabla} p_k \approx \Delta p_k / L$ where Δp_k is the pressure drop across

---

[4] *Steady-state* in fluid dynamics refers to a condition where all system properties are unchanged in time (*i.e.* $\frac{\partial}{\partial t} \to 0$), the opposite of *transient* flow where they can vary. In the absence of driving forces or other time dependencies, all transient flows will tend towards the steady state.

the cylinder, we can write (2.35) as

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z}{\partial r}\right) = -\frac{1}{\nu}\left(\frac{\Delta p_k}{L}\right). \tag{2.36}$$

Since (2.36) is a simple second-order non-homogeneous linear ODE, it is solvable with general solution

$$u_z(r) = -\frac{\Delta p_k}{4\nu L}r^2 + A\log(r) + B, \tag{2.37}$$

where A and B are constants of integration. Since flow must be finite along the central axis (*i.e.* at $r = 0$), $A = 0$. By taking the *no-slip*[5] boundary condition

$$\vec{u}_{||} = u_z(R)\hat{z} = \vec{0}, \tag{2.38}$$

where $\vec{u}_{||}$ is velocity tangential to the boundary, we arrive at Poiseuille's solution [2] of

$$u_z(r) = -\frac{\Delta p_k}{4\nu L}\left(r^2 - R^2\right) = -\frac{\Delta P}{4\eta L}\left(r^2 - R^2\right). \tag{2.39}$$

Thus, the velocity distribution across the pipe is parabolic. Mass flow rate Q with units Kg/s can be easily determined by integrating the mass per unit time ρu_z crossing an annular element $2\pi r \mathrm{d}r$. It is formally defined as

$$Q = \rho \iint \vec{u}\cdot \mathrm{d}\vec{A} = \rho Q_v, \tag{2.40}$$

where Q_v is the *volumetric flow rate.* Using (2.39), we obtain

$$Q = 2\pi\rho\int_0^R r u_z \mathrm{d}r = \frac{\pi\rho\Delta p_k}{8\nu L}R^4 = \frac{\pi\Delta P}{8\nu L}R^4. \tag{2.41}$$

---

[5] The *no-slip* boundary condition asserts that the velocity of a fluid at a solid boundary is equal to the velocity of the boundary itself. For a stationary boundary, this means that the fluid velocity at the surface is zero. For viscous fluids, the velocity gradient (*i.e.* frictional forces) cause the development of a boundary layer described by the Navier-Stokes equations. In contrast, for inviscid flows governed by Euler's equations, frictional forces are neglected, allowing the fluid to theoretically slip past a boundary without resistance, a scenario known as the *perfect slip* boundary condition. This phenomenon has been experimentally observed in superfluid helium-4.

2.1.3.1 Validity of Poiseuille Approximation

The Poiseuille approximation describes incompressible laminar flow through a long cylindrical pipe (*i.e.* $L \gg D$), assuming a fully developed flow with a linear pressure gradient and velocity u_z dependent only on r. This approximation holds sufficiently far downstream from the inlet, where the flow transitions from an initially uniform profile to the parabolic profile given by (2.39). The *entrance region*, where this transition occurs, has a length L_e denoted the *entrance length*, as shown in Fig. 2.1.

The entrance length depends on the balance between inertial and viscous forces, analogous to the Reynolds number. Dimensional analysis suggests $\frac{L_e}{D} \sim \mathrm{Re}$, with a proportionality factor on the order[6] of 0.05. Thus, for the Poiseuille approximation to remain valid, the pipe length must satisfy

$$\frac{L}{D} \gtrsim 0.05\ \mathrm{Re}. \tag{2.42}$$

In many practical cases[7], ensuring $L \gg D$ (*i.e.* $\frac{L}{D} \gtrsim 10$) provides a reasonable approximation for fully developed flow.

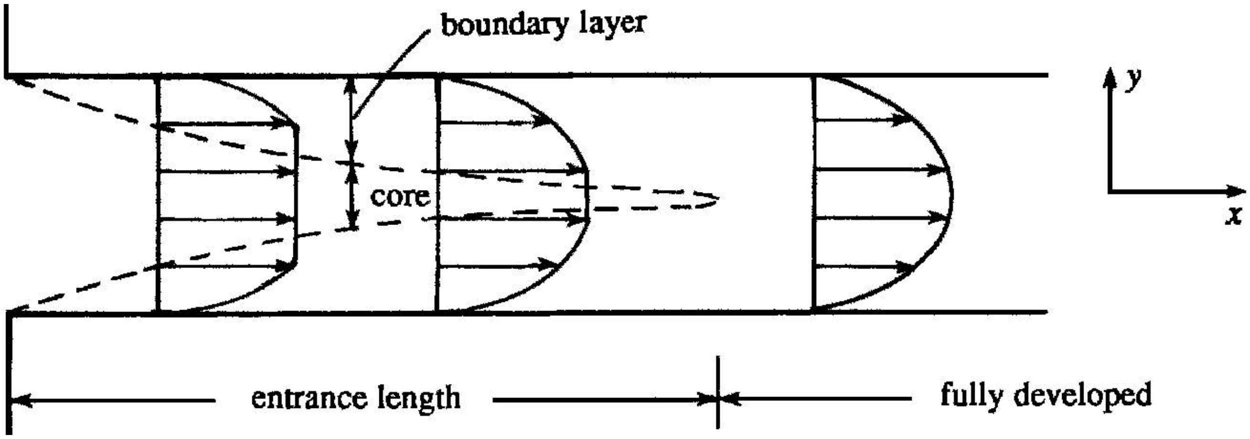


Figure 2.1: Developing and fully developed flows in a cylindrical channel [6].

---

[6] While the exact factor varies in literature, values typically range from 0.03 to 0.08. For example, [8] reports $\frac{L_e}{D} = 0.035$ Re, while [6] reports $\frac{L_e}{D} = 0.06$ Re.

[7] For consistency, if we consider helium-4 at $T = 77$ K with Reynolds number between Re $= 4$ and Re $= 340$ as calculated in Section 2.1.2.2, the Poiseuille approximation is valid between $\frac{L}{D} \gtrsim 0.2$ and $\frac{L}{D} \gtrsim 17$.

2.1.4 Short-Pipe Approximation

Should pipe length L be less than the entrance length L_e, the flow remains within the entrance region and is underdeveloped. As a result, the Poiseuille approximation, which assumes fully developed flow, breaks down, requiring a different approach. To address this, Henry Langhaar (1942) derived a short-pipe model [9] based on the following assumptions:

1. Velocity is constant at inlet and is equal to u_{avg}.
2. The pressure gradient depends on z alone.
3. The z dependence of u_z is at most linear, that is $\vec{\nabla}^2\vec{u} \cdot \hat{z} = 0$.
4. There exists a solution to $\frac{\partial u_z}{\partial t} = \nu\beta^2 u_z$, where β is a function of z only.

Using these assumptions, he linearizes (2.34) as

$$\Delta p_k = \frac{8\nu z}{R^2} u_{avg} + \frac{1}{2}\alpha u_{avg}^2, \tag{2.43}$$

where α is a dimensionless parameter related to fluid acceleration at the pipe outlet. Langhaar estimated $\alpha = 2.28$ [9], a value which was consistent with experiment at the time of publishing. Taking $z = L$, simple arithmetic leads to the following expression for u_{avg},

$$u_{avg} = \frac{8\nu L}{\alpha R^2}\left(\sqrt{1 + \frac{\alpha R^4 \Delta p_k}{32\nu^2 L^2}} - 1\right) = \frac{8\nu L}{\alpha R^2}\left(\sqrt{1 + \frac{\alpha R^4 \Delta P}{32\rho\nu^2 L^2}} - 1\right). \tag{2.44}$$

Following the same procedure as Poiseuille, mass flow Q is

$$Q = \frac{8\pi\rho\nu L}{\alpha}\left(\sqrt{1 + \frac{\alpha R^4 \Delta P}{32\rho\nu^2 L^2}} - 1\right). \tag{2.45}$$

Should $\frac{\alpha R^4 \Delta P}{32\rho\nu^2 L^2} \ll 1$,[8] we expand (2.45) and write as

$$Q \approx \frac{8\pi\rho\nu L}{\alpha}\left(\frac{\alpha R^4 \Delta P}{64\rho\nu^2 L^2}\right) = \frac{\pi \Delta P}{8\nu L} R^4, \tag{2.46}$$

recovering the mass flow condition of Poiseuille for arriving at an identical mass flow condition to Poiseuille for underdeveloped flow.

---

[8] For helium at 77 K and 1 atm, $\rho \approx 0.6$ kg/m^3 and $\nu \approx 1.3 \times 10^{-5}$ m^2/s, and the inequality becomes $\frac{\alpha R^4 \Delta P}{32\rho\nu^2 L^2} \approx \left(7 \times 10^{13}\ \mathrm{m}^{-2}\right)\frac{R^4}{L^2} \ll 1$. If we take $L = 1000$ nm, then this condition is equivalent to $R \ll 350$ nm which is consistent with the $\frac{L}{D} \gtrsim 10$ condition of Poiseuille flow.

2.1.5 Sonic Flow

In fluid dynamics, the speed of sound defines how quickly pressure disturbances propagate through a fluidic medium. For a constant entropy system[9], the speed of sound is defined as

$$c_s = \sqrt{\left(\frac{\partial P}{\partial \rho}\right)_S} \overset{\text{ideal}}{=} \sqrt{\gamma R_s T}, \tag{2.47}$$

where γ is the heat capacity ratio and R_s is the specific gas constant, defined as

$$R_s = \frac{R}{M_{\text{molar}}}, \tag{2.48}$$

where R is the universal gas constant and M_{molar} is the molar mass in kg/mol. Mach number is a dimensionless quantity formally defined as the ratio

$$M = \frac{|\vec{u}|}{c_s}, \tag{2.49}$$

and its value can predict several different compressible regimes.

Flow Regime	**Mach Interval**	**Main Properties**
Incompressible	$M < 0.3$	Density variations can be neglected.
Subsonic	$0.3 \leq M \leq 1.0$	Density and temperature cannot be neglected.
Transonic	$0.8 \leq M \leq 1.2$	Shock waves appear and lead to rapid drag increase.
Supersonic	$1.0 \leq M \leq 3.0$	Information propagates along paths called *characteristics* making calculations easier than subsonic or even incompressible flows.
Hypersonic	$3.0 < M$	Very high flow speeds cause severe heating in boundary layers resulting in dissociation.

Table 2.1: Classification of compressible flow regimes based on Mach number [6].

Using the Mach number definition, the continuity equation (2.1) for one dimensional flow

---

[9] The assumption of constant entropy applies locally where the derivative of (2.47) is taken, rather than requiring entropy to be uniform throughout the system. Physically, this assumption holds when sound waves propagate with a period much smaller than the timescales of thermal diffusion and heat transport, ensuring that perturbations remain nearly isentropic locally.

can be written as

$$u\frac{\partial \rho}{\partial z} + \rho\frac{\partial u}{\partial z} = 0. \tag{2.50}$$

The incompressibility assumption requires[10] that

$$u\frac{\partial \rho}{\partial z} \ll \rho\frac{\partial u}{\partial z}, \qquad \text{or equivalently}^{11} \qquad \frac{\delta\rho}{\rho} \ll \frac{\delta u}{u}. \tag{2.51}$$

Applying the same treatment to (2.2), we obtain

$$u\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial z}, \qquad \text{or equivalently}^{11} \qquad u\delta u \simeq \frac{\delta P}{\rho}. \tag{2.52}$$

From the speed of sound definition (2.47),

$$c_s^2 = \frac{\partial P}{\partial \rho}, \qquad \text{or equivalently}^{12} \qquad \delta P \simeq c_s^2 \delta\rho. \tag{2.53}$$

By combining (2.52) and (2.53), we obtain

$$u\frac{\partial u}{\partial z} = -\frac{c_s^2}{\rho}\frac{\partial \rho}{\partial z}, \qquad \text{or equivalently}^{13} \qquad \frac{\delta\rho}{\rho} \simeq \frac{u^2}{c_s^2}\frac{\delta u}{u} = M^2\frac{\delta u}{u}. \tag{2.54}$$

Comparing (2.51) with (2.54), we see that density changes are negligible if and only if

$$u^2 \ll c_s^2, \qquad \text{or equivalently} \qquad M^2 \ll 1. \tag{2.55}$$

Therefore, the incompressible approximation holds for $M \lesssim 0.3$. A further breakdown of compressible regimes is provided in Table 2.1.

---

[10] For incompressible flow, the steady-state continuity equation must be independent of ρ. In 1D flow, this condition is expressed as $\frac{\partial u}{\partial z} = 0$, but more generally for $\vec{\nabla} \cdot (\rho\vec{u}) = \vec{u} \cdot \vec{\nabla}\rho + \rho(\vec{\nabla} \cdot \vec{u}) = 0$ to be independent of ρ, then $\vec{u} \cdot \vec{\nabla}\rho$ must be negligible compared to $\rho(\vec{\nabla} \cdot \vec{u})$ to arrive at $\vec{\nabla} \cdot \vec{u} = 0$.

[11] Here, we consider variations across some fixed characteristic lengthscale.

[12] Here, we consider perturbations associated with sound waves.

[13] While the variations used in (2.52) and (2.53) represent different physical processes, they can be treated the same provided the sound waves and the background flow in which they propagate are both locally isentropic.

2.1.5.1 Knudsen Number

The Knudsen number is a dimensionless number defined as

$$K_n = \frac{\lambda}{D}, \tag{2.56}$$

where D is the characteristic length of the system defined in Section 2.1.2.2 and λ is the mean free path[14] of the fluid constituents. From the kinetic theory of gases, $\lambda = (\sqrt{2}n\sigma)^{-1}$, where σ is the effective cross-sectional area of the particle. By using the ideal gas law and considering spherical atoms with *hard shell* diameter[15] d, λ is written as

$$\lambda = \frac{1}{\sqrt{2}\pi d^2 n} = \frac{k_B T}{\sqrt{2}\pi d^2 P}, \tag{2.57}$$

and (2.56) becomes

$$K_n = \frac{k_B T}{\sqrt{2}\pi d^2 P D}. \tag{2.58}$$

It is possible to express Knudsen number K_n in terms of easily calculable values previously defined. From kinetic theory, the average velocity of a fluid constituent molecule given by

Transport Regime	**Knudsen Interval**	**Main Properties**
Continuum	$K_n < 0.001$	Inter-molecular collisions dominated system. No rarefaction effects are present.
Slip	$0.001 \leq K_n \leq 0.1$	Rarefaction effects modify flow behaviour near the walls.
Transition	$0.1 \leq K_n \leq 10$	Between slip and molecular flow, gas rarefaction affects are strong and must be analyzed statistically; *i.e.* with Boltzmann's equation.
Molecular	$10 < K_n$	System dominated by kinetic theory of statistical physics, where molecules must be modelled individually; *i.e.* with Monte Carlo.

Table 2.2: Classification of transport regimes based on Knudsen number [10].

---

[14] Mean free path λ is the average distance between collisions of moving particles that result in a substantial change in momentum.

[15] Hard shell diameter, or *kinetic* diameter, is used to model particle interactions as a measure of collision likelihood or as an indication of the size of a molecule as a target. Helium-4 has a hard shell diameter of $d = 2.6$ Å.

the Maxwell-Boltzmann distribution is [8]

$$\overline{v} = \sqrt{\frac{8k_BT}{\pi m}}, \tag{2.59}$$

with a Newtonian-like molecular dynamic viscosity given by

$$\eta = \frac{1}{3}\rho\overline{v}\lambda = \frac{2}{3\pi}\frac{\sqrt{\pi m k_BT}}{\pi d^2}, \tag{2.60}$$

which was obtained by Maxwell in 1860 [11]. Solving for mean free path, we obtain

$$\lambda = \frac{\eta}{\rho}\sqrt{\frac{\pi m}{2k_BT}}, \tag{2.61}$$

with Knudsen number

$$K_n = \frac{\eta}{\rho D}\sqrt{\frac{\pi m}{2k_BT}}. \tag{2.62}$$

If we consider the ratio of the Mach and Reynolds numbers, defined (2.49) and (2.28), using the definitions of speed of sound (2.47) we obtain

$$\begin{aligned}\frac{M}{\mathrm{Re}} &= \frac{u/c_s}{\rho u D/\eta} = \frac{\eta}{\rho D c_s} = \frac{\eta}{\rho D\sqrt{\gamma R_s T}} = \frac{\eta}{\rho D}\sqrt{\frac{m}{\gamma k_B T}} \\ &= \left(\frac{\eta}{\rho D}\sqrt{\frac{m}{\gamma k_BT}}\sqrt{\frac{\gamma\pi}{2}}\right)\sqrt{\frac{2}{\gamma\pi}} = \left(\frac{\eta}{\rho D}\sqrt{\frac{\pi m}{2k_BT}}\right)\sqrt{\frac{2}{\gamma\pi}},\end{aligned} \tag{2.63}$$

where m is the molecular mass in kg. Substituting (2.62) into (2.63) and solving for K_n yields:

$$K_n = \frac{M}{\mathrm{Re}}\sqrt{\frac{\gamma\pi}{2}}. \tag{2.64}$$

For helium-4, equation (2.56) gives $K_n = 0.35$ at 1 atm and $K_n = 0.0035$ at 100 atm, which is well within the slip regime shown in Table 2.2. Estimations from equation (2.64) from $M = 0.2$ to 2 gives a range of $K_n = 0.1$ and $K_n = 0.01$ from $P = 1$ atm to 100 atm, within the slip regime as well.

In the slip regime, boundary condition (2.38) can no longer be used. Instead, assuming a negligible temperature gradient, velocity parallel to the wall of a cylindrical pipe can be written as [8]

$$\vec{u}_{\text{slip}} = \zeta \frac{\partial \vec{u}_{||}}{\partial n}\bigg|_{\text{wall}}, \tag{2.65}$$

where ζ is the *slip parameter*[16] and $\vec{u}_{||}$ is tangential to the wall. For arbitrary vector $\vec{A}$, $\partial\vec{A}/\partial n$ is defined as $\overleftrightarrow{\nabla}\vec{A}\cdot\hat{n}$. Originally proposed by Maxwell in 1879 [12], slip parameter ζ can be written as a function of mean free path λ and the tangential momentum accommodation coefficient σ [13],

$$\zeta = \frac{2-\sigma}{\sigma}\lambda, \tag{2.66}$$

where σ is between 0 and 1 for full slip and no-slip respectively [13].

For helium-4 at $T = 77$ K between $P = 1$ atm and $P = 100$ atm, mean free path λ is between 35 nm and 0.35 nm. Taking $\sigma = \frac{1}{2}$ for partial slip, we arrive at a slip parameter ζ between 100 nm and 1 nm. By approximating $\frac{\partial \vec{u}_{||}}{\partial n} \approx \frac{u_\infty}{D}$, where u_∞ is the free-stream velocity and D is characteristic length, the slip boundary condition becomes

$$\vec{u}_{\text{slip}} = \zeta \frac{\partial \vec{u}_{||}}{\partial n}\bigg|_{\text{wall}} \approx \frac{2-\sigma}{\sigma}\lambda\frac{\vec{u}_\infty}{D} = K_n\frac{2-\sigma}{\sigma}\vec{u}_\infty, \tag{2.67}$$

which at its maximum is $\vec{u}_{\text{slip}} = \vec{u}_\infty$ at $P = 1$ atm for $D = 100$ nm.

More generally, (2.65) can be written as a function of tangential shear stress τ, that is

$$\vec{u}_{\text{slip}} = -\frac{2-\sigma}{\sigma}\frac{\lambda}{\eta}\vec{\tau} = -\frac{2-\sigma}{\sigma}\lambda\left(\hat{n}\cdot\overleftrightarrow{\boldsymbol{\sigma}}'\right)\cdot\left(\overleftrightarrow{\boldsymbol{I}}-\hat{n}\hat{n}^T\right), \tag{2.68}$$

where $\overleftrightarrow{\boldsymbol{I}}$ is the identity tensor and $\overleftrightarrow{\boldsymbol{\sigma}}'$ is the viscous stress tensor. Taking the Navier-Stokes formulation (2.20), we obtain (2.65)

$$\vec{\tau} = -\eta\left(\hat{n}\cdot\overleftrightarrow{\boldsymbol{\sigma}}'\right)\cdot\left(\overleftrightarrow{\boldsymbol{I}}-\hat{n}\hat{n}^T\right) = \eta\left(\frac{\partial \vec{u}_{||}}{\partial n}+\frac{\partial \vec{u}_{\perp}}{\partial \mathrm{t}}\right) = \eta\frac{\partial \vec{u}_{||}}{\partial n}, \tag{2.69}$$

where $\hat{\mathrm{t}}$ is the unit tangent vector and $\vec{u}_\perp = 0$ at an impermeable wall, which is equivalent to (2.65).

---

[16] Slip parameter ζ, commonly called the *coefficient of slip* or *slip length*, is a function of Knudsen number, geometry, and fluid-specific constants, and is often determined experimentally. $\zeta = 0$ implies no slip, whereas $\zeta = \infty$ is perfect slip, shown in Figure 2.2.

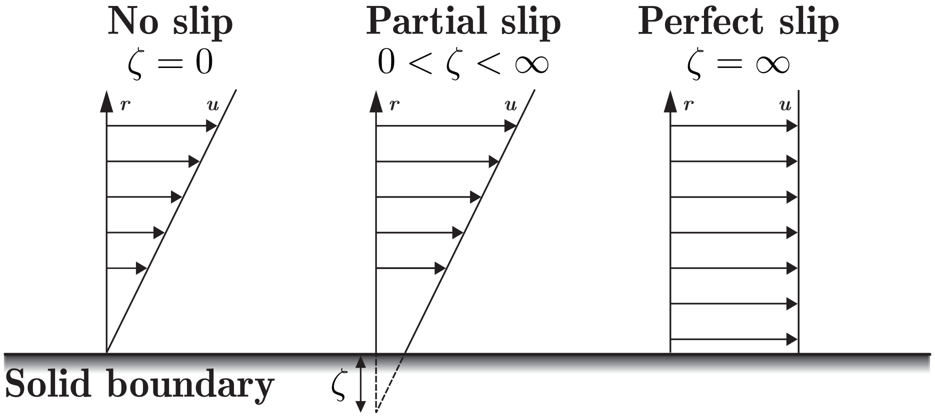


Figure 2.2: Slip boundary conditions dependent on the value of ζ.

2.2 Acoustic Black Holes

Acoustic black holes, also known as sonic black holes, are theoretical phenomena where sound waves within a fluid are unable to escape a particular region, analogous to how light is unable to escape a gravitational black hole. This concept extends the principles of general relativity to fluid dynamics, providing insights into both fields.

The study of acoustic black holes has significant implications for understanding astrophysical black holes. As will be seen later in the chapter, creating conditions where fluid flow exceeds the speed of sound allows one to study an analogous system to an astrophysical black hole safely in a laboratory.

2.2.1 de Laval nozzle

A de Laval nozzle, also known as a converging-diverging (CD) nozzle, is a geometric structure capable of accelerating gases to supersonic speeds. When a subsonic gas or fluid enters the convergent section, its velocity increases as the area decreases.

Starting with the conservation of mass flow equation derived from the mass conservation equation (2.1) and assuming flow is in the z-axis along a streamline

$$Q_m = u_z \rho A = \text{constant}, \tag{2.70}$$

differentiating both sides with respect to the z axis yields

$$\frac{\partial Q_m}{\partial z} = \frac{1}{u_z}\frac{\partial u_z}{\partial z} + \frac{1}{\rho}\frac{\partial \rho}{\partial z} + \frac{1}{A}\frac{\partial A}{\partial z} = 0. \tag{2.71}$$

The momentum equation (2.2) can be written as,

$$u_z\frac{\partial u_z}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial z} = -\frac{1}{\rho}\frac{\partial \rho}{\partial z}\frac{\partial P}{\partial \rho}, \tag{2.72}$$

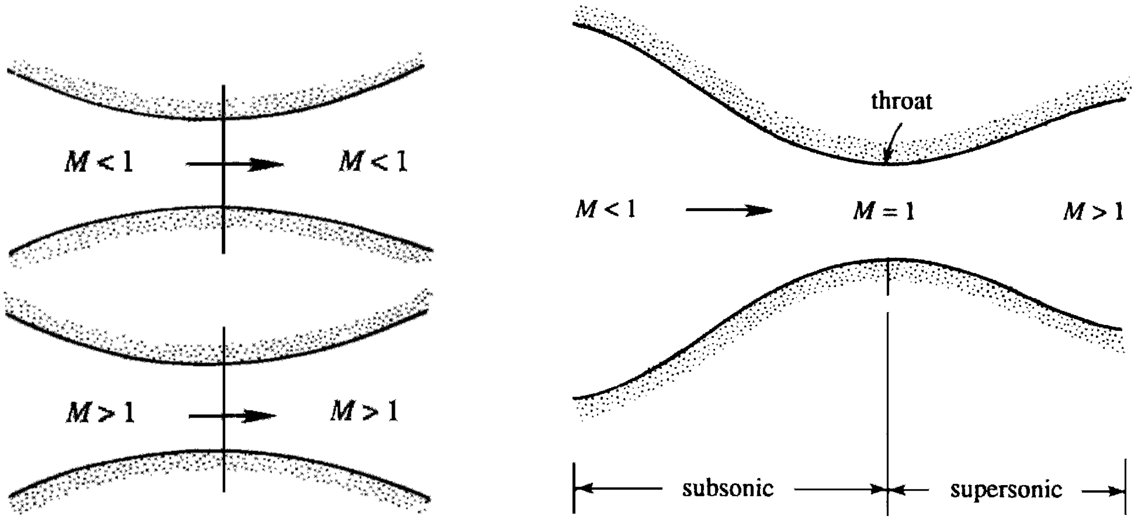
by chain rule. Substituting (2.47), we obtain

$$u_z\frac{\partial u_z}{\partial z} = -\frac{c_s^2}{\rho}\frac{\partial \rho}{\partial z} \Rightarrow \frac{1}{\rho}\frac{\partial \rho}{\partial z} = -\frac{u_z^2}{c_s^2}\left(\frac{1}{u_z}\frac{\partial u_z}{\partial z}\right), \tag{2.73}$$

and by substituting into (2.71), we arrive at

$$\left(1 - M^2\right)\frac{1}{u_z}\frac{\partial u_z}{\partial z} = -\frac{1}{A}\frac{\partial A}{\partial z} = \frac{1}{M}\left(1 - M^2\right)\frac{\partial M}{\partial z}. \tag{2.74}$$

By examining (2.74), we gain insight on the behaviour of different regimes. When the flow is subsonic ($M < 1$), $\frac{\partial u_z}{\partial z}$ and $\frac{\partial A}{\partial z}$ will have opposite signs, meaning a decrease in cross-sectional area will increase velocity and vice-versa. Conversely, when the flow is supersonic ($M > 1$),



(A) Flow in which the $M = 1$ condition at the throat is not reached.

(B) Flow is accelerated from subsonic to supersonic when $M = 1$ is reached at the throat.

Figure 2.3: Flow in a converging-diverging (de Laval) nozzle depending on Mach number at the throat (modified from [6]).

$\frac{\partial u_z}{\partial z}$ and $\frac{\partial A}{\partial z}$ will have the same sign, meaning an increasing area accelerates flow. This behaviour for both subsonic and supersonic flow is shown in Figure 2.3A. While this appears counter intuitive, when $M = 1$, $\frac{\partial A}{\partial z} = 0$, which means that transitions between regimes can only occur at cross section minimas, shown in Figure 2.3B. This is known as the *choked flow*[17] condition. Should this condition be reached, the converging region of the nozzle will be under the subsonic regime, and the diverging region will be under the supersonic regime, making flow through the nozzle transonic.

2.2.2 1D Isentropic Flow Approximation

When a fluid passes through a de Laval nozzle, velocity is nonuniform along the central axis even for fully-developed steady-state flow. If a nozzle has small curvature (*i.e.* $\frac{dA}{dz}$ is small), radial velocity can be completely neglected and flow occurs only along the central axis. With these approximations, we can assume that the flow is *isentropic*, that is a flow where entropy remains constant. For ideal gases, the isentropic relation is

$$\frac{T_2}{T_1} = \left(\frac{P_2}{P_1}\right)^{\frac{\gamma-1}{\gamma}} = \left(\frac{\rho_2}{\rho_1}\right)^{\gamma-1}. \tag{2.75}$$

As such, our flow can be considered as one dimensional, and changes in area can both compress/decompress and accelerate/decelerate flow. The first law of thermodynamics can be written as

$$h_0 = h + \frac{u^2}{2}, \tag{2.76}$$

where h is enthalpy, h_0 is enthalpy at stagnation[18], and u is fluid flow along the central axis. Substituting for enthalpy and assuming constant specific heat capacity c_p at constant pressure, the stagnation temperature is given by

$$T_0 - T = \frac{u^2}{2c_p}, \tag{2.77}$$

---

[17] Choked flow occurs when fluid velocity reaches c_s at the throat, leading to a maximum volumetric flow rate. Further decreases in downstream pressure do not increase the mass flow rate.

[18] Stagnation of any field occurs at the *stagnation point* where fluid velocity is zero (*i.e.* static).

where T_0 is the stagnation or *total* temperature, and T is the fluid temperature at velocity u. Using the definition of the speed of sound (2.47), the definition of Mach number (2.49), the isentropic relation (2.75), and the stagnation temperature (2.77), the ideal gas law for adiabatic compression/expansion gives the following stagnation relationships[19],

$$T_0 = T\left(1+\frac{\gamma-1}{2}M^2\right), \tag{2.78}$$

$$P_0 = P\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}}, \tag{2.79}$$

$$\rho_0 = \rho\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{1}{\gamma-1}}, \tag{2.80}$$

and we can write (2.76) for both static and dynamic conditions as

$$h_0 = \frac{\gamma}{\gamma-1}\frac{P_0}{\rho_0}, \quad h = \frac{\gamma}{\gamma-1}\frac{P}{\rho}. \tag{2.81}$$

By substituting (2.81) into (2.76), using (2.75) to relate stagnant and downstream conditions (*i.e.* $\rho = \rho_0[P/P_0]^{1/\gamma}$) and solve for u, it can be easily shown that

$$u(z) = \sqrt{\frac{2\gamma}{\gamma-1}\frac{P_0}{\rho_0}\left[1-\left(\frac{P(z)}{P_0}\right)^{\frac{\gamma-1}{\gamma}}\right]}. \tag{2.82}$$

If the downstream pressure is not known however, we define the *sonic* temperature T^*, pressure P^*, density ρ^*, and area A^* by computing (2.78-2.80) at $M = 1$, arriving at the choked flow relationships

$$T^* = T_0\left(\frac{2}{\gamma+1}\right), \tag{2.83}$$

$$P^* = P_0\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}}, \tag{2.84}$$

$$\rho^* = \rho_0\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}}, \tag{2.85}$$

---

[19] The stagnation relationships (2.78-2.80) define static values of temperature, pressure, and density as functions of their respective dynamic values at a given Mach number.

or as a function of *local* Mach number[20] $M_L = u/c_s$

$$T^* = T\left(\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}M_L^2\right), \tag{2.86}$$

$$P^* = P\left(\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}M_L^2\right)^{\frac{\gamma}{\gamma-1}}, \tag{2.87}$$

$$\rho^* = \rho\left(\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}M_L^2\right)^{\frac{1}{\gamma-1}}, \tag{2.88}$$

calculated by substituting (2.78-2.80) as a function of M_L back into (2.83-2.85). Conservation of mass (2.40) can then be written as

$$Q = \rho u A = \rho^* c_s^* A^* = \text{const}, \tag{2.89}$$

where c_s^* can be easily determined by substituting (2.83) into (2.47). We can expand (2.89) as

$$Q = \rho u A = \rho M_L \sqrt{\gamma R_s T} A, \tag{2.90}$$

$$\begin{aligned} Q &= \rho^* c_s^* A^* = \rho\sqrt{\gamma R_s T^*} A^* \\ &= \rho\sqrt{\gamma R_s T} A^* \left(\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}M_L^2\right)^{-\frac{1}{\gamma-1}-1/2}. \end{aligned} \tag{2.91}$$

Equating (2.90) and (2.91) lets us write the sonic cross-sectional area as

$$\frac{A}{A^*} = \frac{1}{M_L}\left(\frac{2}{\gamma+1} + \frac{\gamma-1}{\gamma+1}M_L^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}, \tag{2.92}$$

which is an incredibly powerful formula, allowing us to analytically determine both the velocity of the inlet and outlet with only nozzle geometry.

Combining the equations of this section, we arrive at the mass flow equation for a compressible isentropic fluid in a converging-diverging nozzle:

$$Q = P_0 M_L A \sqrt{\frac{\gamma}{R_s T_0}} \left(1 + \frac{\gamma-1}{2}M_L^2\right)^{-\frac{\gamma+1}{2(\gamma-1)}} = P_0 A^* \sqrt{\frac{\gamma}{R_s T_0}} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{2.93}$$

---

[20] The *local* Mach number is defined as the Mach number for non-sonic values in (2.83-2.85).

2.3 Unruh Effect

In 1976, William G. Unruh proposed that bodies undergoing uniform acceleration will detect blackbody radiation at a temperature proportional to their acceleration, even in what would be perceived as the vacuum of space to an inertial observer at rest [14]. This work followed upon Stephen Hawking's proposal on black hole evaporation [15], commonly referred to as *Hawking radiation.* After applying similar mathematical treatments to accelerating bodies as Hawking performed with black holes, Unruh found that accelerating bodies emit radiation in the form of photons, commonly referred to as the *Unruh effect.*

2.3.1 Unruh Radiation

In his 1981 paper, Unruh extended his theoretical work on the Unruh effect by considering the behaviour of phonons in a fluid flowing at transonic speeds. Aiming to provide an experimental analogue for black hole evaporation, he mathematically showed that the sonic event horizon should generate a thermal spectrum of sound waves [16], showing how similar quantum effects could manifest in fluid dynamics. He showed a striking analogy between sonic and astrophysical event horizons, which include analogous forms of surface gravity, temperature spectrum, and even spacetime metric. Fundamentally, in sonic black holes, vacuum is replaced by fluid, electromagnetic waves are replaced by sound waves, and photons are replaced by phonons.

Unruh starts by linearizing Euler's equations (2.1, 2.2) by defining $\vec{u} = \vec{\nabla}\psi$ for some scalar function ψ, with the additional condition of an irrotational fluid ($\vec{\nabla} \times \vec{u} = \vec{0}$). With velocity $\vec{u} = \vec{\nabla}\psi$, Unruh selects exact solutions P_0, ρ_0, and ψ_0, where $\vec{u}_0 = \vec{\nabla}\psi_0$, with linear fluctuations P_1, ρ_1, ψ_1, $\vec{u}_1$, etc. After much manipulation, Unruh applies Euler's equations with these fluctuations to obtain an equation of motion,

$$\frac{\partial}{\partial t}\left(\frac{\rho_0}{c_s^2}\left(\frac{\partial \psi_1}{\partial t}+\vec{u}_0\cdot\vec{\nabla}\psi_1\right)\right)-\vec{\nabla}\cdot\left(\rho_0\vec{\nabla}\psi_1-\frac{\rho_0}{c_s^2}\vec{u}_0\left(\frac{\partial \psi_1}{\partial t}+\vec{u}_0\cdot\vec{\nabla}\psi_1\right)\right)=0, \tag{2.94}$$

which is identical to the wave equation in curved spacetime,

$$\frac{1}{\sqrt{-g}}\frac{\partial}{\partial x^\mu}\left(\sqrt{-g}g^{\mu\nu}\frac{\partial}{\partial x^\nu}\right)\psi=0, \tag{2.95}$$

where $g \equiv \left[det\big(g^{\mu\nu}\big)\right]^{-1}$ for the metric

$$g^{\mu\nu}\left(t,\vec{x}\right)=\frac{1}{\rho_0 c_s}\begin{bmatrix}-1 & \dots & -u_0^\nu \\ \dots & \ddots & \dots \\ -v_0^\mu & \dots & (c_s^2\delta_{\mu\nu}-u_0^\mu u_0^\nu)\end{bmatrix}, \tag{2.96}$$

which is the inverse metric of the *acoustic metric*, given by [3]

$$g_{\mu\nu}^{\text{acoustic}}\left(t,\vec{x}\right)=\frac{\rho_0}{c_s}\begin{bmatrix}-\left(c_s^2-u_0^2\right) & \dots & -[u_0]_\nu \\ \dots & \ddots & \dots \\ -[u_0]_\mu & \dots & \delta_{\mu\nu}\end{bmatrix}. \tag{2.97}$$

Unruh then defines the following change of time coordinate

$$\tau=t+\int dr\frac{u_0(r)}{c_s^2-u_0^2(r)}, \tag{2.98}$$

assuming that background flow is spherically symmetric, stationary, and convergent. Near the sonic horizon at some $r=R$, background velocity $u_0=\vec{\nabla}\psi_0$ exceeds the speed of sound c_s. By assuming that a smooth transonic transition occurs at the horizon, Unruh approximates the background velocity u_0 near horizon $r=R$ as

$$u_0\approx -c_s+\alpha(r-R)+\mathcal{O}\left((r-R)^2\right), \tag{2.99}$$

which when applied to (2.97) gives the following spacetime metric:

$$ds^2 = \frac{\rho_0}{c_s}\left((c_s^2 - u_0^2)d\tau^2 - \frac{c_s dr^2}{c_s^2 - u_0^2} - r^2(d\theta^2 + \sin^2\theta d\phi^2)\right) \tag{2.100}$$

$$\approx \frac{\rho_0}{c_s}\left(2c_s\alpha(r-R)d\tau^2 - \frac{dr^2}{2\alpha(r-R)} - r^2 d\Omega^2\right), \tag{2.101}$$

where $d\Omega^2 = d\theta^2 + \sin^2\theta d\phi^2$ is the metric on the unit two-sphere. This metric is nearly identical to the near-horizon approximation[21] of the Schwarzschild metric of an astrophysical black hole which, taking $G = \hbar = c = k_B = 1$, is

$$ds^2 \approx \left(\frac{r-2M}{2M}\right)dt^2 - \left(\frac{2M}{r-2M}\right)dr^2 - r^2 d\Omega^2. \tag{2.102}$$

By performing a similar derivation of radiation temperature as Hawking did, Unruh determined the phonon temperature spectrum along the direction of flow,

$$T_{Unruh} = \frac{\hbar}{2\pi k_B}\frac{\partial u_0}{\partial r}\bigg|_{horizon} = \frac{\hbar\kappa}{2\pi k_B c_s}, \tag{2.103}$$

where κ is surface gravity, given by [17]

$$\kappa = c_s\frac{\partial u_0}{\partial r}\bigg|_{horizon}, \tag{2.104}$$

which compares to the Schwarzschild surface gravity of ([15], in SI units)

$$\kappa = \frac{c^4}{4GM}, \tag{2.105}$$

and to the Hawking's temperature of [15, 16]

$$T_{Hawking} = \frac{\hbar c^3}{8\pi k_B GM} = \frac{\hbar\kappa}{2\pi k_B c}. \tag{2.106}$$

In the case of a de Laval nozzle with flow along the central axis z, (2.103) would be written as

$$T_{nozzle} = \frac{\hbar}{2\pi k_B}\frac{\partial u_z}{\partial z}\bigg|_{throat}. \tag{2.107}$$

---

[21] Often given without proof, the *near-horizon* approximation of the Schwarzschild metric simplifies the description of spacetime close to the event horizon for easier mathematical treatment. Taking $r = 2M + \epsilon$ for some small ϵ, we can write $g_{tt} = -g_{rr}^{-1} = 1 - \frac{2M}{r} = 1 - \frac{2M}{2M+\epsilon} = \frac{\epsilon}{2M+\epsilon} \approx \frac{\epsilon}{2M} = \frac{r-2M}{2M}$.

The rate of phonon emission can be calculated from the Stefan-Boltzmann law

$$j = \varepsilon \sigma T^4, \tag{2.108}$$

where j is the energy radiated per unit surface area, ε is the emissivity[22], σ is the Stefan-Boltzmann constant[23] and T is phonon temperature (2.103). The sonic Stefan-Boltzmann constant σ_{sound} is given by [3]

$$\sigma_{sound} = \frac{\pi^2}{120} \frac{k_B^4}{c_s^2 \hbar^3}. \tag{2.109}$$

By assuming perfect blackbody radiation (*i.e.* $\varepsilon = 1$) with phonon energy given by $E = k_B T$, the phonon emission rate Φ_{phonon} for a CD nozzle can be found by multiplying equation (2.108) by the throat area A^* and dividing by the phonon energy, resulting in

$$\Phi_{phonon} = \varepsilon \frac{jA^*}{E_{phonon}} \approx \frac{\sigma_{sound} A^* T_{nozzle}^4}{k_B T_{nozzle}} = \frac{\pi^2 A^*}{120} \frac{k_B^3}{c_s^2 \hbar^3} T_{nozzle}^3 = \frac{A^*}{960 \pi c_s^2} \frac{\partial u_z}{\partial z}\bigg|_{throat}^3. \tag{2.110}$$

If the same mathematical treatment is applied to astrophysical black hole evaporation, *i.e.* treating emitted photons as perfect blackbody radiation that emit on the thermal spectrum derived by Stephen Hawking given by equation (2.106), we arrive at

$$\Phi_{photon} = \varepsilon \frac{jA}{E_{photon}} \approx \frac{\sigma A \, T_{Hawking}^4}{k_B T_{Hawking}} = \frac{2A}{960 \pi c^2} \left(\frac{\kappa}{c}\right)^3. \tag{2.111}$$

This result is nearly identical[24] to equation (2.110), demonstrating the elegant connection between astrophysical event horizons and their sonic analogues. Thus, Unruh's derivations bridge the theoretical framework of black hole event horizons to its experimental counterpart in fluid dynamics, offering the potential to explore the complex and largely unknown physics of astrophysical black holes in a laboratory setting.

---

[22] The *emissivity* is a measure of effectiveness in emitting energy as thermal radiation. Typical values are between 0 and 1, where $\varepsilon = 1$ is the case for a perfect blackbody.

[23] The Stefan-Boltzmann constant is given by $\sigma = \frac{2\pi^5 k_B^4}{15 c^2 h^3} = \frac{\pi^2}{60} \frac{k_B^4}{c^2 \hbar^3}$.

[24] Recall that $\frac{\partial u_z}{\partial z}\big|_{throat} = \frac{\kappa}{c_s}$.

Chapter 3

Computational Fluid Dynamics

3.1 Philosophy

Computational Fluid Dynamics (CFD) uses numerical analysis to solve fluid dynamics problems. CFD allows for the modelling of complex systems and provides approximate solutions where analytical solutions are otherwise impossible. Let us consider calculating fluid flow along a pipe using CFD. In order to perform meaningful calculations in fluid dynamics, you first require a description of the problem by defining the domain occupied by the fluid (*i.e.* the internal region of the pipe), selecting governing equations that specifically apply to the expected fluid behaviour, and choosing conditions at domain boundaries and within the domain itself. Afterwards, selected numerical methods are used to iteratively solve the problem, enabling a detailed analysis of the system in question.

3.2 Discretization Methods

Discretization is the process of transforming continuous functions into discrete forms. There are two main types of discretization: domain discretization and mathematical model discretization. Domain discretization consists of dividing a continuous spatial domain into a finite number of small discrete segments. In the limit of infinite discrete segments, we recover the continuous spatial domain. Mathematical model discretization then converts the mathematical model (*i.e.* the set of governing equations) to relations between the discrete points of the domain, allowing for numerical calculation at all points.

3.2.1 Finite Difference Method

The Finite Difference Method (FDM) is a discretization process consisting of approximating the derivatives of governing equations by using the *difference approximation.* The first step in obtaining numerical solutions with FDM is to discretize the spatial domain with a finite number of nodes, *i.e.* a numerical grid for each spatial dimension. In FDM, the resulting grids typically look structured. Each node is uniquely identified by a set of indices, (i, j) for example, which are typically the indices of the grid lines in which they coincide. Neighboring nodes are defined by incrementing or decrementing one of the indices. The grid lines, which do not intersect, are typically selected parallel to the local coordinate

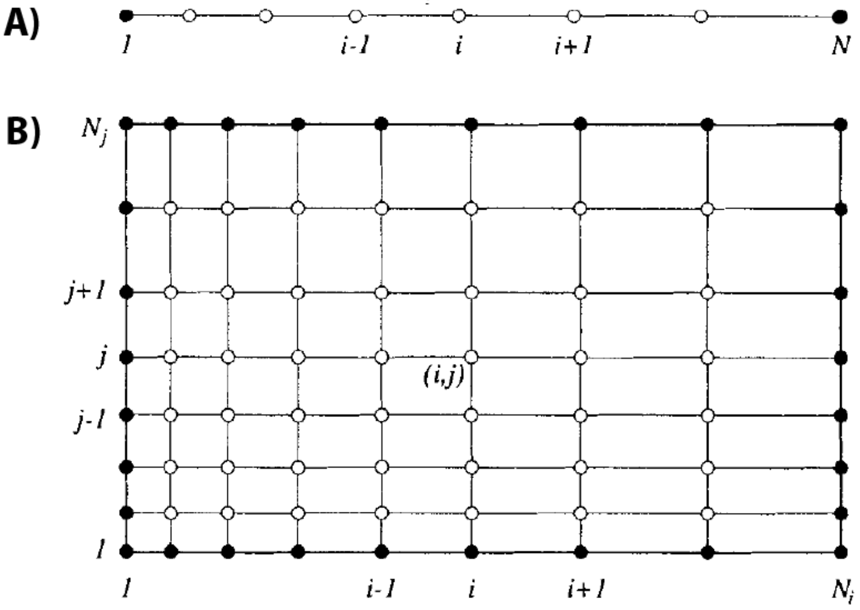


Figure 3.1: Example structured non-uniform grid used in the Finite Difference Method for **(A)** a one-dimensional spatial domain and **(B)** a two-dimensional spatial domain, from [18].

system axes. An example is shown in Figure 3.1 for a non-uniform grid for both one and two dimensions.

The second step involves discretizing the mathematical model, *i.e.* a set of PDEs, by relating the derivatives at grid points with their neighbours. The approximation comes directly from Taylor's theorem, which states that any continuous differentiable function $\phi(x)$ can, in the vicinity of x_i, be expressed as

$$\phi(x) = \phi_i + (x - x_i)\left(\frac{\partial \phi}{\partial x}\right)_i + \cdots + \frac{(x - x_i)^n}{n!}\left(\frac{\partial^n \phi}{\partial x^n}\right)_i + h_n(x)(x - x_i)^n, \tag{3.1}$$

where $h_n(x)$ is called the remainder function and approaches zero as $x \to x_i$ and ϕ_i is shorthand for $\phi(x_i)$. By replacing[1] x with x_{i+1} or x_{i-1} and isolating the first derivative of (3.1), we obtain

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + H, \tag{3.2}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_i - \phi_{i-1}}{x_i - x_{i-1}} - \frac{x_i - x_{i-1}}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + H, \tag{3.3}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_{i-1}}{x_{i+1} - x_{i-1}} - \frac{(x_{i+1} - x_i)^2 - (x_i - x_{i-1})^2}{2(x_{i+1} - x_{i-1})}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + H, \tag{3.4}$$

where H denotes higher order terms. Assuming that the difference between adjacent nodes is sufficiently small, we can take the first term in each approximation to obtain, for equally spaced points,

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} + \mathcal{O}(\Delta x), \tag{3.5}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_i - \phi_{i-1}}{\Delta x} + \mathcal{O}(\Delta x), \tag{3.6}$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2), \tag{3.7}$$

which is known as the forward difference scheme (FDS), backward difference scheme (BDS), and central difference scheme (CDS) respectively, and their accuracies are shown in Figure

---

[1] The forward difference scheme substitutes x with x_{i+1}, the backward difference scheme substitutes x with x_{i-1}, the central difference scheme substitutes both x with x_{i+1} and x_i with x_{i-1}.

3.2. The omitted terms from (3.5-3.7) are called *truncation errors* which measure the approximation accuracy. For second-order derivatives, the same procedure can be taken with the approximation that

$$\left(\frac{\partial^2\phi}{\partial x^2}\right)_i \approx \frac{\left(\frac{\partial\phi}{\partial x}\right)_{i+1} - \left(\frac{\partial\phi}{\partial x}\right)_i}{x_{i+1} - x_i}, \qquad \text{(FDS/BDS)} \qquad (3.8)$$

$$\left(\frac{\partial^2\phi}{\partial x^2}\right)_i \approx \frac{\left(\frac{\partial\phi}{\partial x}\right)_{i+\frac{1}{2}} - \left(\frac{\partial\phi}{\partial x}\right)_{i-\frac{1}{2}}}{\frac{1}{2}(x_{i+1} - x_{i-1})}, \qquad \text{(CDS)} \qquad (3.9)$$

using the forward/backward and central difference schemes respectively. For equidistant spacing of nodes, (3.8) and (3.9) become

$$\left(\frac{\partial\phi}{\partial x}\right)_i \approx \frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{(\Delta x)^2}. \qquad (3.10)$$

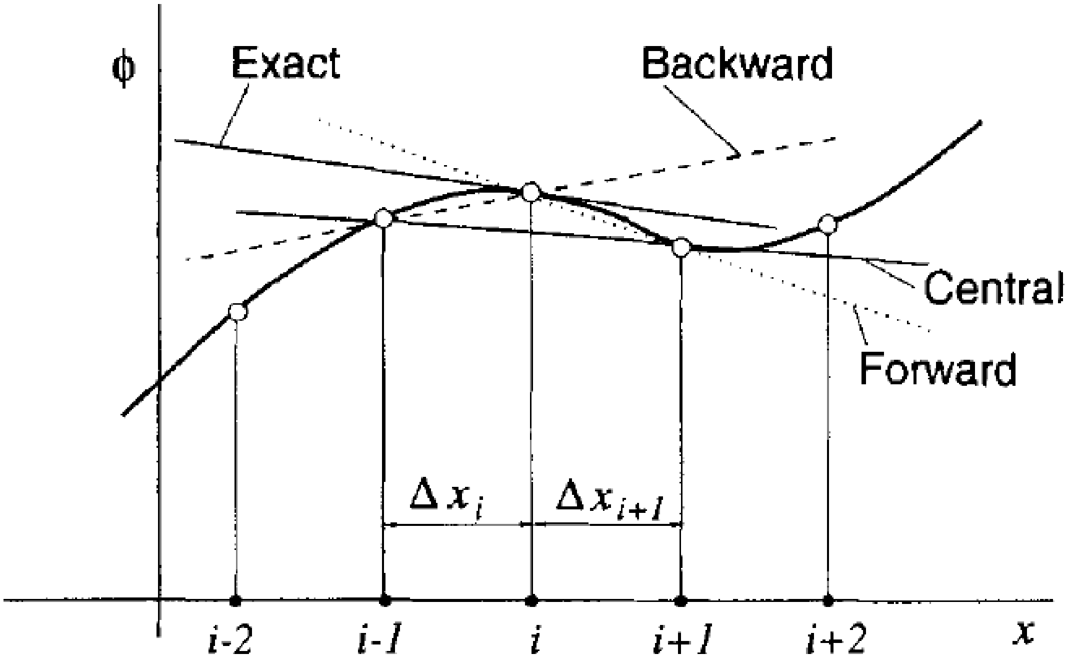


Figure 3.2: Comparison of first order derivative discretization schemes, from [18].

Although one of the oldest discretization methods, it is the easiest to implement in simple geometries, but more complicated geometries get rather cumbersome [18].

3.2.2 Finite Volume Method

Instead of discretizing partial derivatives directly, the Finite Volume Method (FVM) takes the integral form of a partial derivative equation as a starting point, and thus is conservative by construction. The solution domain is divided into a finite number of small arbitrarily shaped control volumes (CVs) by a suitable grid[2] and are assigned a computational node to the CV center[3], shown in Figure 3.3A. For the mathematical model, terms that contain $\vec{\vec{\nabla}}$ are converted to surface integrals using the divergence theorem and are evaluated as fluxes at the surface of each finite volume, and scalar terms are computed directly as volume integrals centered at computational nodes.

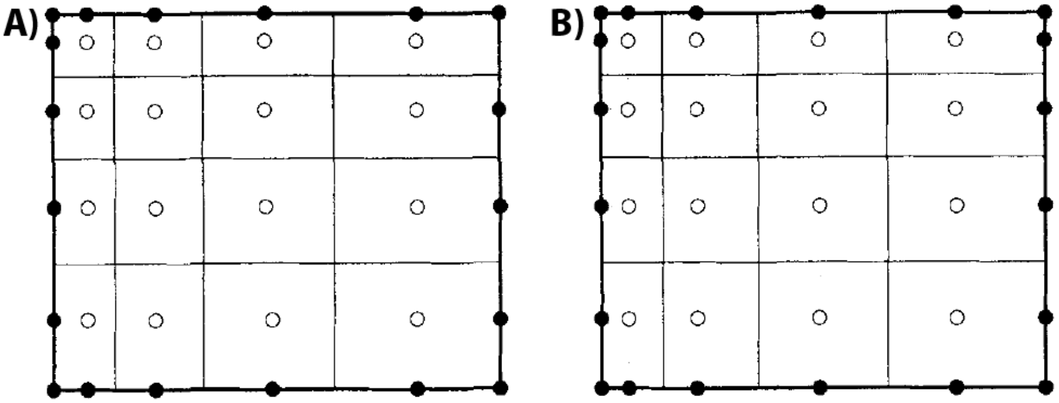


Figure 3.3: Types of Finite Volume grids with **(A)** nodes centered in CVs and **(B)** CV faces centered between nodes, from [18].

For some scalar function f bounded by CV surface S, the surface integral can be decomposed as the sum of all CV cell faces

$$\oint_S f \mathrm{d}S = \sum_k \int_{S_k} f \mathrm{d}S, \tag{3.11}$$

---

[2] A major benefit of FVM is that it can accommodate any type of grid, *i.e.* the selected grid is not required to be structured or uniform. Furthermore, the grid only defines the CV boundaries thus does not need any relation to the local coordinate system. Since control volumes may be arbitrarily shaped, the grid only needs to be well-defined.

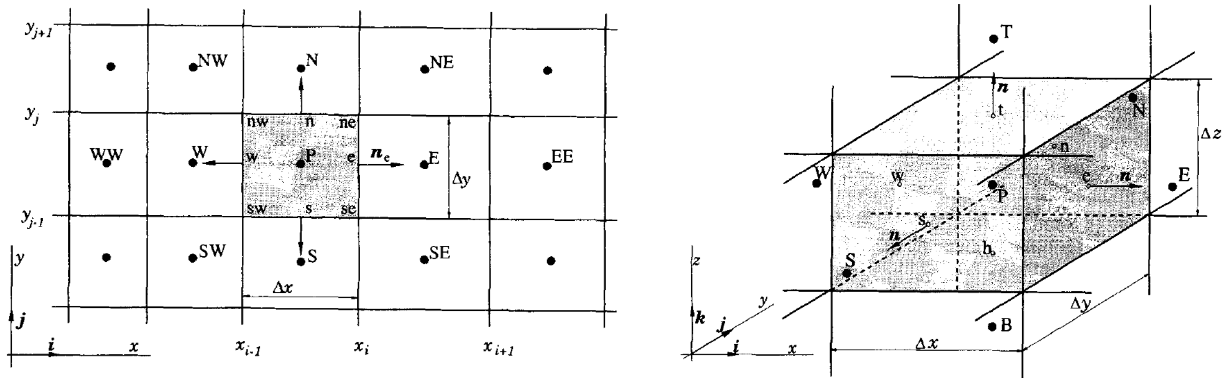
[3] Computational nodes need not necessarily be at the CV center. Instead, the solution domain can be divided by computational nodes, with the CV faces subsequently centered between adjacent nodes (see Figure 3.3B). For simplicity, we will assume that the computational node is selected at the CV center.

where f in fluid dynamics may be selected as the flux vector in the direction normal to each CV face k, denoted by a letter corresponding to their orientation (*i.e.* n, e, s, w, for 2D with t and b as well for 3D, as shown in Figure 3.4). The conservative integral then takes the form of discrete algebraic equations through integral approximations. As an example, using the midpoint rule approximation, each cell face integral in (3.11) is written as

$$\int_{S_k} f \mathrm{d}S = \bar{f}_k S_k \approx f_k S_k, \tag{3.12}$$

for some cell face k. The midpoint rule approximation is of second-order accuracy provided that the value of f_k is known at the cell face center. Since the value of f is typically only known at the computational node, to preserve the second-order accuracy of the midpoint rule, some arbitrary approximation scheme[4] must be selected to compute f_k with at least second-order accuracy. For an example of a higher-order approximation, the cell face integral may be approximated using Simpson's rule with fourth-order accuracy,

$$\int_{S_k} f \mathrm{d}S \approx \frac{S_k}{6}(f_{c_1} + 4f_k + f_{c_2}), \tag{3.13}$$



(A) 2D Orientation **(B)** 3D Orientation

Figure 3.4: Orientiation of a single control volume in Finite Volume Method, from [18].

---

[4] The two most common interpolation schemes are: *upwind* interpolation, which uses forward or backward differences (3.5, 3.6) based on flow direction, and *linear* interpolation, which uses central difference (3.7). These offer first-order and second-order accuracy, respectively. Higher-order schemes fit a polynomial of degree n to achieve $(n+1)$-order accuracy.

Despite its lower accuracy, upwind interpolation is numerically diffusive, inherently satisfying the *boundedness criterion* and producing non-oscillatory (bounded) results [18]. Therefore, hybrid schemes exist which alternate between upwind and linear interpolation [18, 19].

where c_1 and c_2 are opposite corners on the surface S_k. Although seemingly simple, higher-order approximations require evaluating the integrand f at more locations with higher-order accuracy, which becomes increasingly difficult to implement. As such, the midpoint rule is most widely used.

For volume integrals, the simplest approximation with second-order accuracy involves simply multiplying the volume-averaged integrand by the CV volume,

$$\int_V q \mathrm{d}V = \bar{q}V \approx q_P V, \tag{3.14}$$

where subscript P indicates the CV center value (*i.e.* the value at the computational node). Furthermore, this approximation is exact if integrand q is constant or scales linearly within the CV (*i.e.* where $\bar{q} = q_P$). For uniform grids, the volume integral may be approximated with higher accuracy with additional nodal values at the edge face centers.

As an example, starting with the most general conservation equation for some scalar quantity ϕ,

$$\frac{\partial \phi}{\partial t} + \vec{\nabla} \cdot \vec{g}(\phi \vec{u}) = q, \tag{3.15}$$

taking the volume integral of both sides, and applying the divergence theorem, we can write (3.15) as

$$\frac{\partial}{\partial t} \int_V \phi \mathrm{d}V = - \oint_S \vec{g}(\phi \vec{u}) \cdot \mathrm{d}\vec{S} + \int_V q \mathrm{d}V. \tag{3.16}$$

By applying (3.14) and (3.11, 3.12), each cell can be evaluated as

$$\frac{\partial \phi_P}{\partial t} = -\frac{1}{V} \oint_S \vec{g}(\phi \vec{u}) \cdot \mathrm{d}\vec{S} + q_P, \tag{3.17}$$

$$\approx -\frac{1}{V} \sum_k S_k \vec{g}_k(\phi_k \vec{u}_k) \cdot \hat{n}_k + q_P, \tag{3.18}$$

or equivalently for some vector quantity $\vec{u}$, with conserving equation

$$\frac{\partial \vec{u}}{\partial t} + \vec{\nabla} \cdot \vec{\vec{\mathbf{\Pi}}}(\vec{u}) = \vec{g}, \tag{3.19}$$

each cell can be evaluated as

$$\frac{\partial \vec{u}_P}{\partial t} = -\frac{1}{V}\oint_S \overleftrightarrow{\mathbf{\Pi}}(\vec{u}) \cdot \mathrm{d}S + \vec{g}_P, \tag{3.20}$$

$$\approx -\frac{1}{V}\sum_k S_k \overleftrightarrow{\mathbf{\Pi}}_k(\vec{u_k}) \cdot \hat{n}_k + \vec{g}_P. \tag{3.21}$$

If we set $\phi = \rho$, $\vec{g}(\phi\vec{u}) = \rho\vec{u}$, and $g = 0$ in (3.15), we recover the mass continuity equation (2.1), and if we set $\overleftrightarrow{\mathbf{\Pi}} = \vec{u} \otimes \vec{u} - \overleftrightarrow{\boldsymbol{\sigma}}/\rho$ and $\vec{g}$ to body force $\vec{f}$ or $\vec{0}$ in (3.19), we recover the Navier-Stokes momentum equation (2.18).

In FDM, boundary conditions are simply imposed directly by specifying the value or derivative of a parameter at grid points on the domain boundary. In FVM however, boundary conditions are applied by defining the fluxes across CV faces coinciding with the boundary. These fluxes may be specified as constant values or expressed in terms of interior values and other boundary-specific parameters[5].

By using the direct integral forms of conservation equations, FVM is a naturally conserving discretization scheme. It is also significantly easier to implement complex geometries compared to FDM and remains stable with discontinuities [18]. As such, FVM was selected as the discretization method for this study.

3.3 Solving Methods

In CFD, solving methods are algorithms designed to numerically solve systems of equations defined over discretized domains. These methods incorporate strategies for iteratively solving coupled equations simultaneously, while maintaining accuracy and handling convergence criteria to achieve efficient and robust solutions. The goal is to balance efficiency with

---

[5] Typically, inflow boundaries have a fixed convective flux expressed as $\vec{u} \cdot \hat{n}\phi = \text{const}$, where $\vec{u}$ is velocity, $\hat{n}$ is unit normal to the boundary, and ϕ is the quantity in question. Impermeable walls and symmetric planes employ zero-flux conditions, *i.e.* $\vec{u} \cdot \hat{n} = 0$ normal to the boundary. Outflow boundaries are generally less restrictive and often use the zero gradient $\partial_{\hat{n}}\phi = 0$. In some cases however, specific exit values or more complex conditions may be used.

generality, handling a wide range of flow scenarios while minimizing inaccuracies. For all solving methods presented in this section, under-relaxation[6] may be applied at any point where a new value is calculated.

3.3.1 SIMPLE

The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm, developed by Patankar and Spalding in 1972 [20], is an iterative method for decoupling pressure and velocity fields to solve the Navier-Stokes equations. Originally designed for incompressible flows, the typical steps which are repeated until convergence criteria are met, are:

1. Advance to the next iteration $t = t^{n+1}$.
2. Initialize $\vec{u}^{n+1}$ and P^{n+1} using $\vec{u}^{n}$ and P^{n} and apply boundary conditions.
3. Solve the momentum equation to obtain an intermediate velocity field $\vec{u}^{*}$ that does not necessarily satisfy the continuity equation.
4. Solve a pressure correction equation derived from the continuity equation using intermediate velocity field $\vec{u}^{*}$ to correct pressure field P^{n+1}.
5. Correct the velocity field $\vec{u}^{n+1}$ using the newly updated pressure field.
6. Repeat steps 2–5 until the residuals fall below a specified threshold.

For compressible flows, intermediate density ρ^{*} is calculated after solving the momentum equation in step 3. Additionally, the energy equation and other thermodynamic models are solved between steps 3 and 4 to compute temperature or other thermodynamic properties, which are then used to update density ρ^{n+1} using the equation of state in step 5. When applicable, mass fluxes are calculated and corrected along with velocity in steps 3 and 5.

---

[6] *Under-relaxation* is a technique used to stabilize iterative methods at the expense of convergence time by controlling the rate at which variables are updated during each iteration, thus preventing divergence or oscillations. For a discrete variable ϕ, under-relaxation is mathematically expressed as $\phi^{n+1} = \phi^{n} + \alpha\left(\phi^{\text{new}} - \phi^{n}\right)$ [18], where α is the under-relaxation factor, typically between 0 and 1, and ϕ^{new} is the value calculated in the current iteration. A value of $\alpha = 0$ would result in no change in the solution across iterations, while $\alpha = 1$ implies no under-relaxation, meaning the full update is applied.

3.3.2 PISO

The Pressure-Implicit with Splitting of Operators (PISO) algorithm, developed by Issa in 1986 [21], extends the SIMPLE algorithm to allow non-iterative computation of unsteady compressible and incompressible flows. The algorithm consists of two primary steps: the predictor stage and the corrector stage. The steps in a single iteration for incompressible flows are as follows:

1. Apply initial and boundary conditions.
2. **Predictor:** Solve the momentum equation using an initial guess for P^{n+1} to obtain an intermediate velocity field $\vec{u}^*$.
3. **Corrector:** Solve a pressure correction equation to update P^{n+1} and correct the velocity field $\vec{u}^{n+1}$ to satisfy the continuity equation.
4. Repeat the corrector step for a predetermined number of times to account for non-orthogonality and other discretization errors.

The PISO algorithm is particularly well suited for transient flow problems due to its ability to handle unsteady conditions efficiently and with greater stability compared to the SIMPLE algorithm. For compressible flows, all other transport equations are solved and density ρ^{n+1} is calculated after step 3 using the equation of state. When applicable, mass fluxes are calculated at the cell faces in step 2 and corrected in step 3.

3.3.3 Variations of SIMPLE and PISO

Several variations of the SIMPLE and PISO algorithms exist to enhance their performance. The *SIMPLE-Consistent* (SIMPLEC) algorithm is a modification of SIMPLE that reduces the computational time by simplifying the velocity correction step [18]. This simplification involves neglecting certain terms in the momentum equation that contribute less to the overall solution, thereby accelerating convergence. The *SIMPLE-Revised* (SIMPLER) algorithm, introduced by Patankar in 1980 [22], further improves the efficiency of SIMPLE by solving an additional equation for pressure correction before solving the momentum equation. This pre-correction reduces the need for under-relaxation leading to faster convergence. The *PIMPLE* algorithm is a hybrid of SIMPLE and PISO, designed to combine the robustness of SIMPLE with the accuracy of PISO for transient flows [23]. In PIMPLE, an outer SIMPLE loop manages the coupling of pressure and velocity, while an inner PISO loop ensures field corrections. PIMPLE is particularly advantageous in cases where both steady-state and transient behaviours are important, offering a balance between stability and computational efficiency.

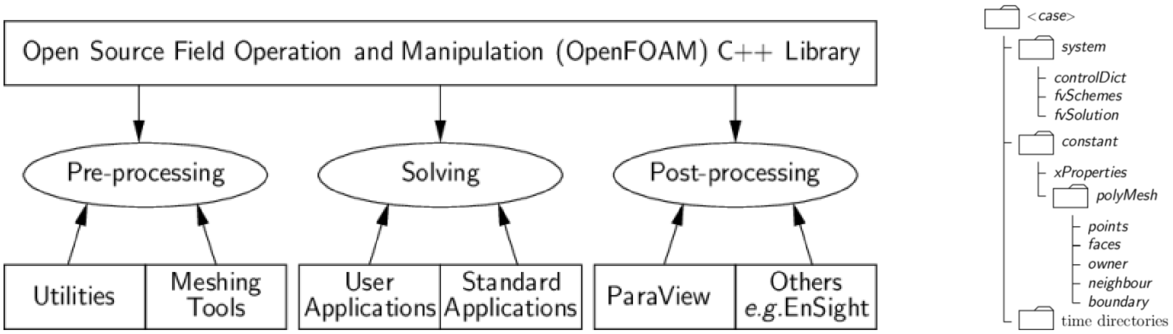
Chapter 4

OpenFOAM

OpenFOAM (Open Source Field Operation and Manipulation) is an open-source CFD framework developed in C++. It enables customization and extension of numerical solvers for simulating fluid flow, turbulence, and heat transfer using FVM. OpenFOAM offers pre-configured solvers and utilities for complex physics simulations.

Originally named Field Operation and Manipulation (FOAM) by Professor Henry Weller and colleagues at Imperial College in the late 1980s, it was created to be more flexible and powerful than commercial CFD software. In 2000, Weller and colleagues formed Nabla Ltd., which later became OpenCFD Ltd., to commercialize their software. OpenFOAM was released under the GNU General Public License (GPL) in 2004, making it one of the first open-source CFD packages available. Its open-source nature allowed users to modify and distribute the software freely, fostering a collaborative community and increasing its popularity in academia and industry.

In 2011, the OpenFOAM Foundation was established to independently manage and maintain the OpenFOAM codebase. In 2012, ESI Group acquired OpenCFD Ltd. to integrate OpenFOAM with its engineering tools and offer commercial support. OpenFOAM then



(A) OpenFOAM overall structure.

(B) OpenFOAM case structure.

Figure 4.1: An overview of OpenFOAM [23].

branched into two main distributions: Foundation OpenFOAM, driven by community development, and ESI-OpenFOAM, managed by ESI Group with additional proprietary features and commercial support. Initially, differences between the two versions were minor, but they have diverged over time. ESI-OpenFOAM now offers more specialized solvers for industry, while Foundation OpenFOAM offers fewer solvers focused on generalization driven by community needs and academic research.

Foundation OpenFOAM v8 was selected for this project along with community-developed swak4foam for the freedom to expand upon a generalized solver with additional boundary conditions and postprocessing calculation functions.

4.1 How it works

OpenFOAM operates as a comprehensive framework for solving complex CFD problems. It provides a wide range of solvers, utilities, and libraries to address different aspects of fluid dynamics and continuum mechanics.

4.1.1 Solvers

OpenFOAM solvers are divided into categories based on the type of continuum mechanics they address. These categories include [23]:

- **Basic flow**: Solvers for simple equations such as Laplace, vector potential, etc.
- **Incompressible flow**: Solvers for both laminar and turbulent flows for Newtonian and non-Newtonian fluids based on Euler's equations.
- **Compressible flow**: Solvers for flows involving changes in density.
- **Multiphase flow**: Solvers for simulating multiple interacting phases.
- **Direct numerical simulations**: Solvers for highly detailed simulations capturing all scales of turbulence.
- **Combustion**: Solvers for reactive flows with chemical reactions.
- **Heat transfer and buoyancy-drive flows**: Solvers for thermal simulations.
- **Particle-tracking flows**: Solvers for tracking particles within fluid flow.
- **Discrete methods**: Solvers for methods like Monte Carlo and Lattice-Boltzmann.
- **Electromagnetics**: Solvers for electromagnetic fields.
- **Stress analysis of solids**: Solvers for structural mechanics and solid deformation.

To closely match the specifics of an experimental setup, customized solvers and boundary conditions can be developed by modifying existing solvers or from scratch.

Exponent	**Property**	**Unit**
a	Mass	kilogram (kg)
b	Length	metre (m)
c	Time	second (s)
d	Temperature	Kelvin (K)
e	Quantity	mole (mol)
f	Current	ampere (A)
g	Luminous intensity	candela (cd)

dimensionSet $[a\ b\ c\ d\ e\ f\ g]$;

Table 4.1: Base dimension unit system of OpenFOAM [23].

4.1.2 Case

An OpenFOAM case is a directory structure containing all the files necessary for a simulation. Split into three main sections, a case contains all configuration files, initial and boundary conditions, solver settings, control parameters, domain geometry mesh, and thermophysical properties.

OpenFOAM files share a similar syntax to C++, with parameters taking the form of dictionaries[1]. A wide range of data structures are supported, such as strings, scalars, vectors, lists, and several macros like `#include`[2] are implemented. OpenFOAM defines units of measurement using the International System of Units (SI), illustrated in Table 4.1, where the first column corresponds to the exponents declared in a file *via* dictionary key `dimensionSet`. As an example, pressure P with units $\mathrm{Pa} = \mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-2}$ would have dictionary entry `dimensionSet` [1 -1 -2 0 0 0 0], whereas kinematic pressure $p_k = P/\rho$ with units $\mathrm{Pa{\cdot}m^3{\cdot}kg^{-1}} = \mathrm{m}^2 \cdot \mathrm{s}^{-2}$ would have dictionary entry `dimensionSet` [0 2 -2 0 0 0 0].

In order to classify all the necessary parameters, the OpenFOAM case is split into three main components: the `system` directory, the `constant` directory, and the `time` directories.

---

[1] In the context of OpenFOAM, a dictionary is a set of key-value pairs that define various simulation settings and parameters. OpenFOAM dictionaries have the following syntax: `dictionary` {`key value;`}.

[2] The `#include` macro allows for on-the-fly text substitution with the contents of another file.

4.1.2.1 The `system` Directory

The `system` directory contains all setting parameters associated with the solution procedure itself. The three mandatory files in the `system` directory are the `controlDict` dictionary, the `fvSchemes` dictionary, and the `fvSolution` dictionary. Several other optional dictionaries for postprocessing may be placed in the `system` directory as well.

The `controlDict` dictionary manages the overall control of the simulation and contains parameters such as which solver will be used (*e.g.* `rhoSimpleFoam`), and time management parameters such as `startTime`, `endTime`, `deltaT`, and output writing parameters like `writeInterval` and `writePrecision`. Furthermore, functions can optionally be defined in the subdictionary `functions` to execute calculations and write outputs at each `writeInterval`, such as Mach number.

The `fvSchemes` dictionary defines all the discretization schemes for spatial and temporal derivatives used in the simulation. The relevant fields are `ddtSchemes` which defines the time derivative scheme, `divSchemes` which defines the divergence schemes, `gradSchemes` which defines the gradient schemes, and `laplacianSchemes` which defines the Laplacian scheme. Schemes are generally specified by the `default` key, *e.g.* `default linear`, but may be specified per operation by adding a key-value pair in the form `operation scheme`. As an example, `divSchemes` entry `div(phi,U) Gauss upwind` specifies that term $\vec{\nabla} \cdot (\vec{u} \otimes \vec{u})$ is discretized using the divergence theorem with upwind interpolation. Steady-state simulating can be explicitly specified by dictionary entry `ddtSchemes { default steadyState;}`, where `steadyState` is replaced by `Euler`[3] for transient flow. The plethora of supported numerical schemes are defined in Chapter 4.5 of [23].

The `fvSolution` dictionary defines all of the numerical solver parameters used in the simulation. The `solvers` field specifies solver parameters for specific fields, such as pressure and

---

[3] The `Euler` temporal discretization scheme is equivalent to the forward difference scheme: $\frac{\partial \phi}{\partial t} \approx \frac{\phi_{t+1} - \phi_t}{\Delta t}$.

velocity. The `SIMPLE`, `PISO`, or `PIMPLE` fields contain settings specific to the chosen pressure-velocity coupling algorithm, including residual convergence criteria via subdictionary entry `residualControl`. Importantly, for transonic simulations, dictionary entry `transonic true;` must be specified. Finally, `relaxationFactors` specifies the optional under-relaxing factors used to stabilize the solution, which are of significance for boundary conditions which cause strong discontinuities from the initial fields[4].

4.1.2.2 The `constant` Directory

The `constant` directory contains all physical properties of the problem, as well as the geometric mesh used in the simulation. All files relating to the mesh are saved in the `constant/polyMesh/` subdirectory. For incompressible flows, `transportProperties` contains physical properties of the fluid such as viscosity and the type of transport used.

For compressible flows, `thermophysicalProperties` contains all thermodynamic and transport properties. Physical models are explicitly defined in the `thermoType` subdictionary. All physical constants are defined in the `mixture` subdictionary, such as fluid-specific properties like molar mass in entry `specie`, thermodynamic properties such as c_p and H_f in entry `thermodynamics`, transport properties specific to the selected model in entry `transport`, and equation of state properties in entry `equationOfState`.

For simulations that support turbulence, an additional dictionary titled `turbulenceProperties` must be defined to specify the turbulence model used. For laminar flows, this dictionary is simply the single entry `turbulenceType laminar`.

---

[4] Relaxation is crucial for CFD simulations, as often only boundary conditions are known and internal fields are initialized arbitrarily (*i.e.* by a constant such as zero), creating strong discontinuities for the first few iterations.

4.1.2.3 The `time` Directories

At every `writeInterval`, a time directory is created, named either *via* the number of iterations (for steady state) or the elapsed time in seconds (for transient). The only required `time` directory is `0`, which contains all initial and boundary conditions of the simulation. The contents of `0` may vary between OpenFOAM solvers, but typically include a pressure file `p`, a velocity file `U`, and a temperature file `T`, which define the values/boundary type at each boundary and the initial field throughout the mesh, which may be initialized to a constant *via* the `uniform` keyword.

The contents of all subsequent `time` directories share an identical format as `0`, with the internal fields being lists of resulting values from the simulation at each write interval. They may also contain additional files, such as `rho` for density based solvers, and other calculated values of functions defined in the `controlDict`.

4.1.3 Mesh Generation

As discussed earlier, geometries in OpenFOAM are defined by several dictionary files located in the `constant/polyMesh` subdirectory. Computational nodes are defined spatially and indexed in the `points` dictionary, while CV faces are defined by sequences of computational nodes in the `faces` dictionary. CVs themselves are represented as owners of these faces in the `owner` dictionary. Neighbouring cells are defined in the `neighbour` dictionary, and mesh boundaries are specified in the `boundary` dictionary, where each boundary is uniquely identified by name, grouped by CV faces, and assigned a physical type. Although these mesh files can be manually created, OpenFOAM provides utilities such as `blockMesh` to assist in mesh generation. For `blockMesh`, a single dictionary, `blockMeshDict`, must be placed in the `system` directory. This file includes the scaling parameter `convertToMeters`, along with entries defining vertices, edges, blocks, and patches.

Additionally, meshes can be converted from various other file formats. OpenFOAM offers

22 utilities for mesh conversion, which are listed in Chapter 3.6.3 and discussed in Chapter 5.5 of [23]. Many of these supported file formats are easier to generate and can be tailored to specific design requirements. However, when using these conversion utilities, any specialized boundary conditions beyond simple patches (such as `symmetric` or `wedge` boundaries[5]) must be manually added in the `constant/polyMesh/boundary` dictionary.

4.1.3.1 Azimuthal Symmetry and Optimization

OpenFOAM offers a variety of optimization techniques to improve the efficiency of simulations, particularly for complex or computationally expensive geometries. One of the most effective ways to reduce computational cost in CFD is to exploit inherent symmetries in the geometry or flow. For the case where a system exhibits rotational symmetry, which is the case with nozzle flows, OpenFOAM's `wedge` boundary patch type can be used to model the full nozzle profile with only a narrow sector.

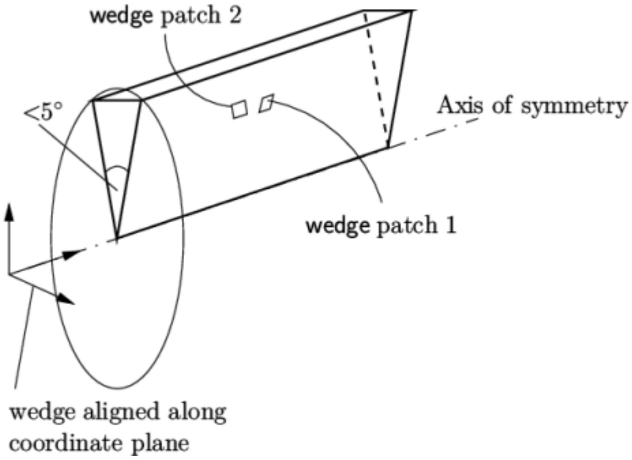


Figure 4.2: Illustration of the OpenFOAM `wedge` boundary patch type for a system with cylindrical symmetry, modified from [23].

The `wedge` boundary patch, as described in the OpenFOAM user guide [23], applies to

---

[5] OpenFOAM supports the following patch types: `patch`, a generic boundary surface with no special conditions; `symmetryPlane`, a plane of symmetry with zero normal gradients; `empty`, for defining the front and back planes of a 2D geometry; `wedge`, for defining the left and right planes for an axi-symmetric (rotationally symmetric) geometry; `cyclic`, for periodic boundary surfaces where flow exits one patch and re-enters through another; `wall`, an impermeable surface with zero normal velocity and specified tangential velocity, important for turbulent flows; and `processor`, which defines inter-processor boundaries for parallel computations.

geometries that are axisymmetric. The wedge sector is defined with two opposing wedge boundaries spanning a small angle, often less than 5°, as shown in Figure 4.2. The angle of the wedge must be small enough to ensure numerical stability and accuracy, as smaller angles minimize the volumetric difference caused near the boundary wall. By only simulating the narrow slice, the computational load is exponentially reduced, as significantly fewer calculations are required. OpenFOAM automatically assigns periodic boundary conditions to the wedge faces, effectively "rotating" the slice, to ensure that the physical properties are accurately applied as though the entire geometry were being simulated.

By utilizing this optimization, the computational domain was reduced from the full 3D geometry to a thin slice, which, combined with parallelization discussed below, cut simulation times drastically. What would normally take several days of computation was reduced to a couple of hours. This reduction in computational cost, discussed in Appendix A, did not sacrifice any accuracy and was a key factor in enabling the completion of over one million subsimulations in a feasible timeframe.

4.2 Modelling Laboratory Experiments

In Guillaume Gervais' lab, former master's student Sam Neale conducted mass flow measurements on nanopores with diameters around 50 nm, drilled into 100 nm thick silicon nitride membranes. As detailed in Neale's thesis [4], the experimental setup consisted of a helium-4 gas reservoir connected to one end of the nanopore, with a mass spectrometer at the other end to measure the flow through the nanopore. Both connections to the nanopore were made *via* steel capillaries submerged in liquid nitrogen, maintaining the helium-4 gas at a fixed temperature of 77 K. The pressure in the reservoir was controlled using valves and monitored with pressure gauges. Measurements involved recording static pressure from the gauges and mass flow from the mass spectrometer after achieving a stable baseline. Pressure was then increased using the control valves and the measurements were repeated until the nanopore broke.

To simulate this experimental setup using CFD, each measurement must be modelled by applying a fixed static pressure at the inlet, corresponding to the pressure gauge reading, and then calculating the mass flow through the nanopore after convergence. Once the simulation reaches a steady state, subsequent simulations are run using the results of the previous simulations as initial conditions with an adjusted inlet pressure, equivalent to increasing pressure *via* control valves. These simulations, termed "subsimulations" and described in sections 4.2.5 and 4.2.6, should start at sufficiently low pressures to ensure accuracy and should incrementally increase pressure to maintain stability.

4.2.1 Selection of `rhoSimpleFoam` solver

The main objective of this study is to model pressure-driven transonic flow. As indicated in Table 2.1, accurately modelling such flows requires a solver capable of handling compressible fluids. The `rhoSimpleFoam` solver in OpenFOAM, specifically designed for steady-state, pressure-driven compressible flows, is ideally suited for this study. The solver uses the SIMPLE algorithm to solve the governing equations of compressible flow (2.24, 2.25). The solver supports transonic flow, as it integrates functionality from the former OpenFOAM solver `sonicFoam`, with transonic behaviour specified in the `fvSchemes` dictionary. Although `rhoSimpleFoam` supports turbulence modeling, the flows in this study are assumed to be laminar, and no turbulence model was selected.

4.2.2 Transport Model

OpenFOAM provides a variety of transport models to describe how physical properties, such as viscosity, vary within a system. Among these, the most commonly used is called

Sutherland transport, originally defined as [24]

$$\frac{\eta}{\eta_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{1 + \frac{C}{T_0}}{1 + \frac{C}{T}}, \tag{4.1}$$

where C is a constant of integration with units K and T_0, η_0 are reference values. By defining the *Sutherland coefficient* A_s as

$$A_s \equiv \eta_0 \frac{T_0 + C}{T_0^{3/2}}, \tag{4.2}$$

and defining *Sutherland temperature* $T_s \equiv C$, we can write (4.1) as

$$\eta(T) = A_s \frac{T\sqrt{T}}{T + T_s} = A_s \frac{\sqrt{T}}{1 + \frac{T_s}{T}}. \tag{4.3}$$

For helium-4, taking Sutherland coefficient $A_s = 1.47458 \times 10^{-6}$ Pa·s/K$^{1/2}$ and temperature $T_s = 80$ K, derived from [7], provides a viscosity profile that is consistent with STP values.

4.2.3 Thermophysical Models

Thermophysical models define how energy, pressure, and density relate. In OpenFOAM, models are selected to describe the equation of state, enthalpy, and thermodynamic values such as heat capacity. A compressibility-based top-level thermophysical model `hePsiThermo`[6] was selected, and a pure mixture consisting solely of helium-4 gas was specified. The equation of state used is that of a perfect gas,

$$\rho = \frac{1}{RT} P = \psi P, \tag{4.4}$$

where ψ is compressibility. Other equations of state, defined in Chapter 7.1.5 of [23], are consistent with the perfect gas model for helium-4 in the temperature range at question when using the values provided by [7].

---

[6] The two top-level thermophysical models in OpenFOAM for fixed composition fluids are `hePsiThermo` and `heRhoThermo`. The `hePsiThermo` model calculates density using compressibility $\psi = (RT)^{-1}$ and is best suited for pressure-driven compressible flows where pressure and temperature are the primary variables. The `heRhoThermo` model, on the other hand, calculates density based on specific enthalpy h and specific internal energy e, making it more appropriate for density-driven compressible flows where density and energy are the primary variables.

4.2.4 Thermodynamic Model

For the thermodynamic model, the heat capacity of helium-4 gas for both constant pressure and volume remain approximately constant as a function of temperature and pressure [7]. OpenFOAM provides the following models for thermodynamics:

- `hConst`: assumes a constant c_p.
- `eConst`: assumes a constant c_v.
- `janaf`: calculates $c_p = R((((a_4T + a_3)T + a_2)T + a_1)T + a_0)$, with additional constants of integration to calculate entropy and enthalpy.

Using the values provided by the NIST-JANAF thermochemical table of [7] with the `janaf` thermodynamic model, between 0 K and 300 K, c_p has a change of merely 2%. As such, `hConst` was selected as the thermodynamic model with supplied specific heat capacity of $c_p = 5188$ J/kg · K.

4.2.5 Boundary Conditions

For compressible pressure-driven flow in OpenFOAM solver `rhoSimpleFoam`, the initial and boundary conditions for pressure, temperature, and velocity must be specified for each simulation.

4.2.5.1 Pressure

The `fixedValue` boundary condition assigns a constant static pressure at the boundary and was selected at the inlet boundary to accurately replicate the experimental apparatus, as pressure gauges measure static pressure. The `zeroGradient` boundary condition assumes that the pressure gradient normal to the boundary is zero, allowing for natural flow development as pressure is calculated by flow in the domain. This boundary condition was selected for the impermeable wall, and is also ideally suited for outlet boundaries where flow is expected to exit the domain freely. However, if the flow becomes underdeveloped, it may prevent a pressure drop across the nozzle from forming, leading to no flow through the nozzle and even possibly infinitesimal inflow. To address this, hybrid boundary condition `freeStream` was used, which imposes `zeroGradient` when outflow occurs, and when no outflow is detected imposes `fixedValue` pressure condition, the value defined by the `freestreamValue` entry. A low pressure of $P_{\text{freestream}} = 1$ Pa is used to “shock” the fluid into the correct behaviour for a single iteration, after which the non-restrictive `zeroGradient` condition resumes.

In the first subsimulation, the internal field was initialized with a pressure of $P_{\text{internal}} = 1$ Pa and an inlet pressure of $P_{\text{inlet}} = 10$ Pa was defined. For the second subsimulation, internal field P_{internal} was determined by the results of the first subsimulation, with the inlet pressure increased to $P_{\text{inlet}} = 50$ Pa. The inlet pressure was then incrementally increased by 50 Pa in subsequent simulations until reaching $P_{\text{inlet}} = 1000$ Pa. Following this, the inlet pressure was increased in increments of 1000 Pa until $P_{\text{inlet}} = 100000$ Pa was reached. At this point, the inlet pressure is increased to $P_{\text{inlet}} = 101325$ Pa $= 1$ atm and increased by 1 atm until the final pressure of $P_{\text{inlet}} = 100$ atm was achieved. This process resulted in a total of 220 subsimulations per full simulation, with 100 data points above 1 atm.

4.2.5.2 Temperature

In experiment, the helium-4 reservoir was submerged in liquid nitrogen. As such, an inlet temperature of $T = 77$ K was enforced *via* the `fixedValue` boundary condition. The `zeroGradient` boundary condition, selected for the impermeable wall and outlet boundary, assumes zero heat flux across the boundary and calculates temperature from the internal domain conditions. The internal field was initialized uniformly with $T = 77$ K, with subsequent subsimulations using the previous result for initialization.

4.2.5.3 Velocity

For pressure-driven flow, the `zeroGradient` boundary condition for velocity allows flow to naturally develop based on the internal domain values, as the normal gradient of the velocity component perpendicular to the boundary is zero, implying that flow can enter/exit the domain without specifying an exact velocity profile. This boundary condition was selected for the inlet and outlet boundaries. At the boundary wall, the `slip` boundary condition was used, as defined in Section 2.1.5.1. The internal field was initialized uniformly to 0.001 m/s along the central axis, several orders of magnitude lower than the expected velocity values to avoid similar zero net flow issues as the pressure boundary conditions.

Optionally, the `pressureInletVelocity` boundary condition may be used at the inlet with the same specified value as the initial internal field, resulting in similar conditions to `zeroGradient` once flow is fully developed. Here, velocity at the inlet is derived from the flux normal to the patch faces, where flux $\phi = \rho\vec{u} \cdot \hat{n}$. However, when pressure is fixed, the flux is corrected by the pressure equation, potentially enhancing convergence time. In testing, the `pressureInletVelocity` only slightly increased convergence time for the initial few data points with no change in final velocity profile. Therefore the less restrictive `zeroGradient` boundary condition was selected for nozzle simulations.

4.2.6 Subsimulations

For each nozzle geometry, 220 subsimulations were conducted, with each representing a distinct simulation at a fixed inlet pressure. These simulations were divided into two segments aimed at progressively increasing the inlet pressure while maintaining system stability. The first segment consisted of 120 data points ranging from near vacuum to atmospheric pressure, using small increments between 50 Pa and 1000 Pa. Once atmospheric pressure was reached, the second segment of 100 simulations followed, each increasing the inlet pressure by 1 atm.

To ensure numerical stability and avoid introducing large pressure gradients, the results from each simulation were used as the initial field values for the next. This process, managed within a single case directory, mirrors the experimental approach described at the start of this section.

Each simulation was allowed to run for up to 10 000 iterations before stopping and writing the results. Pressure file (`time/p`) was then updated by incrementing the inlet pressure for the next simulation, continuing until all subsimulations were executed.

4.3 Python Wrapper

All simulations were executed using a custom Python application developed specifically for this work, serving as the primary framework for managing the entire simulation process. This framework was designed to handle a wide range of input parameters and to automate geometry creation, simulation execution, and postprocessing efficiently.

The framework is composed of several key components:

- A geometry handler for managing geometry input and generation.
- A parameter handler that creates and organizes a list of simulations based on specified geometry ranges and simulation parameters.
- A foam file writer and reader for handling OpenFOAM operations.
- A wrapper to interface with third-party software through system calls.
- A thread manager responsible for parallelizing simulations.

All these components are utilized in the executable file, `runSimulations.py`, which contains a single function called by the thread manager for each simulation. The thread manager generates a comprehensive list of simulation parameters based on the range of geometry inputs and supplies them to the Python function individually. Each simulation took one entry from the parameter list, in the form of a data object containing geometry and boundary conditions, as input. Since each subsimulation is relatively short by design, the parallelization module distributes entire simulations across individual threads, enabling concurrent execution. The HP Z420 quad-core Xeon server used in this study ran up to eight simulations simultaneously, which greatly improved computational efficiency.

For each simulation, the Python function initiates by creating a case directory in volatile memory (`tmpfs`) to minimize read/write delays. It writes the necessary simulation parameters using the foam file reader/writer module and then generates the geometry mesh, described in detail in the following section. Once the boundary conditions are applied, the subsimulations begin. Upon completion, the pressure file (`time/p`) is copied to `time/p.orig`, and a new pressure value is written to the original file before the next subsimulation runs. This loop continues until all pressure values are simulated, at which point the original `p` files are restored.

After all subsimulations have completed, the function performs postprocessing using OpenFOAM's postprocessing utility. This uses a custom control dictionary, `controlDict.Post`, which includes additional functions to calculate key flow properties, such as mass flow and volumetric flow, and records data at the throat of the nozzle. Additional sampling is performed along the central flow axis and across a slice through the central and radial axes for more detailed flow analysis.

Once postprocessing and sampling are complete, the simulation results is read into Python and stored in persistent memory for further analysis. To conserve memory, the case files are removed from volatile memory after the extracted data has been saved. Once done, the Python function returns identifying information to the thread manager, and the next simulation is run on the thread.

Each simulation performed in this study generated roughly 100 MB of raw data, amounting to 11 995 individual files per run. This data was read and organized into five essential files, reducing the total size to about 12 MB per simulation. The largest portion of the data—approximately 11 MB—consisted only of the sampled data in the r-z plane. In total, 6 155 simulations were conducted, consuming 72.9 GB of storage.

4.3.1 Geometry

4.3.1.1 Meshing

Mesh generation was performed using `gmsh` [25]. For each geometry, a `.geo` file was first created in Python, defining the essential nodes and edges. This file was processed by `gmsh` to generate a `.msh` file, which was then converted into OpenFOAM format using the `gmshToFoam` utility. More details on mesh creation may be found in Appendix A.

To accommodate different nozzle geometries, two Python objects were created, both derived from an abstract geometry class. The first, called "variable" geometry, represents nozzles based on a predefined shape, which is scaled and interpolated according to the provided parameters. The second, called "opening angle" geometry, constructs nozzles linearly using the given geometric parameters. This flexible framework allows the geometry handler in the Python application to work with various geometry types by accepting specific parameters and calculating the radius $r(z)$ for any given axial position z.

Furthermore, the abstraction allows for the integration of additional geometry types beyond variable and opening angle geometries, as long as the calculating function $r(z)$ is defined. These geometries can take any parameters provided by the top-level thread manager.

4.3.1.2 Variable Geometry

Variable geometries are defined by files[7] that contain r and z values, representing the nozzle shape on the continuous interval $0 \leq z \leq 1$. The minimum value of r corresponds to the throat, with cubic spline interpolation used to define continuous function $r(z)$. These geometries need not be symmetric, although all samples in this study were. For each simulation, key parameters such as the inner radius R_{in}, outer radius R_{out}, length L, and throat position z_{throat} were specified. The function $r(z)$ was then scaled to the range $0 \leq z \leq L$, with R_{in} located at $z = z_{throat}$ and R_{out} at $z = 0$ and $z = L$. This scaling technique was applied to interpolate the converging and diverging sections of the nozzle. The function was then translationally shifted for the throat to occur at $z = 0$ before being written. The study included two continuously defined shapes based on arctangent, parabolic, and conic functions, along with two experimental geometries (samples 1 and 2) obtained from TEM measurements reported in [4]. The purpose of the variable geometry type is to perform direct simulations with a predefined nozzle shape.

4.3.1.3 Opening Angle Geometry

In contrast to variable geometries, the converging and diverging sections of the opening angle geometries are defined by straight lines. Key parameters, including R_{in} or R_{out}, opening angle α, length L, and throat position z_{throat}, were specified for each simulation. The same scaling methodology was implemented to compute the corresponding $r(z)$ function, ensuring a consistent approach across different geometry types.

Finally, it is important to note that for opening angle geometries, only one of R_{in} or R_{out} may be supplied, as they are related to each other *via* opening angle α and throat position z_{throat}. In this study, both inner radius and outer radius simulations were performed, with throat positions varied along the entirety of the nozzle.

---

[7] The sample files used in this study were obtained *via* experiment or by sampled points from a continuously defined function.

Chapter 5

Simulations of Laboratory Experiments and Results

5.1 Model Validation

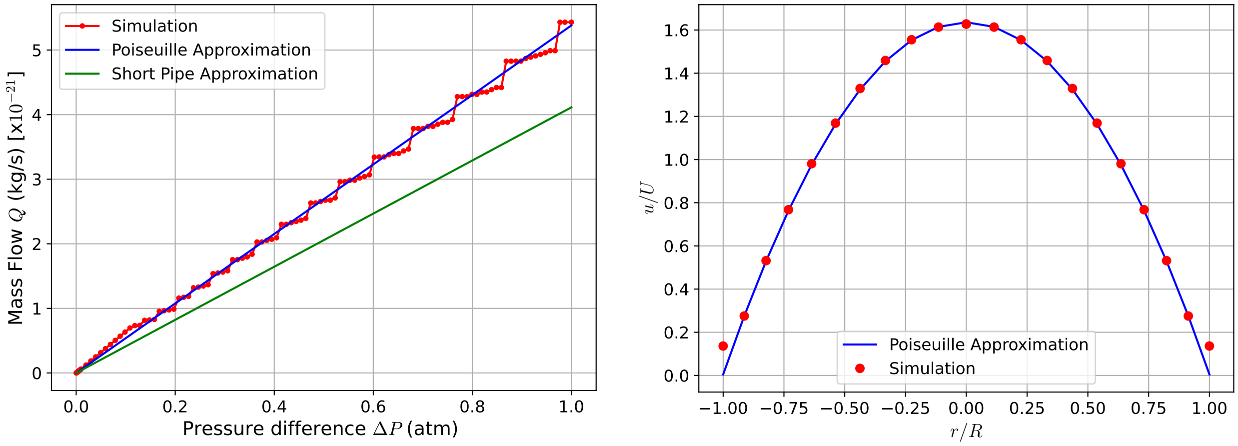
To validate the simulation framework and ensure the reliability of OpenFOAM solver `rhoSimpleFoam` used in this study, validation was performed against established analytical solutions in various conditions.

5.1.1 Poiseuille Approximation

The first validation test was the Poiseuille approximation, a classical benchmark for laminar flow, described in Section 2.1.3. This scenario provides an analytical solution for the velocity profile and mass flow rate, provided the no-slip boundary condition is employed. While the simulations of converging-diverging nozzles presented in this study employ a slip boundary condition, the analytical derivation of the Poiseuille solutions explicitly assume no-slip at the boundary walls to determine the parabolic velocity profile and associated mass flow rate. Deviations from this assumption would result in solutions that no longer conform to the analytical standard.

Despite this difference in boundary conditions, the use of no-slip condition remains a valid test for the simulations presented later in this work. This is because the simulations of converging-diverging nozzles operate within the same Knudsen slip regime, ensuring that slip effects at the wall are minimal. Viscosity, which dictates the flow profile in the Poiseuille approximation, still plays a significant role in determining the flow dynamics of a converging-diverging nozzle, due to the shear stresses caused by the deformation from changing cross-sectional areas. The bulk fluid motion is described by the Navier-Stokes equation, which includes the viscous term $\nu\vec{\nabla}^2\vec{u}$. Near the CD nozzle throat, steep flow gradients amplify viscous effects. Here in a straight pipe, the same governing equations apply, but with shear stress originating from the outer boundary rather than deformation.

While the Poiseuille approximation is inherently defined for incompressible flow, which assumes negligible density variations, this assumption is only valid as long as $M \lesssim 0.3$, as shown in Section 2.1.5. For higher velocities where the incompressible approximation no longer holds, the analytical framework must be adjusted. Specifically, the Poiseuille-



(A) Mass flow Q vs pressure difference ΔP. **(B)** Radial velocity profile at $\Delta P = 1$ atm.

Figure 5.1: Poiseuille long-pipe approximation. A tube of radius $R = 20$ nm and length $L = 1000$ nm was simulated up to $\Delta P = 1$ atm. The velocity profile remained constant throughout the simulated region.

derived expressions can be modified to account for compressibility, as shown in Appendix B.1. Importantly, the compressible solution can be written in an identical form as the incompressible solution, by taking the cylinder-averaged density instead of the fluid reservoir density.

A total of 10 simulations were performed with a cylinder length of $L = 1000$ nm and radii ranging from $R = 20$ nm to $R = 200$ nm. Although all simulations converged, most subsimulation reached the maximum number of iterations and therefore contributed to noisy data, which manifested as steps in the results which appear to grow with the pressure differences. Since the geometry remained unchanged throughout the simulation region, a fixed pressure difference was necessary to maintain consistency. This necessitated the use of an additional `fixedValue` boundary condition at the outlet. As such, these steps likely stem from the solver's difficulty in resolving the steep gradients forcibly induced by the boundary conditions at each pressure step until the difference becomes sufficiently high.

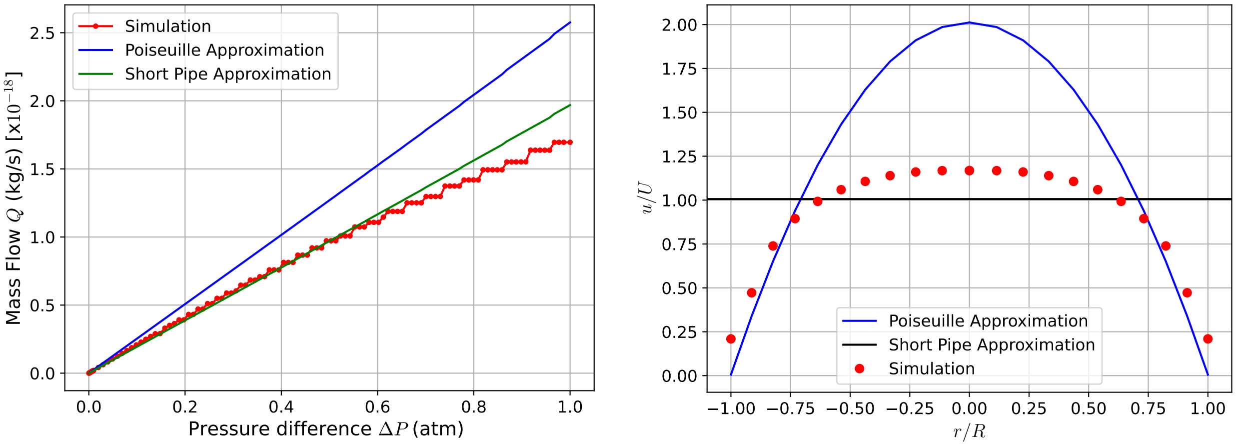
Despite the aforementioned challenges with the solver, all simulations exhibited the expected behaviour of Poiseuille's solution. The simulation selected to present, with radius $R = 20$ nm, had the maximum simulated ratio of cylinder length to radius of $\frac{L}{R} = 50$, well within the valid region defined in Section 2.1.3.1. As shown in Figure 5.1, the simulation yielded exact matches within noise to Poiseuille's solution with the mass flow profile in panel (A) matching equation (2.41) and the velocity profile in panel (B) matching equation (2.39). The only discrepancy occurs at the boundary and is most likely due to utilizing the `wedge` patch types described in Section 4.1.3.1, which places the boundary wall slightly outside the simulation region.

A maximum Mach number of $M = 0.06$ was reached at $\Delta P = 1$ atm, thus the incompressible condition holds. The limit in velocity was due to the no-slip assumption of Poiseuille flow. It is also worth noting that for cases involving slip flow, as discussed in Appendix B.2, an analytical solution for mass flow remains possible by incorporating the boundary condition involving the slip length defined in Section 2.1.5.1. This length, defined by molecular and geometric constraints, adjusts the boundary condition to account for wall effects. Importantly, setting the slip length to zero in these equations recovers the no-slip condition employed, resulting in the same mass flow as Poiseuille's solution. Thus, the no-slip validation retains relevance even for flows with slip effects.

5.1.2 Short Pipe Approximation

Following the validation of Poiseuille flow, the simulation framework was tested with Langhaar's short-pipe approximation defined in Section 2.1.4. This configuration provides a benchmark for underdeveloped flow, deviating from the idealized parabolic profile of Poiseuille flow while maintaining a simplified flow regime suitable for analytical comparison.

A total of 20 simulations were conducted at a fixed cylinder length of $L = 100$ nm, spanning a range of radii from $R = 10$ nm to $R = 200$ nm. Of these, only the first



(A) Mass flow vs pressure drop.

(B) Radial velocity profile where $\Delta P = 0.1$ atm.

Figure 5.2: Short-pipe approximation. A tube with radius $R = 50$ nm and length $L = 100$ nm was simulated up to $\Delta P = 1$ atm. Unlike the fully developed Poiseuille flow, the velocity profile in this short-pipe geometry varied throughout the simulated region. The profile shown in Fig. 5.2B was taken at the outlet with a pressure difference of $\Delta P = 0.1$ atm, which remained within the incompressibility limit.

five simulations successfully converged up to a pressure difference of $\Delta P = 1$ atm. The lowest L/R ratio achieved within the valid short-pipe range was $L/R = 2$, corresponding to $R = 50$ nm.

In the $R = 50$ nm case, the Mach number M exceed 1 at a pressure difference of $\Delta P = 0.65$ atm. However, the incompressibility condition $M \lesssim 0.3$ was satisfied for pressure differences below $\Delta P = 0.15$ atm, at which point nonlinearities in mass flow appeared, as shown in Figure 5.2B. To ensure reliable comparisons with Langhaar's model, the velocity analysis was conducted at $\Delta P = 0.1$ atm, where $M = 0.2$, comfortably below the compressible flow threshold.

Although Langhaar's short-pipe model does not explicitly provide a velocity profile, the predicted average velocity u_{avg}, as given in equation (2.44), was found to be consistent with the area-weighted average simulated velocity U. The simulated U was calculated as

$$\begin{aligned} U &= \frac{\sum_{i=1}^{n-1} \pi \left(r_{i+1}^2 - r_i^2\right) \left(u_{i+1} + u_i\right)/2}{\pi R^2}, && (5.1) \\ &\approx \frac{\sum_{i=1}^{n} \pi \left[(r_i + \frac{\Delta r}{2})^2 - (r_i - \frac{\Delta r}{2})^2\right] u_i}{\pi R^2} = \frac{\sum_{i=1}^{n} \left(2\pi r_i \Delta r\right) u_i}{\pi R^2}, && (5.2) \end{aligned}$$

where r_i represents the radial position of the i-th point, u_i is the velocity at r_i, R is the pipe radius, and Δr is the radial spacing of data points, assumed uniform for the approximate calculation neglecting higher-order terms in (5.2). The predicted and simulated average velocities were compared by calculating their ratio, shown as the black horizontal line in Fig. 5.2B. Using the higher-accuracy equation (5.1), the ratio was found to be $u_{avg}/U = 1.0053$, an acceptable difference of ~0.5%. This small deviation demonstrates the solver's reliability for underdeveloped flow. For comparison, the ratio was computed as $u_{avg}/U = 1.0096$ using equation (5.2), which assumed uniform point spacing. While both results demonstrate <1% agreement with the predicted value, the calculation presented in Fig. 5.2B offers greater precision at a higher computational cost. In both cases, uncertainty is related to point spacing, as equations (5.1, 5.2) are approximations of the integral $\langle f(r,\theta) \rangle = \frac{1}{A} \iint_A f(r,\theta) \mathrm{d}A$.

5.2 Geometric Effects and Data Matching

Building on simulations designed to replicate laboratory conditions, this section focuses on comparing previous experimental data to a wide array of simulations to identify the best match for each experiment. The dataset used for comparison consists of 6064 full-profile simulations, each comprising of 220 sub-simulations (*i.e.* data points) spanning the predefined range of inlet pressures. These simulations differ by variations in critical geometric parameters, enabling an effective search for profiles that closely matched experimental measurements in a diverse parameter space.

To perform data matching, the simulated mass flow as a function of inlet pressure was interpolated *via* cubic splines, creating a smooth univariate representation. In the case where flow was underdeveloped, *i.e.* prior to the choking point[1], missing data points were extrapolated using a least-squares linear fit based on the developed flow region. This allowed for the evaluation of simulated mass flow at the discrete inlet pressures recorded during the experiments. The quality of each simulation's match to experimental data was quantified using a scoring system designed to account for both the magnitude and the sign of deviations. The score was computed as

$$\text{Score} = \left| \sum_i \text{sgn}(x_i - y_i) \right| \times \sum_i |x_i - y_i|, \tag{5.3}$$

where x_i and y_i represent the experimental and simulated results at a given pressure P_i. This scoring methodology favored simulations that consistently aligned with experimental data across multiple points or deviated minimally. Furthermore, it penalized profiles that exhibited systematic offsets or consistently overshot/undershot the experimental results. Notably, this approach also mitigated the influence of outlier behaviour at high inlet pressures, where experimental data deviated from the linear trends observed at lower pressures.

---

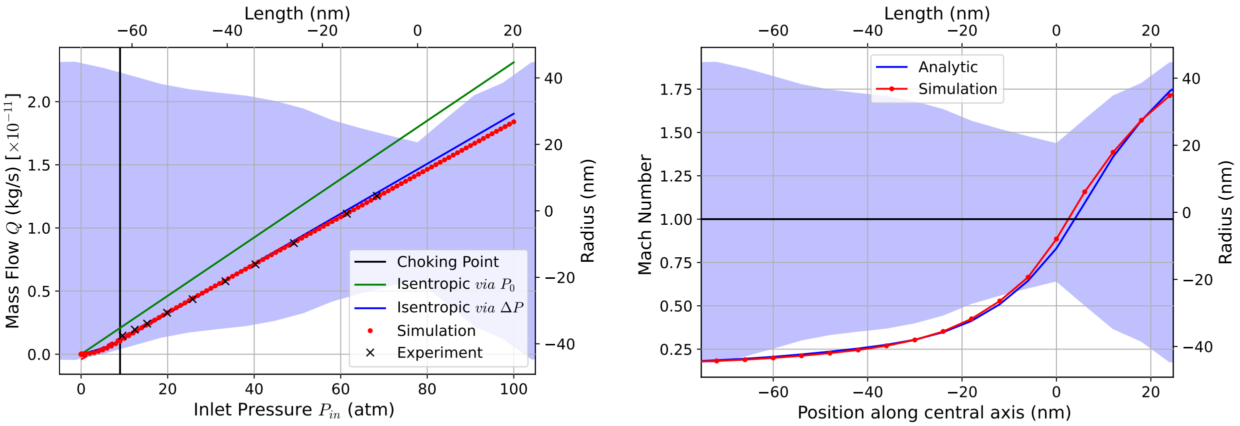
[1] The choking point occurs when the ratio of inlet to downstream throat pressure, referred to as the sonic pressure P^* in Chapter 2, satisfies the condition defined in equation (2.84), *i.e.* when $P^*/P_{in} \approx 0.487$ for monatomic helium-4 gas. Since downstream pressure decreases during flow acceleration, $P_{out} \leq P^* \leq P_{in}$, this condition is equivalently expressed as $P_{out}/P_{in} \leq 0.487$.

The results of this matching process are presented in the subsequent subsections, where specific profiles are analyzed in detail. By identifying simulations that best capture the experimentally observed mass flow behaviour, the critical role of geometric effects on acoustic horizon formation can be studied accurately.

5.2.1 Variable Nozzle Geometries

This subsection presents the simulation results for variable nozzle geometries and their comparison to experimental data. The input parameters defining these geometries include inner radius R_{in}, outer radius R_{out}, and the throat position along the central axis. The nozzle length L was fixed at 100 nm, corresponding to the thickness of the silicon nitride substrate used in experiments. Opening angles were calculated post-simulation, defined as the angle formed between the throat and the outlet. Arbitrary nozzle shapes were parameterized using cubic splines, scaled appropriately, and discretized. For each combination of R_{in} and R_{out}, the throat position was varied across 20 evenly spaced points along the discretized central axis, generating a comprehensive dataset. Simulations were conducted for all geometries with R_{in} ranging from 20 nm to 150 nm and R_{out} ranging from 30 nm to 200 nm. Among the simulated geometries were representations of two experimentally imaged samples, Sample 1 and Sample 2 of [4], along with generic shapes such as an arctangent, parabolic, and a conic profiles. A total of 3710 simulations were successfully performed.

For Sample 1, the best-matched simulation corresponded to its experimentally TEM imaged geometry, with an inner radius of $R_{in} = 20$ nm and an opening angle of $\alpha = 38.9°$. This simulation was also the closest to the experimentally determined geometric values simulated, which were measured *via* low-pressure conductance tests, and yielded an inner radius of 19.25 nm and an opening angle of $\alpha = 40°$. This represents a difference of only 3.75% and 2.75% for the inner radius and opening angle, respectively. The simulated nozzle had an outer radius of $R_{out} = 45$ nm and the throat position was 75 nm away from the inlet. The



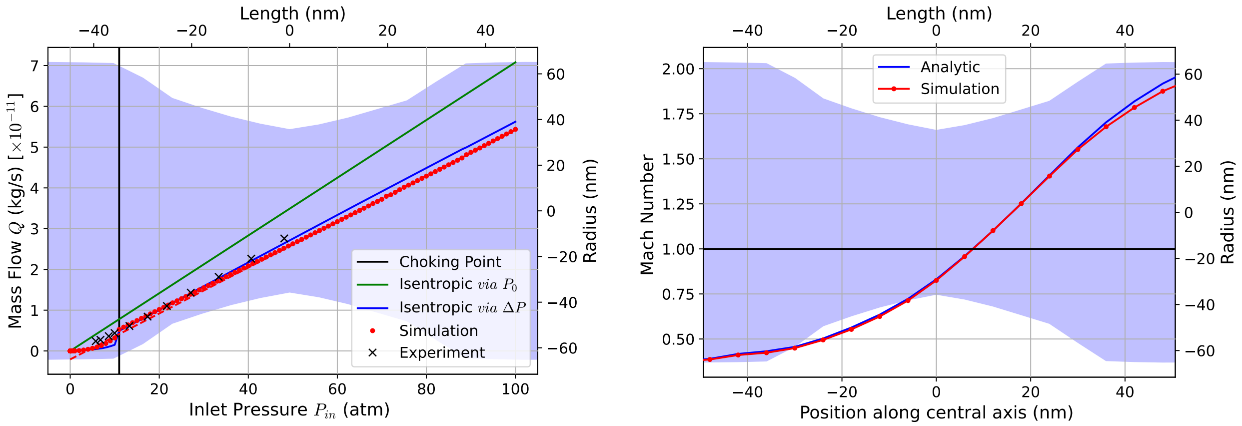
(A) Mass flow vs inlet pressure. **(B)** Radial velocity profile along central axis.

Figure 5.3: Simulation which best-matched the experimentally obtained mass flow profile for Sample 1. In **(A)**, the mass flow is compared with the predicted values given by equation (2.93), with a black line marking the first point where the choking condition is satisfied. In **(B)**, the velocity profile, extracted along the central axis at the highest pressure obtained experimentally, is compared to equation (2.82). The shaded regions in both plots represent the nozzle profile, with the top and right axes corresponding to its dimensions.

experimentally measured geometry for Sample 1 relied on electron intensity reconstruction, capturing only the inlet side of the pore. The geometry presented in [4] mirrored the pore to approximate its full structure, as the sample was unfortunately not flipped to image the outlet. By extracting data points from the reconstructed geometry, the lowest radius measured was approximately 72 nm away from the substrate surface, which is consistent with the simulation's throat position within 4% difference.

The velocity profile derived from equation (2.82) using the nonlinear pressure distribution within the nozzle showed excellent agreement with the simulated velocity profile along the central axis, shown in Fig. 5.3B, strongly corroborating the isentropic relation. However, the mass flow profile predicted by the isentropic relation given by equation (2.93) deviated from both the simulated and experimental results, as shown in green in Fig. 5.3A, by the same factor observed experimentally, *i.e.* reaching only 85.6% of the choked flow value.

For Sample 2, the best-matched simulation also corresponded to its experimentally imaged geometry, with an inner radius of $R_{in} = 35$ nm and an opening angle of $\alpha = 28°$.



(A) Mass flow vs inlet pressure.

(B) Radial velocity profile along central axis.

Figure 5.4: Simulation which best-matched the experimentally obtained mass flow profile for Sample 2. In **(A)**, the mass flow is compared with the predicted values given by equation (2.93), with a black line marking the first point where the choking condition is satisfied. In **(B)**, the velocity profile, extracted along the central axis at the highest pressure obtained experimentally, is compared to equation (2.82). The shaded regions in both plots represent the nozzle profile, with the top and right axes corresponding to its dimensions.

Experimentally determined values were $R_{in} = 33.8$ nm and $\alpha = 25^\circ$, with the latter derived as the average of 20° and 30° reported in [4]. This represents a difference of 3.4% for the inner radius, which was also the closest geometric match simulated. The opening angle differed by 10.7%, though this remains consistent with the range of variation presented in earlier work. The simulated nozzle had an outer radius of $R_{out} = 65$ nm, and the throat position was exactly in the middle of the nozzle, *i.e.* 50 nm away from both the inlet and outlet. The experimentally measured geometry for Sample 2 was also reconstructed using electron intensity, capturing only the inlet side of the pore. As with Sample 1, the geometry presented in [4] mirrored the inlet to infer the full structure. From the reconstructed data, the minimum radius of the nanopore was measured approximately 50 nm from the substrate surface, aligning with the midpoint of the 100 nm-thick substrate. Remarkably, the best-matched simulation captured this geometric feature.

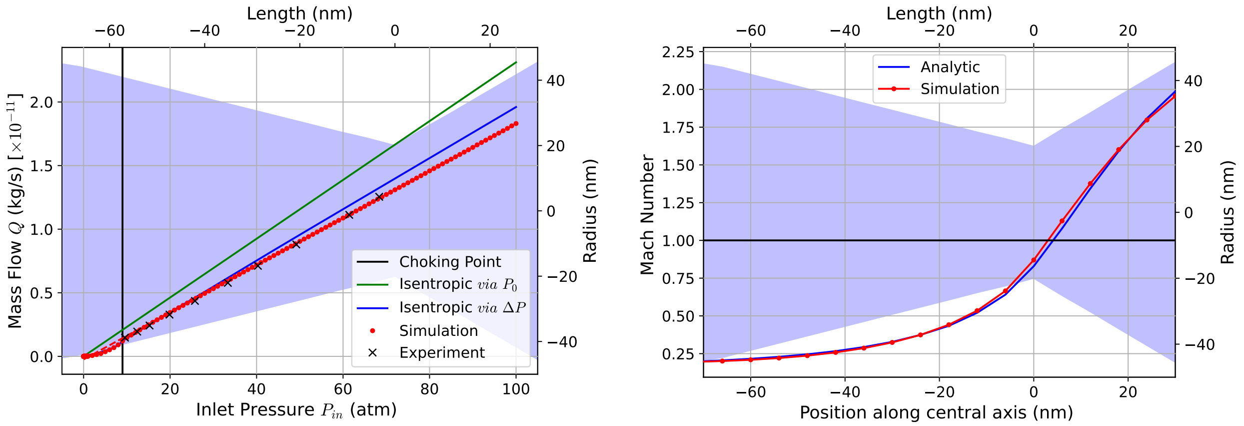
As with Sample 1, the velocity profile for Sample 2, derived from equation (2.82) and calculated using the pressure distribution within the nozzle, closely matched the simulated velocity profile along the central axis, shown in Fig. 5.4B. However, the predicted mass flow profile deviated from both the simulated and experimental results, shown in green in Fig. 5.4A. Notably, the final two experimental data points diverged from the simulated profile, likely due to nozzle deformations occurring prior to structural failure. As with Sample 1, the mass flow profile predicted by the isentropic relation deviated from both the simulated and experimental results, reaching only 87.3% of the choked flow value.

Interestingly, for both samples, when the pressure drop ΔP across the nozzle was used directly as input for the mass flow calculation, the isentropic relation (coloured blue) aligned closely with both the simulated and experimental mass flow rates. This alignment, while potentially coincidental, warrants further investigation and is discussed in Appendix C. In both cases, the choking point—where the acoustic horizon forms—was identified and is marked by a vertical black line in Figs. 5.3A and 5.4A.

5.2.2 Opening Angle Simulations

This subsection presents the results of the opening angle simulations and their comparison to experimental data. The input parameters required for this simplified model is either inner radius R_{in} or outer radius R_{out}, opening angle α, and the throat position along the central axis. For this geometry, a closing angle may be provided instead of α, adding variance to the calculable opening angle. Of the 2354 opening angle simulations performed successfully, 639 of them were initiated by providing a closing angle instead. Like the variable geometry simulations, nozzle length L was fixed at 100 nm, corresponding to the thickness of the silicon nitride substrate used in experiments. In the cases where outer radius was given, inner radius was calculated post-simulation. For each combination of radii and angle, the throat position was varied across 20 evenly spaced points along the discretized central axis.

For Sample 1, the best-match simulation had an inner radius of $R_{in} = 20$ nm and an opening angle of $\alpha = 35°$. While the inner radius was the same as the variable geometry



(A) Mass flow vs inlet pressure.

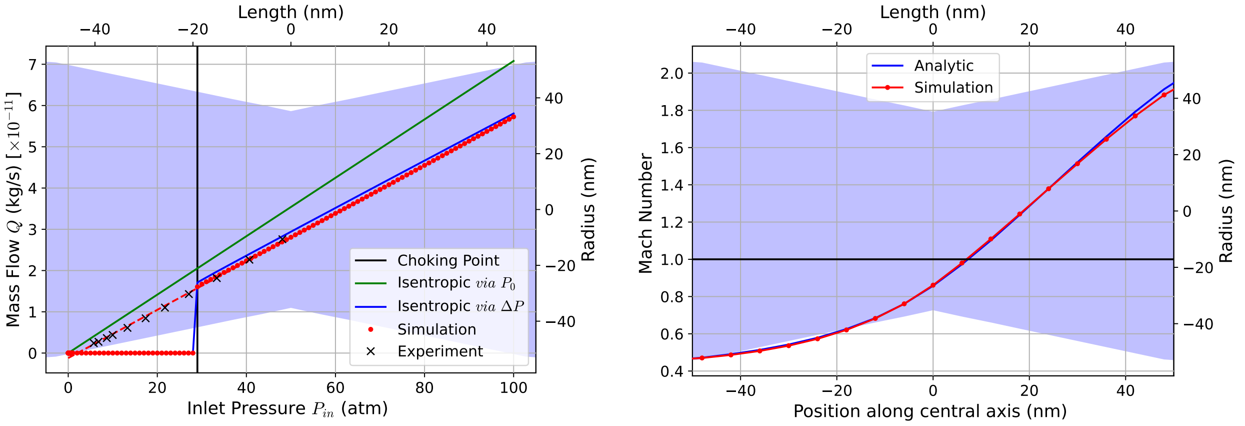
(B) Radial velocity profile along central axis.

Figure 5.5: Opening angle simulation which best-matched the mass flow profile experimentally obtained for Sample 1. The shaded regions in both plots represent the nozzle profile, with the top and right axes corresponding to its dimensions. The nozzle is remarkably similar to the variable geometry simulation match shown in Fig. 5.3.

simulation, the opening angle deviated further from the experimentally determined value of $\alpha = 40°$. The simulated nozzle had an outer radius of $R_{out} = 45.5$ nm, and the throat offset was 70 nm away from the inlet. The maximum speed reached nearly Mach 2 at the outlet, which was higher than the maximum speed of around Mach 1.75 reached in the variable geometry simulation. The mass flow and velocity profiles are shown in Fig. 5.5.

For Sample 2, the best-match simulation had an inner radius of $R_{in} = 35$ nm and an opening angle of $\alpha = 18.6°$. Post-simulation calculations resolved an outer radius of $R_{out} =$ 53 nm, and like the variable geometry match, the throat position is precisely at the midpoint of the nozzle. The mass flow and velocity profiles are shown in Fig. 5.6. While the opening angle of the matched simulation deviates significantly from the measured opening angle and the opening angle matched in the variable geometry simulations, the experimental Sample 2 geometry is nonlinear, and the opening angle was defined from throat to the outlet. If we focus on the vicinity of the throat, the opening angle of the immediately downstream region of the best-matched variable nozzle throat is 19.1°, which represents a difference of merely 2.6%. However, flow choking occurred much later in this simulation, around 30 atm, casting doubt on its accuracy below 30 atm.

While this simplified model can be employed to estimate inner radius R_{in}, to accurately determine opening angle, more geometric information is required.



(A) Mass flow vs inlet pressure.

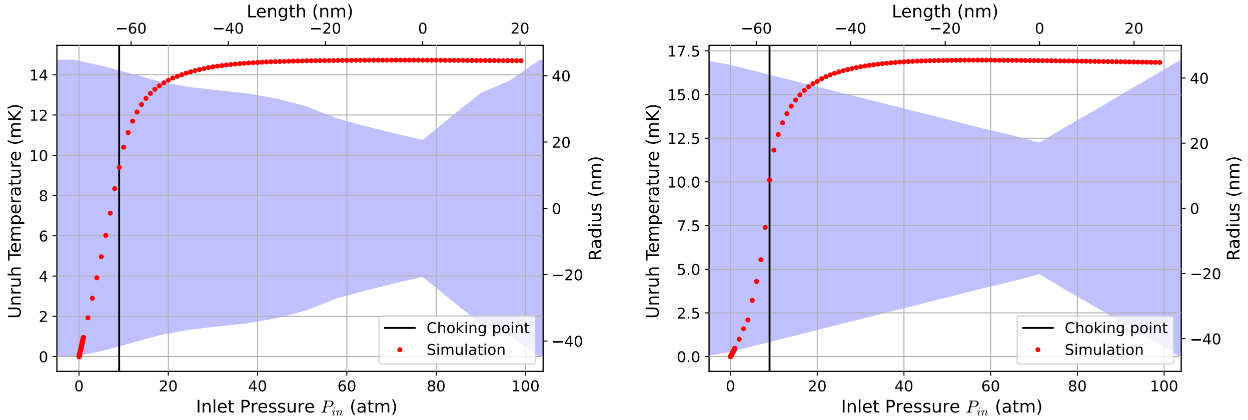
(B) Radial velocity profile along central axis.

Figure 5.6: Opening angle simulation which best-matched the mass flow profile experimentally obtained for Sample 2. The shaded regions in both plots represent the nozzle profile, with the top and right axes corresponding to its dimensions. When compared with the result of the variable geometry match shown in Fig. 5.4, the nozzle is consistent around the throat but largely deviates near the inlet/outlet. Furthermore, the choking condition occurs at a significantly higher pressure.

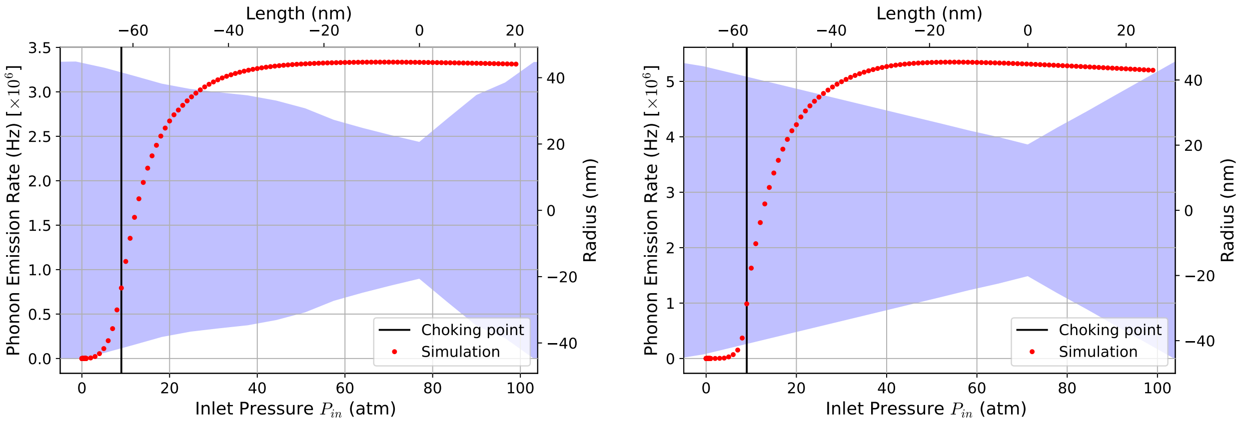
5.3 Horizon Forming and Phonon Emissions

This section presents the phonon emission rates at the sonic horizon and the associated Unruh temperatures for both variable geometry and opening angle simulations. These results depend on the velocity gradient at the throat, $\frac{\partial u_z}{\partial z}$, approximated using the central difference scheme given in equation (3.7). The calculations assume phonon radiation behaves as a perfect blackbody, using equation (2.110) with emissivity $\varepsilon = 1$ for phonon emission rates and equation (2.107) for Unruh temperatures.

For the variable geometry simulations, Sample 1 exhibited an initial phonon emission rate of 0.8 GHz with a corresponding Unruh temperature of 9.4 mK at choking pressure



(A) Variable geometry Unruh temperature. **(B)** Opening angle Unruh temperature.



(C) Variable geometry phonon emission rate. **(D)** Opening angle phonon emission rate.

Figure 5.7: Phonon emission rates and corresponding Unruh temperature for experimental Sample 1, for both variable geometry and opening angle simulations. The shaded regions in all plots represent the nozzle profile, with the top and right axes corresponding to its dimensions.

$P_{in} = 9$ atm. The emission rate increased rapidly with pressure, reaching within 1% of its maximum value at $P_{in} = 45$ atm. The maximum phonon emission rate of 3.3 GHz and corresponding Unruh temperature of 14.7 mK were observed at $P_{in} = 69$ atm, with these values persisting up to $P_{in} = 100$ atm despite a slight decreasing trends. In Sample 2, the phonon emission rate was 2.3 GHz with a corresponding Unruh temperature of 8.9 mK at choking pressure $P_{in} = 11$ atm. The rate increased steadily with pressure, peaking at 2.9 GHz and 9.5 mK at $P_{in} = 25$ atm. Beyond this point, both quantities declined steadily, reaching 2.1 GHz and 8.6 mK by $P_{in} = 100$ atm, as shown in Fig. 5.8C. The pronounced decrease may result from compressibility effects or flow geometry near the horizon, though it remains unknown whether these trends influence the measurable thermal properties of the horizon or are artifacts of the numerical simulation. This discrepancy warrants further investigation.

In the simplified opening angle simulations, the phonon emission rates and Unruh temperatures deviated significantly from the variable geometry models, despite good agreement with the mass flow profiles and geometric properties shown earlier. For Sample 1, the phonon emission rate was 1.0 GHz with a corresponding Unruh temperature of 10.1 mK at the choking pressure of $P_{in} = 9$ atm. Similar to the variable geometry simulation, the emission rate increased rapidly, peaking at 5.3 GHz and 17 mK at $P_{in} = 55$ atm. Beyond this point, both quantities decreased slightly to 5.2 GHz and 16.8 mK, respectively, by $P_{in} = 100$ atm, shown in figures 5.7D and 5.7B. The emission rate reached within 1% of its maximum by $P_{in} = 45$ atm. Notably, the phonon emission rate in this simplified model was 60% higher than in the variable geometry simulation. For Sample 2, the maximum phonon emission rate of 1.96 GHz occurred at the choking point, $P_{in} = 29$ atm, accompanied by a peak Unruh temperature of 8.6 mK. Both values declined significantly, reaching 1.37 GHz and 7.6 mK at $P_{in} = 100$ atm, shown in figures 5.8D and 5.8B. These reductions represent average differences[2] of 35% for the phonon emission rate and 12% for the Unruh temperature. Comparatively, the variable geometry model for Sample 2 showed average differences of 32% and 10% for these quantities, respectively.

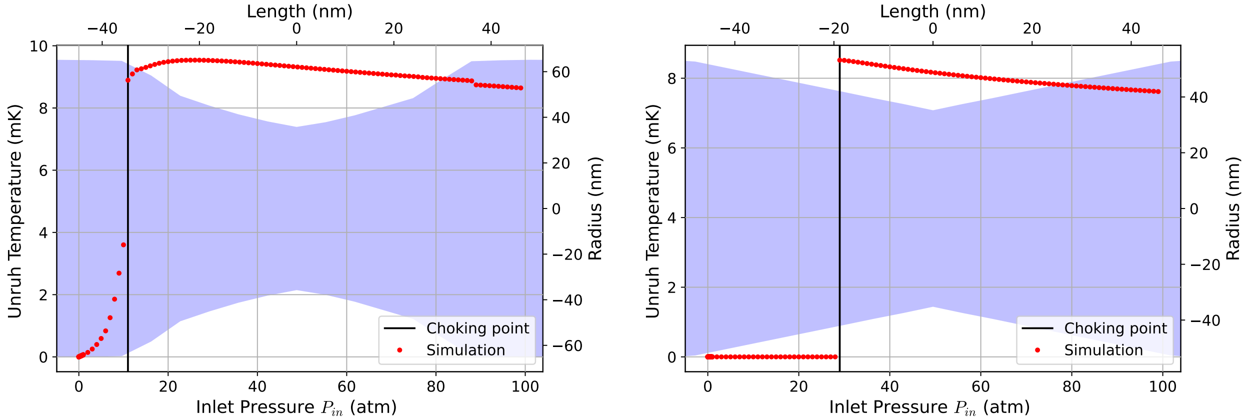
The opening angle simulation for Sample 1 overestimated phonon emission rates relative to the variable geometry, whereas for Sample 2, achieved only 68% of the maximal phonon emission rate seen in the variable geometry simulation. By $P_{in} = 100$ atm, the simplified model produced just 65% of the variable geometry rate. These differences illustrate the

---

[2] Defined as $\frac{|x-y|}{(x+y)/2}$.

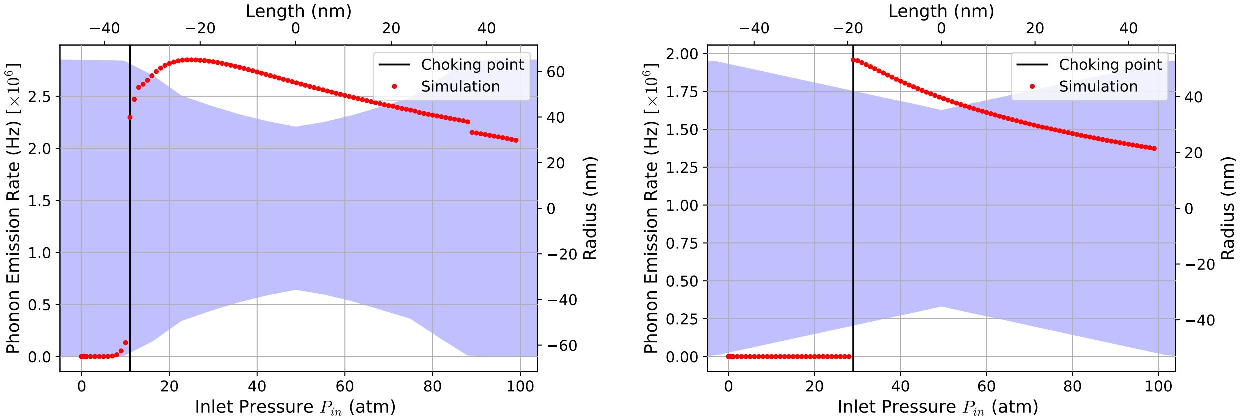
sensitivity of phonon emission rates to the precise nozzle geometry near the horizon, as both samples show opposite behavior.

The discrepancies between simulation models highlight the critical dependence of the velocity profile on the nozzle geometry, particularly near the horizon. Since the Unruh temperature scales linearly with $\frac{\partial u_z}{\partial z}$ and the phonon emission rate scales cubically, small differences in velocity gradients propagate into substantial deviations in phonon emission rates and Unruh temperature. Importantly, the values of $\frac{\partial u_z}{\partial z}$ used here were calculated along



(A) Variable geometry Unruh temperature.

(B) Opening angle Unruh temperature.



(C) Variable geometry phonon emission rate.

(D) Opening angle phonon emission rate.

Figure 5.8: Phonon emission rates and corresponding Unruh temperatures for experimental Sample 2, for both variable geometry and opening angle simulations. The shaded regions in all plots represent the nozzle profile, with the top and right axes corresponding to its dimensions.

the central axis at the throat, which does not coincide with the actual horizon position, as shown in the previous section. Without experimental verification, these results may only be interpreted as approximations. The geometric parameter dependencies for emission rates and Unruh temperature are provided in Appendix D, which may aide in experimental design.

Notably, Unruh's estimations, conducted for superfluid helium, predicted that a single micron-sized pore would have an Unruh temperature of 0.3 mK, and 10^4 micron-sized pores would emit phonons at a rate of 0.5 GHz with a corresponding Unruh temperature of 6 mK [3]. These predictions are of a similar order to the results obtained for a single nanopore, which is more practical than dealing with 10^4 pores, making the single nanonozzle approach more feasible for experimental investigations.

Chapter 6

Conclusion

In this study, the theoretical and computational aspects of fluid dynamics in the context of sonic black holes were investigated through advanced simulations and analyses. By utilizing the capabilities of OpenFOAM with a custom Python framework, this work successfully simulated the formation of sonic black hole horizons, both corroborating the successful formation in previous experimental work and providing new insights into the detailed behaviour of fluid flows in the nanoscopic regime.

The simulation framework was rigorously validated against known analytic solutions, ensuring the reliability of the computational results. Importantly, the model achieved strong agreement (<3%) with experimental mass flow rates without requiring any assumptions about the precise nozzle geometry. Instead, an extensive set of simulations was performed with a plethora of different geometries, where mass flow profiles alone were used to determine the agreement. This approach underscores the robustness of the model in scenarios where experimental data may be incomplete or difficult to obtain.

A simplified geometric model was also employed, focusing primarily on matching the inner radius and opening angle, parameters which are known to have the most significant impact on mass flow[1] and are easy to measure [4]. The simulations demonstrated close matches for inner radius, with the best-matched opening angles being closer to the immediate throat angle of the experimental samples rather than the experimentally-measurable angle spanning the throat and outlet. This finding suggests a potential method for estimating throat radius at high pressures, where direct measurements of conductance are unreliable. However, the lack of specific geometric details renders this simplified model insufficient for accurately determining the full opening angle, particularly for high-pressure regimes.

While the simplified models provided reasonably accurate predictions for mass flow, they were less reliable for estimating phonon emission rates, as these rates showed a strong dependency on the full nozzle geometry[2]. Simulations revealed that even slight variations in geometry could result in significant differences in phonon emission rates. Appendix D illustrates that when the complete nozzle profile is known and parameterizable, phonon emission rates can be estimated within acceptable accuracy, particularly for nozzles with small inner radii ($R_{in} \lesssim 60$ nm).

In conclusion, this work establishes a powerful computational framework for studying fluid flows and associated phenomena in nanoscale nozzles, particularly for sonic black hole research. The framework enables predictions of critical details such as mass flow, compressibility effects, and phonon emission rates, which are essential in designing an optimal experimental apparatus to probe the predicted Unruh-Hawking radiation in a laboratory setting. Thus, these findings provide a foundation for future experimental investigations of sonic black holes, potentially advancing our understanding of astrophysical black holes.

---

[1] See Appendix C.

[2] As shown in equation (2.74), $\frac{\partial u_z}{\partial z}$ has an explicit dependence on the nozzle profile *via* $\frac{\partial A}{\partial z}$, which is undefined at the horizon.

Bibliography

[1] S. Carroll, *Spacetime and geometry: an introduction to general relativity*, Cambridge university press, Cambridge, 2019.

[2] L. Landau and E. Lifshitz, *Fluid Mechanics: Volume 6*, Elsevier Science, 2013.

[3] M. Novello, M. Visser, and G. Volovik, *Artificial Black Holes*, World Scientific, 2002.

[4] S. Neale, High Speed Flow Through Silicon Nitride Nanopores as a Potential Dumb Hole, Master's thesis, McGill University, 2013.

[5] M. Petrescu, Computational Fluid Dynamics in Mesoscopic Nozzles, Master's thesis, McGill University, 2015.

[6] P. K. Kundu and I. M. Cohen, *Fluid Mechanics*, Elsevier, Amsterdam, 4th edition edition, 2008.

[7] P. Linstrom, NIST Chemistry WebBook, NIST Standard Reference Database 69, 1997.

[8] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport phenomena*, J. Wiley, New York, 2nd edition edition, 2002.

[9] H. L. Langhaar, Journal of Applied Mechanics **9**, A55 (1942).

[10] J. C. Harley, *Compressible Gas Flows in Microchannelsand Microjets*, PhD thesis, University of Pennsylvania, Philadelphia, PA, 1993.

[11] J. C. Maxwell, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **19**, 19 (1860).

[12] J. C. Maxwell, Philosophical Transactions of the Royal Society **170**, 231 (1879).

[13] D. A. Lockerby, J. M. Reese, D. R. Emerson, and R. W. Barber, Phys. Rev. E **70**, 017303 (2004).

[14] W. G. Unruh, Physical Review D **14**, 870 (1976).

[15] S. W. Hawking, Communications In Mathematical Physics **43**, 199 (1975).

[16] W. G. Unruh, Physical Review Letters **46**, 1351 (1981).

[17] H. Furuhashi, Y. Nambu, and H. Saida, Classical and Quantum Gravity **23**, 5417 (2006).

[18] J. Ferziger and M. Perić, *Computational methods for fluid dynamics*, Springer, Berlin, 3rd revised edition edition, 2002.

[19] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding, Two calculation procedures for steady, three-dimensional flows with recirculation, in *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, edited by H. Cabannes and R. Temam, volume 19, pages 60–68, Springer, Berlin, Heidelberg, 1973.

[20] S. Patankar and D. Spalding, International Journal of Heat and Mass Transfer **15**, 1787 (1972).

[21] R. Issa, Journal of Computational Physics **62**, 40 (1986).

[22] S. V. Patankar, *Numerical heat transfer and fluid flow*, Series in computational methods in mechanics and thermal sciences, McGraw-Hill, New York, NY Hamburg, 1980.

[23] C. Greenshields, *OpenFOAM v8 User Guide*, The OpenFOAM Foundation, London, UK, 2020.

[24] W. Sutherland, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **36**, 507 (1893).

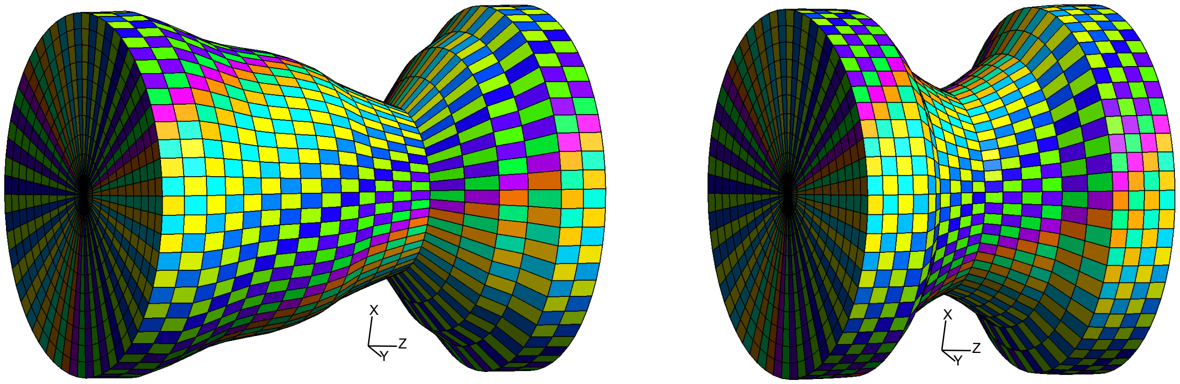
[25] C. Geuzaine and J. Remacle, International Journal for Numerical Methods in Engineering **79**, 1309 (2009).

[26] J. Woodward, Source Modeling – Discharge Rates, in *Reference Module in Chemistry, Molecular Sciences and Chemical Engineering*, Elsevier, 2014.

[27] T. C. Lilly, S. F. Gimelshein, A. D. Ketsdever, and G. N. Markelov, Physics of Fluids **18**, 093601 (2006).

Appendix A

Mesh Creation

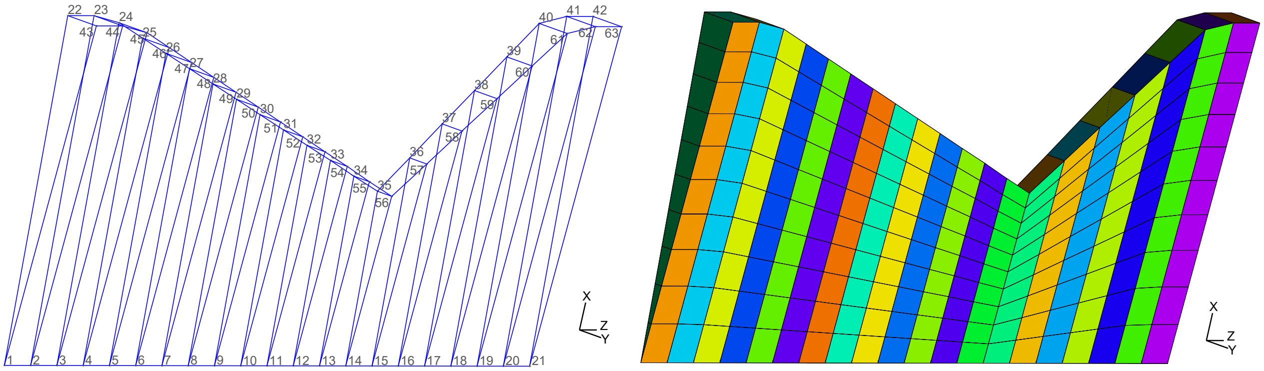


(A) Best-matched Sample 1 full nozzle. **(B)** Best-matched Sample 2 full nozzle.

Figure A.1: An example of the full nozzle mesh for the best-matched profiles found with the variable geometry simulations. The graphical mesh representations are scaled to fit the page uniformly. Both nozzles have a fixed length of $L = 100$ nm.

The initial simulations of this study used the full nozzle geometry. The geometry, constructed using `gmsh`, had 1302 points, 3781 curves, 3680 surfaces, and 1200 total control volumes. The geometry was ultimately defined by 20 faces along the central axis, 1 initial surface along the radial axis, and 60 surfaces along the azimuthal axis. The radial axis was then split into 10 surfaces during meshing. When meshed, the geometry had a total of 12831 computational nodes consisting of 15310 line segments, 1260 triangles, 24740 quadrangles, 10800 hexahedra, and 1200 prisms. The mesh file alone required 275 MB of storage. Due to the high computational cost, initial simulations on this geometry took over **one week** to complete for a single applied pressure using a simple parabolic nozzle profile. An example of what such a mesh would look like is shown in Fig. A.1 for the best-matched variable geometry nozzles for Sample 1 and Sample 2.

To address the computational inefficiencies, the geometry was simplified using the `wedge` approach discussed in Section 4.1.3.1. The wedge geometry, representing a 6° slice of the full geometry shown in Fig. A.1, significantly reduced the computational load while maintaining the same number of points along the central and radial axes. The slice was defined by 63 points, 123 curves, 81 faces, and 20 volumes. An example geometry for a slice is shown in Fig. A.2A. To optimize the mesh for flow simulations, a transfinite horizontal progression was used to increase the mesh density near the boundary walls, where gradients in flow properties are highest. This measure was employed to improve the resolution of geometric effects while minimizing computational overhead. The mesh is shown in Fig. A.2B, and the horizontal progression is visible along the radial axis. When meshed, the `wedge` geometry only contained 441 computational nodes, connected between 501 lines, 21 triangles, 609 quadrangles, 180 hexahedra, and 20 prisms.



(A) Example `wedge` geometry. The 63 defining points are numbered.

(B) Example `wedge` mesh. Original surfaces from the geometry definition are like-coloured.

Figure A.2: Example wedge geometry and mesh. The mesh is coloured by originally defined faces. Transfinite horizontal progression was used to refine CVs near the outer boundary.

Appendix B

Poiseuille in different regimes

B.1 Compressible Fluid

In compressible fluids, the assumption $\vec{\nabla} P \approx \Delta P/L$ taken in Section 2.1.3 may not necessarily be valid, so the equation of motion (2.36) is

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z}{\partial r}\right) = -\frac{1}{\eta}\left(\frac{\partial P}{\partial z}\right), \tag{B.1}$$

which when integrated as in Section 2.1.3 yields

$$u_z(r) = -\frac{1}{4\eta}\left(r^2 - R^2\right)\frac{\partial P}{\partial z}, \tag{B.2}$$

$$Q = \frac{\pi\rho R^4}{8\eta}\left(-\frac{\partial P}{\partial z}\right). \tag{B.3}$$

Using the ideal gas law in the form $P/\rho = P_0/\rho_0$, where subscript 0 denotes inlet values ($z = 0$), mass flow can be written as

$$Q = \frac{\pi R^4}{8\eta}\frac{\rho_0}{P_0}\left(-P\frac{\partial P}{\partial z}\right). \tag{B.4}$$

Since mass flow is a conserved quantity, taking the tube average by way of integration gives

$$Q = \frac{1}{L}\int_0^L Q\mathrm{d}z = \frac{\pi R^4}{16\eta L}\frac{\rho_0}{P_0}\left(P_0^2 - P_L^2\right), \tag{B.5}$$

where subscript L denotes outlet values ($z = L$) and η is assumed constant. Writing $P_0^2 - P_L^2 = (P_0 + P_L)(P_0 - P_L)$ and using the ideal gas law, the Poiseuille approximation for compressible flows is

$$Q = \frac{\pi R^4 \rho_{\text{avg}}}{8\eta L}\left(P_0 - P_L\right) = \frac{\pi \Delta P}{8\nu_{\text{avg}} L}R^4, \tag{B.6}$$

where $\rho_{\text{avg}} = \frac{1}{2}(\rho_0 + \rho_L)$ is the average density calculated at average pressure $P_{\text{avg}} = \frac{1}{2}(P_0 + P_L)$, and $\nu_{\text{avg}} = \eta/\rho_{\text{avg}}$. This result is consistent with the incompressible Poiseuille approximation given by equation (2.41).

B.2 Slip Flow

Slip flow, defined by slip coefficient ζ *via* equation (2.65), varies inversely with pressure by way of the mean-free path dependence (2.57), *i.e.* $\zeta = \zeta_0/P$ where ζ_0 is a molecular and geometric constant. For a cylindrical pipe, the slip boundary condition is defined as

$$u_z(R) = -\zeta \left.\frac{\partial u_z}{\partial r}\right|_{r=R}. \tag{B.7}$$

By integrating (B.1) in the same way as in Section 2.1.3 with the slip boundary condition (B.7) at $r = R$, the resulting velocity and mass flow is

$$u_z(r) = \left[-\frac{1}{4\eta}\left(r^2 - R^2\right) + \frac{\zeta_0 R}{2\eta P}\right]\left(\frac{\partial P}{\partial z}\right), \tag{B.8}$$

$$Q = 2\pi\rho\int_0^R r u_z(r)\mathrm{d}r = \frac{\pi\rho}{8\eta}R^4\left(1 + \frac{4\zeta_0}{RP}\right)\frac{\partial P}{\partial z}. \tag{B.9}$$

With identical treatment as the previous section, equation (B.9) becomes

$$Q = \frac{\pi R^4}{8\eta L}\frac{\rho_0}{P_0}\left[\frac{1}{2}\left(P_0^2 - P_L^2\right) + \frac{4\zeta_0}{R}\left(P_0 - P_L\right)\right], \tag{B.10}$$

$$= \frac{\pi \Delta P R^4 \rho_{\text{avg}}}{8\eta L}\left(1 + \frac{4\zeta_0}{RP_{\text{avg}}}\right) = \frac{\pi \Delta P R^4}{8\nu_{\text{avg}} L}\left(1 + \frac{4\zeta_0}{RP_{\text{avg}}}\right), \tag{B.11}$$

which converges to the incompressible no-slip Poiseuille solution (2.41) for high average pressures.

Appendix C

Geometric Dependence of Mass Flow

C.1 Small Subset

To study the geometric dependence of inner radius and opening angle on mass flow, opening angle simulations at a fixed throat position were compared at discrete points of inner radius and opening angle. When plotted individually, coloured by the specific geometric parameters, inner radius appears to have the strongest effect on mass flow, as expected by the ideal flow under the isentropic relation given by equation (2.93), which has a quadratic dependence on inner radius due to its cross-sectional area, A^*.

Naively, one would assume that mass flow is only dependent on the throat's cross-sectional area, which would be consistent with the 1D isentropic flow approximation presented in Section 2.2.2. However, real systems deviate from the idealized models. Quantifying this deviation relies on a dimensionless number called the *discharge coefficient*, denoted C_d, which is related to the efficiency of a flow through a nozzle. The discharge coefficient is defined as the ratio [8]

$$C_d = \frac{Q_{exp}}{Q_{ideal}}, \tag{C.1}$$

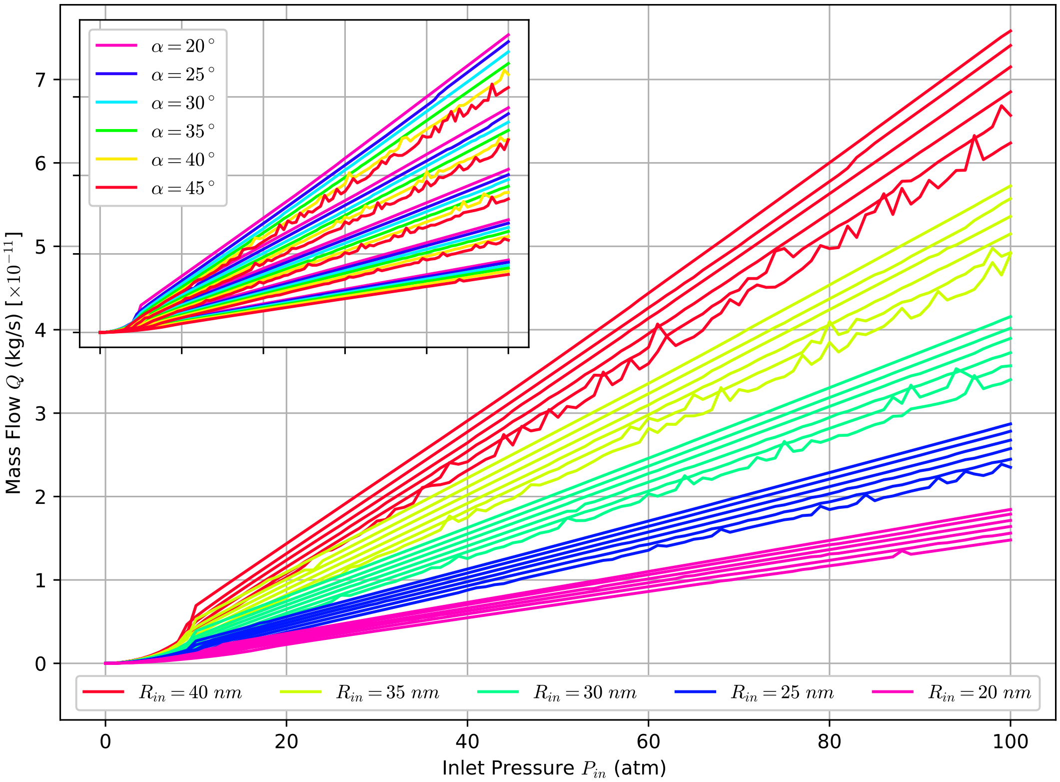


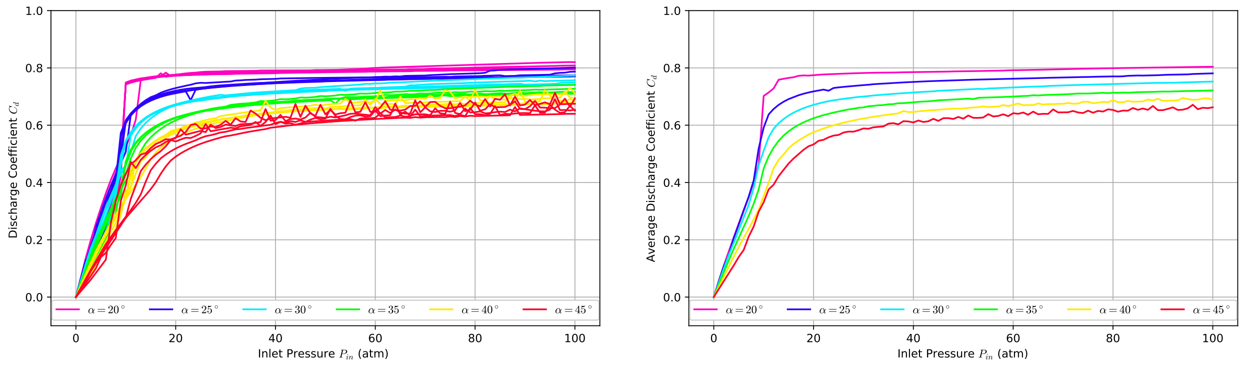
Figure C.1: Mass flow Q as a function of inlet pressure P_{in}, coloured based on geometric parameters. Opening angle simulations fixing throat radius R_{in} and opening angle α were performed, with the throat position fixed at the midpoint of the nozzle.

where Q_{ideal} is the ideal flow. In our case, Q_{ideal} is the mass flow of a compressible isentropic fluid, given by equation (2.93). Q_{exp} is the real flow rate, determined by measurement or simulation. Substituting equation (2.93) into the definition of the discharge coefficient, we can write the real flow rate as,

$$Q_{exp} = C_d P_0 A^* \sqrt{\frac{\gamma}{R_s T_0}} \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma+1}{2(\gamma-1)}} . \tag{C.2}$$

While theoretical work on this dimensionless number is lacking, it is typically measured to determine a relative efficiency of some nozzle. For the purposes of this study, the discharge coefficient was calculated for the mass flow profiles shown in Fig. C.1 using equation (C.1).

As seen in Fig. C.2, the discharge coefficients calculated are consistent for each opening angle and are independent of inner radius R_{in}. While discharge coefficients have been



(A) Raw discharge coefficients.

(B) Binned discharge coefficients.

Figure C.2: Discharge coefficients calculated for the mass flow profiles shown in Fig. C.1. The raw data **(A)** was binned **(B)** by taking the average for each opening angle α.

correlated as a function of nozzle aspect ratio and Reynolds number [8], there is a direct, seemingly-linear, relationship between the discharge coefficient C_d and opening angle α in this system. The discharge coefficient also appears to depend on the applied inlet pressure P_{in}, asymptotically approaching some value as pressure increases. If we assume that the discharge coefficient is linear in α, the relationship can be approximated as

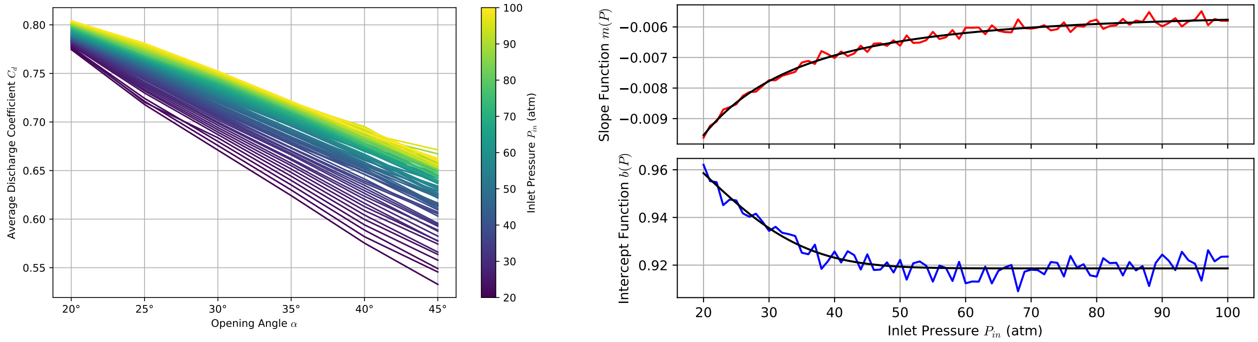
$$C_d \approx m(P)\alpha + b(P), \tag{C.3}$$

where $m(P)$ and $b(P)$ are the slope and intercept values as a function of inlet pressure. A least-squares linear fit to the binned discharge coefficient profiles were performed for each pressure above $P_{in} = 20$ atm. The discharge coefficient as a function of opening angle α is shown in Fig. C.3A, coloured based on inlet pressure. The fit parameters, plotted as a function of inlet pressure, are shown in Fig. C.3.

From the linear fit, we see that the $\alpha = 0°$ discharge coefficient for high pressures stabilizes at $C_d = 0.92$. Below $P_{in} \approx 40$ atm, the discharge coefficient begins to increase, reaching its maximal value of $C_d = 0.96$ at the lowest pressure studied, $P_{in} = 20$ atm. Should this increase continue, we would obtain a discharge coefficient of $C_d \simeq 1$ at $P_{in} = 0$ atm and $\alpha = 0°$, which is consistent with our close match to the idealized analytic mass flow profiles seen in Section 5.1.1 at low pressures $P_{in} \lesssim 1$ atm.

Experimental measurements of the discharge coefficient for orifices in rarefied gases are notably sparse, compounding the lack of theoretical work. For most gases, the discharge coefficient is typically reported as $C_d \simeq 0.975$ [26]. Specific measurements conducted on small orifices using helium gas revealed a pressure-dependent behaviour, with values ranging from $C_d = 0.6$ at low pressures to nearly $C_d = 0.9$ at higher pressures [27]. These findings, which they confirmed are captured *via* Navier-Stokes simulations, are consistent with the trends observed in our study. For orifices with larger aspect ratios, the discharge coefficient can drop further, reaching as low as $C_d = 0.2$ at lowest pressures measured. While the pressure ranges in [27] differ significantly from ours, the measurements were performed on fluids with Knudsen numbers spanning $0.003 \leq K_n \leq 0.05$. This range contained our Knudsen numbers of $0.0035 \leq K_n \leq 0.35$, calculated in Section 2.1.5.1.

These findings provide an explanation for the deviations of measured and simulated values from the ideal ones, which were previously erroneously attributed to Mach 1 not being reached in our group's past studies [4].



(A) C_d per α, coloured by P_{in}.

(B) Computed fit parameters $m(P)$ and $b(P)$.

Figure C.3: The binned discharge coefficient was linearly fit via least-squares method for each pressure. This analysis yields a maximum discharge coefficient of $C_d = 0.92$ at $\alpha = 0°$, tending towards $C_d = 1$ as $P_{in} \to 0$.

C.2 Large Subset

For consistency, the same analysis as in the previous section was performed for all simulations with the same fixed inner radius as before. In this expanded dataset, all simulations that did not reach the choking condition by $P_{in} = 10$ atm were excluded. This filtering left a total of 79 simulations for comparison out of the 253 shown in Fig. C.4. Both variable geometry and opening angle simulations were included, with the opening angle α computed for each simulation. Despite the increased noise, the same general trends observed in Fig. C.1 are evident in the larger dataset, as shown in Fig. C.4. A more detailed analysis of the discharge coefficient C_d was conducted on this expanded dataset. The discharge coefficients, computed from the mass flow profiles in Fig. C.4, are presented in Fig. C.5. The raw discharge coefficients are shown in Fig. C.5A, and the binned results, averaged over 5° increments of α, are displayed in Fig. C.5B.

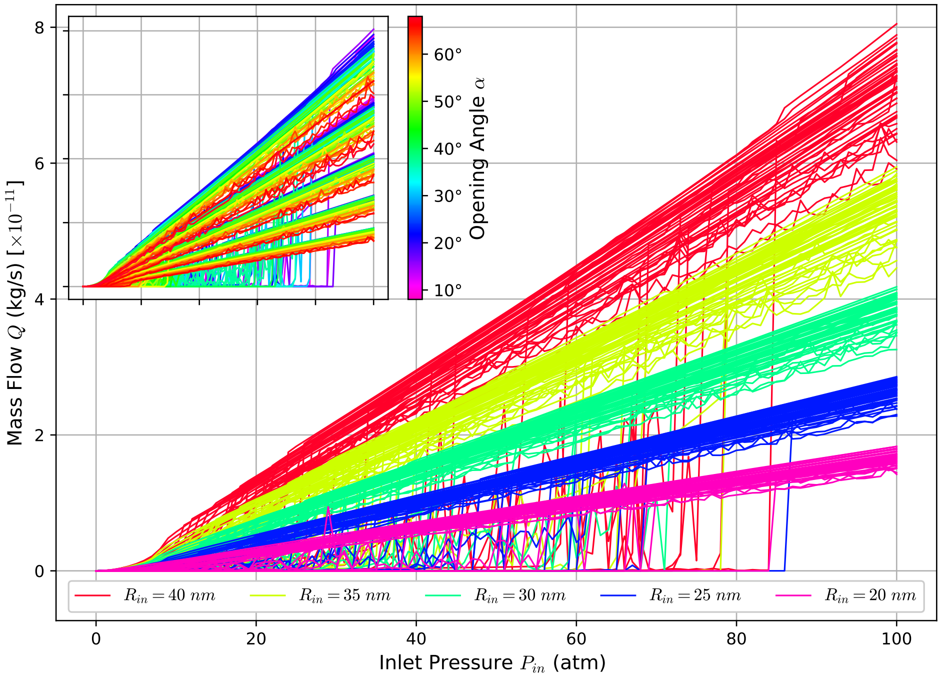
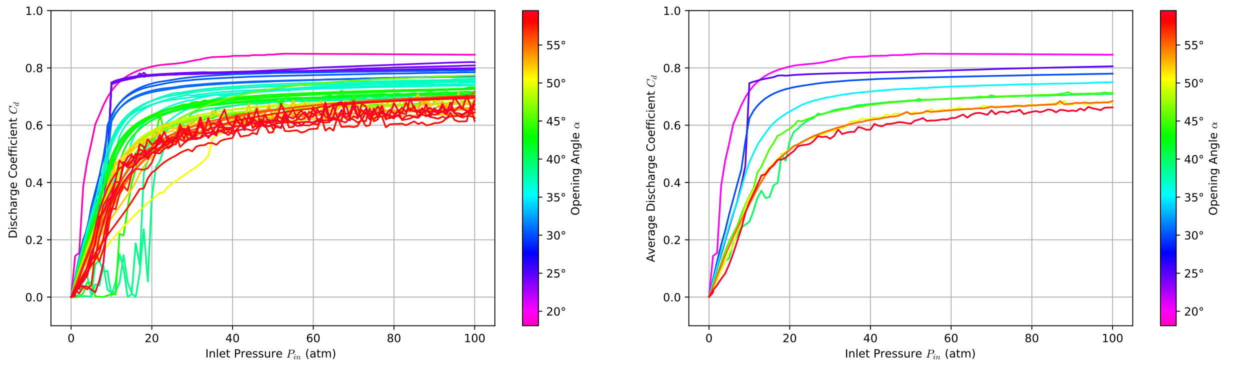
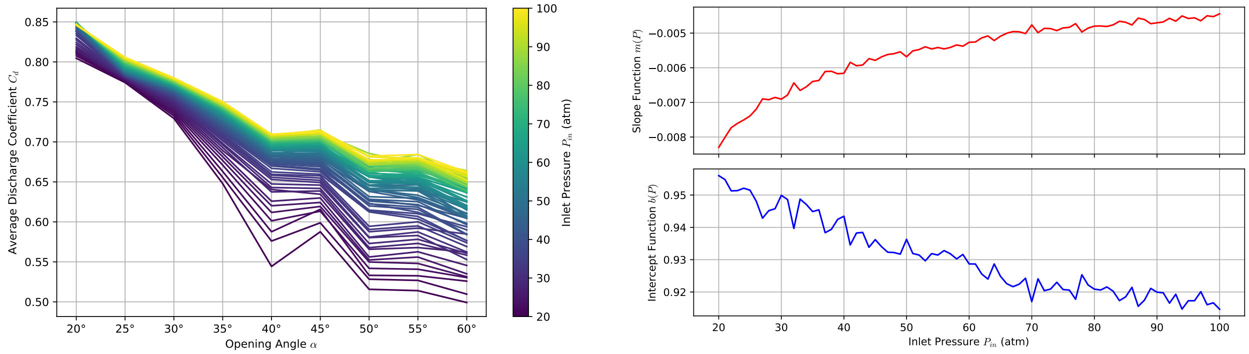


Figure C.4: Mass flow Q as a function of inlet pressure P_{in}, coloured based on geometric parameters. Simulations were performed with discrete R_{in} values and continuously varying opening angles. The opening angle dependence on mass flow is shown in the inset, consistent with the results from the smaller dataset.



(A) Raw discharge coefficients.

(B) Binned discharge coefficients.



(C) C_d per α, coloured by P_{in}.

(D) Computed fit parameters $m(P)$ and $b(P)$.

Figure C.5: Discharge coefficients calculated for the mass flow profiles shown in Fig. C.4. Profiles that did not reach the choking condition by $P_{in} = 10$ atm were omitted. The raw data **(A)** were binned **(B)** by averaging discharge coefficients over 5° increments of opening angle α. Linear fits *via* the least-squares method were applied to the binned data for each pressure, shown in **(C)**. Despite increased noise, the linear trend persists. Fit parameters, shown in **(D)** as a function of inlet pressure, are consistent with Fig. C.3. This larger dataset yields a maximum discharge coefficient of $C_d = 0.92$ at $\alpha = 0°$, with the same trend towards $C_d = 1$ at $P_{in} = 0$ atm.

The enlarged dataset corroborates the findings from the smaller subset, with consistent trends in discharge coefficients across both. These results further validate the robustness of the observed trends, even when analyzed across a broader range of simulations. While the noise is more pronounced due to the inclusion of additional geometries and pressure points, the trends in C_d's dependence on α and P_{in} remain clear, confirming the relationship's persistence under varying conditions.

C.3 Relation to Pressure Difference

In Section 5.2, we saw that the isentropic mass flow $Q_{isentropic}\left(P_0\right)$, as defined in equation (2.93), more closely matched the measured and simulated mass flow Q_{exp} when calculated using the pressure drop across the nozzle (*i.e.* $P_0 = \Delta P$) rather than the inlet pressure (*i.e.* $P_0 = P_{in}$). If we take $Q_{exp} \approx Q_{isentropic}\left(\Delta P\right)$, the discharge coefficient C_d can be approximated as

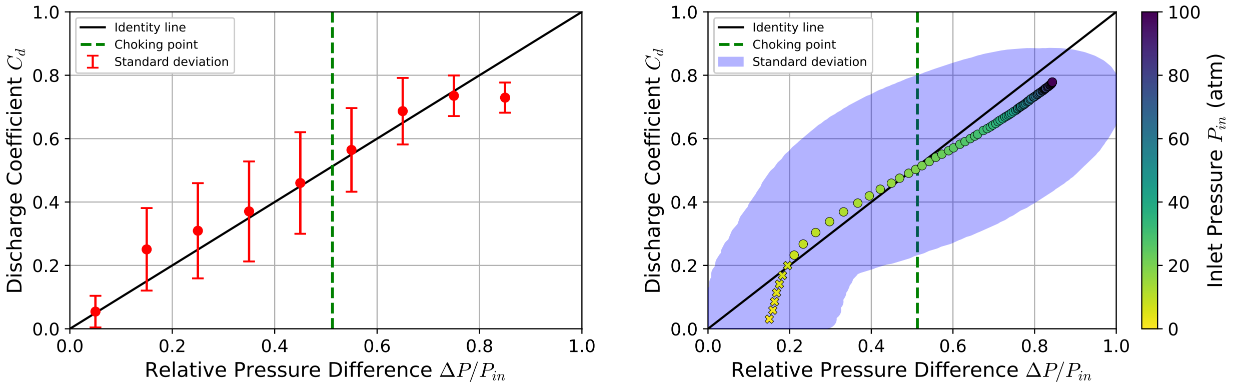
$$C_d = \frac{Q_{exp}}{Q_{isentropic}\left(P_{in}\right)} \approx \frac{Q_{isentropic}\left(\Delta P\right)}{Q_{isentropic}\left(P_{in}\right)} = \frac{\Delta P}{P_{in}}, \tag{C.4}$$

and thus $C_d \approx \Delta P / P_{in}$.

To construct a meaningful dataset for analysis, a broader subset of simulations were selected based on specific physical and geometric criteria. Firstly, each simulation selected was required to exhibit choked flow *via* equation (2.84), ensuring that comparison to the isentropic limit would be valid. Additionally, only simulations with opening angles between $5^\circ < \alpha < 65^\circ$ were considered, and throat position was constrained to the central region of the nozzle (within 25% and 75% of its length). Applying these criteria left 1072 simulations, with all $R_{in} > 5$ nm and $R_{out} < 150$ nm. 100 pressure data points of each simulation from $P_{in} = 1$ atm to $P_{in} = 100$ atm were deliberately chosen to include sub-choked flow—where C_d tends to be smaller than those analyzed in the previous section—capturing the transition to choked flow and the trend of C_d and $\Delta P / P_{in}$ as P_{in} increases.

For each data point, C_d was calculated from the simulated mass flow Q_{exp} using equation (C.2) with $P_0 = P_{in}$, the pressure drop was computed directly as $\Delta P = P_{in} - P_{out}$, and Knudsen number K_n was calculated for regime identification. Of the 107 200 data points, 2 190 were in the transition regime ($K_n > 0.1$, all below $P_{in} = 8$ atm), and the remaining 105 010 were in the slip regime. Between $P_{in} = 8$ atm and $P_{in} = 100$ atm, all data points had Knudsen numbers between 0.0970 and 0.00125.

To investigate the global trend among the selected simulations, data points within the

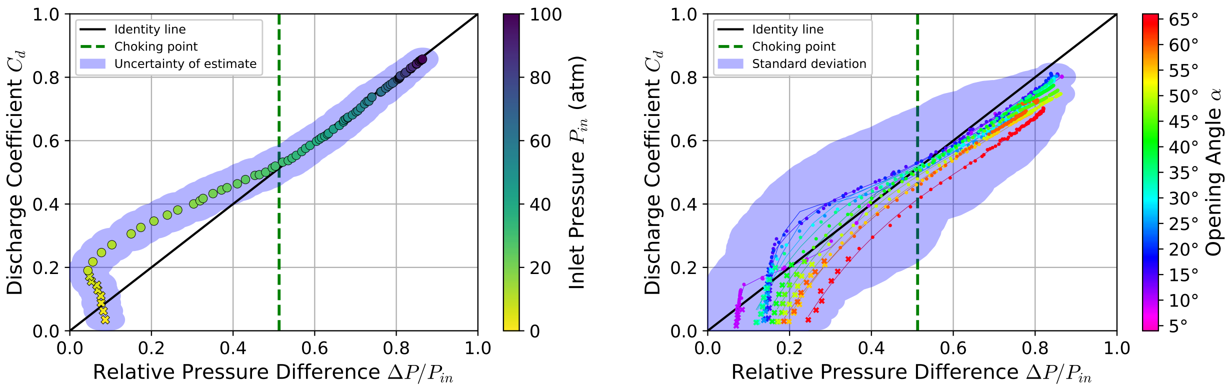


(A) C_d binned by $\Delta P/P_{in}$ in 0.1 intervals. **(B)** Mean C_d and $\Delta P/P_{in}$ at each P_{in}.

Figure C.6: Discharge coefficient C_d as a function of relative pressure difference $\Delta P/P_{in}$. **(A)** C_d binned by $\Delta P/P_{in}$ for data in the slip regime ($K_n < 0.1$), with error bars showing the standard deviation of the mean. **(B)** C_d and $\Delta P/P_{in}$ averaged at each P_{in}, with standard deviations in both quantities shaded blue. Data points marked with a cross indicate bins containing one or more simulations in the transition regime, where rarefaction effects may cause deviations from the overall trend.

slip regime were binned by $\Delta P/P_{in}$ in increments of 0.1. Within each bin, the means and standard deviations of C_d were calculated. As shown in Figure C.6A, C_d increases with $\Delta P/P_{in}$, closely following the identity line within error until $\Delta P/P_{in} > 0.8$. A secondary analysis was performed by averaging C_d and $\Delta P/P_{in}$ at each inlet pressure P_{in}. The results, shown in Figure C.6B, confirm that both quantities tend to increase with P_{in} and follow the identity line.

Despite the general agreement seen in Figure C.6, variability remains significant due to the enormous geometric spread in simulations. To isolate the geometric influence, least-squares linear regression was performed on 1072 data points for each 100 inlet pressures, with C_d and $\Delta P/P_{in}$ independently regressed against opening angle α. The $\alpha = 0°$ limits of both C_d and $\Delta P/P_{in}$ at each pressure are shown in Figure C.7A. These intercepts are in close agreement with each other and lie on the identity line, suggesting $C_d \to \Delta P/P_{in}$ as $\alpha \to 0°$. The dataset used for this regression is shown in Figure C.7B, where C_d and $\Delta P/P_{in}$ were binned by opening angle in 5° intervals.



(A) Zero-angle intercept of C_d and $\Delta P/P_{in}$.

(B) Mean C_d and $\Delta P/P_{in}$ binned by α.

Figure C.7: Dependence of discharge coefficient C_d and relative pressure difference $\Delta P/P_{in}$ on opening angle α. **(A)** Least-squares linear regression used to extrapolate to $\alpha = 0°$ at each inlet pressure P_{in}. **(B)** Data used in regressions of **(A)** binned by opening angle α in 5° intervals. Above the choking point, the trend $C_d \lesssim \Delta P/P_{in}$ converges onto the identity line as $\alpha \to 0°$, with deviations increasing as α increases.

From the isentropic choking condition given by equation (2.84), and recognizing that pressure decreases monotonically along the nozzle, we have $P_{out} \leq P^*$, leading to the inequality

$$\frac{\Delta P}{P_{in}} \geq 1 - \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \approx 0.53, \tag{C.5}$$

for a monatomic gas with $\gamma = \frac{5}{3}$. This threshold (dashed green), corresponds to the onset of choked flow, and in Figure C.7A, it marks where C_d and $\Delta P/P_{in}$ fall on the identify line.

Ultimately, this finding is practically useful as C_d, a quantity determined from external measurements of flow, serves as a reliable approximation for $\Delta P/P_{in}$. In turn, this allows experimental estimation of internal flow quantities such as velocity $u(z)$ *via* equation (2.82), and by extension phonon emission rate *via* equation (2.110), by estimating $P(z)$ with ΔP. Furthermore, the general trend of $C_d \lesssim \Delta P/P_{in}$ is consistent with $Q_{exp} \lesssim Q_{isentropic}(\Delta P)$ seen in all cases reported in Section 5.2.

Appendix D

Phonon Emission Rate Estimations

In this chapter, we approximate $\frac{\partial u_z}{\partial z}$ using the equations derived in Chapter 2 and compare them to the simulation results. The purpose of these derivations is to allow rough approximations for the phonon emission rates of sonic horizons formed in converging-diverging nozzles with few geometric details. This would allow for the assessment of specific geometries to develop and optimize an experimental design for measuring emitted phonons, before fabricating the devices and performing measurement.

D.1 Geometry-Dependent Relation

From Euler's equations for inviscid flow, assuming a constant entropy system, it is possible to estimate the phonon emission rate by estimating the velocity profile at the throat of the nozzle. We start with the differentiated mass flow conservation equation (2.71),

$$\frac{1}{u_z}\frac{\partial u_z}{\partial z} + \frac{1}{\rho}\frac{\partial \rho}{\partial z} + \frac{1}{A}\frac{\partial A}{\partial z} = 0, \tag{D.1}$$

and the result of manipulating Euler's momentum equation in one dimension (2.73),

$$\frac{1}{\rho}\frac{\partial \rho}{\partial z} = -\frac{u_z^2}{c_s^2}\left(\frac{1}{u_z}\frac{\partial u_z}{\partial z}\right). \tag{D.2}$$

Substituting equation (D.2) into (D.1), we obtain the relationship between central-axis velocity and change in area:

$$\frac{1}{u_z}\frac{\partial u_z}{\partial z} - \frac{u_z}{c_s^2}\frac{\partial u_z}{\partial z} + \frac{1}{A}\frac{\partial A}{\partial z} = 0, \tag{D.3}$$

which can be written as,

$$\frac{c_s^2 - u_z^2}{u_z c_s^2}\frac{\partial u_z}{\partial z} = -\frac{1}{A}\frac{\partial A}{\partial z}. \tag{D.4}$$

Near the horizon, we can assume that velocity is linearly perturbed from the speed of sound, that is,

$$u_z(z) = c_s + \delta u_z(z), \tag{D.5}$$

and we can write,

$$u_z(z)^2 = (c_s + \delta u_z(z))^2 = c_s^2 + 2c_s\delta u_z + (\delta u_z)^2 \approx c_s^2 + 2c_s\delta u_z, \tag{D.6}$$

dropping higher-order terms. From the Taylor series expansion, we can estimate velocity near the horizon as

$$\delta u_z(z) \approx (z - z_0)\left.\frac{\partial u_z}{\partial z}\right|_{z=z_0}, \tag{D.7}$$

and taking $z_0 = 0$ at the throat, we can approximate the numerator of equation (D.4) as

$$u_z^2 = c_s^2 + 2c_s z\frac{\partial u_z}{\partial z} \quad \text{or} \quad c_s^2 - u_z^2 = -2c_s z\frac{\partial u_z}{\partial z}. \tag{D.8}$$

Substituting (D.8) into equation (D.4), we arrive at,

$$\frac{2z}{u_z c_s}\left(\frac{\partial u_z}{\partial z}\right)^2 = \frac{1}{A}\frac{\partial A}{\partial z} \approx 2z\left(\frac{1}{c_s}\frac{\partial u_z}{\partial z}\right)^2, \tag{D.9}$$

near the throat. This is a powerful relation which can be applied to any nozzle geometry to obtain an approximation of $\frac{\partial u_z}{\partial z}$ near the sonic horizon.

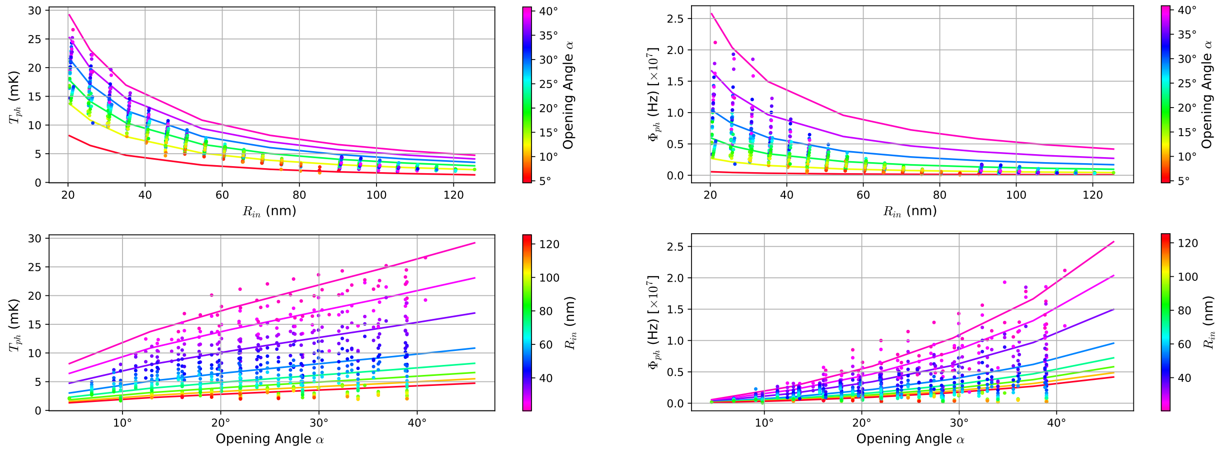
D.1.1 Linear Approximation

Taking a quadratic approximation of area, $A = A_0 + \beta z^2$ for unitless constant β, which would be the case for a mostly linear geometry, we can approximate $\frac{\partial u_z}{\partial z}$ near the throat as

$$\frac{\partial u_z}{\partial z} \approx c_s \sqrt{\frac{\beta}{A_0}}. \tag{D.10}$$

Taking $\beta = \pi \tan \alpha$ near the throat, where α is the opening angle of the nozzle, we can approximate $\frac{\partial u_z}{\partial z}$ as

$$\left.\frac{\partial u_z}{\partial z}\right|_{u_z = c_s} = c_s \sqrt{\frac{\pi \tan \alpha}{\pi R_{in}^2}} = \frac{c_s \sqrt{\tan \alpha}}{R_{in}}. \tag{D.11}$$



(A) Unruh temperature. **(B)** Phonon emission rate.

Figure D.1: Estimation of $\frac{\partial u_z}{\partial z}$ for Sample 1 and comparison to simulated data shown with respect to both inner radius and opening angle. Simulated values, shown as coloured points, are the maximal values of $\frac{\partial u_z}{\partial z}$ obtained *via* simulation, whereas estimated values calculated from equation (D.11) are drawn as curves. For Unruh temperature T_{ph}, which is linearly related to $\frac{\partial u_z}{\partial z}$, equation (D.11) shows good agreement between the simulated data and the estimation derived in this section. For phonon emission rate Φ_{ph}, which has a cubic relation to $\frac{\partial u_z}{\partial z}$, the estimation begins to deviate by up to a factor of 2 for $R_{in} \lesssim 60$ nm.

In this study, experimental Sample 1 has a geometry that is approximately linear. When using (D.11) to approximate the simulation results, an acceptable match is found considering the generous approximations used. The results are shown in Figure D.1.

D.1.2 Quadratic Approximation

For Sample 2, which shows a more quadratic form, similar mathematical treatment can be taken. If consider $A(z) = \pi r(z)^2$, then we can write

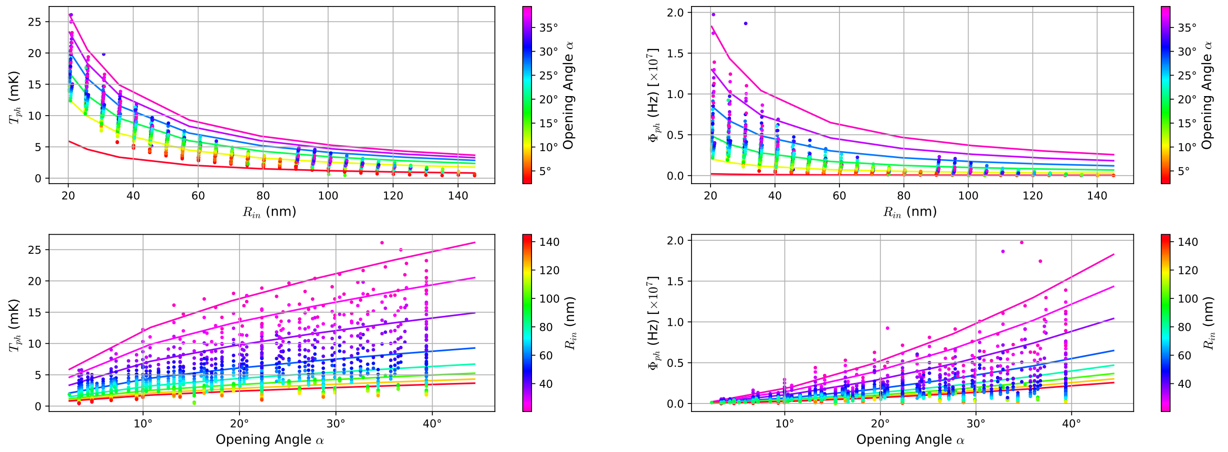
$$\frac{1}{A}\frac{\partial A}{\partial z} = \frac{1}{\pi r^2}\frac{\partial 1}{\partial z}(\pi r^2) = \frac{2}{r}\frac{\partial r}{\partial z}, \tag{D.12}$$

and (D.9) becomes

$$z\left(\frac{1}{c_s}\frac{\partial u_z}{\partial z}\right)^2 = \frac{1}{r}\frac{\partial r}{\partial z}, \tag{D.13}$$

Considering a quadratic geometry defined by $r(z) = R_{in} + \frac{1}{2}\beta z^2$, where β is some constant, then our equation similarly reduces to

$$\frac{\partial u_z}{\partial z} = c_s\sqrt{\frac{\beta}{R_{in}}}, \tag{D.14}$$



(A) Unruh temperature. **(B)** Phonon emission rate.

Figure D.2: Estimation of $\frac{\partial u_z}{\partial z}$ for Sample 2 and comparison to simulated data shown with respect to both inner radius and opening angle. Simulated values, shown as coloured points, are maximal values of $\frac{\partial u_z}{\partial z}$ obtained *via* simulation, whereas estimated values calculated from equation (D.16) are drawn as curves. Similar to Figure D.1, equation (D.16) shows a decent agreement for T_{ph}, which deviates by up to a factor of 2 for Φ_{ph} for small opening angle and inner radius.

giving a very good approximation for $\frac{\partial u_z}{\partial z}$ if β is known. To relate it to opening angle, we can consider that $\tan\theta \equiv \frac{\partial r}{\partial z}$ near the throat, and thus

$$\frac{\partial r}{\partial z} = \tan\theta(z) = \beta z. \tag{D.15}$$

Since β should be constant with respect to z, β can be evaluated at any place where the opening angle is known at a distance z.

When dealing with experimental nanopores, however, finding a value of β in relation to opening angle α may prove troublesome, thus would be difficult to compare with simulated results. With the nozzles simulated in this study, β had an additional dependence on relative throat position. A visual inspection of the Sample 2 geometry shows that the opening angle near the throat is approximately half of the opening angle measured between the throat and the outlet. As such, we can apply equation (D.11) with half of the angle measured in the simulations, *i.e.*

$$\left.\frac{\partial u_z}{\partial z}\right|_{u_z=c_s} \approx \frac{c_s\sqrt{\tan\frac{\alpha}{2}}}{R_{in}}, \tag{D.16}$$

which has a similar agreement to experimental Sample 1 with the simulated values, shown in Figure D.2.

D.1.3 General Relation

William Unruh in [3] derived an additional relationship between u_z and A near the horizon, by integrating equation (D.4) from the throat to some position z along the central axis. Starting by integrating the left-hand side of equation (D.4),

$$\int_0^z \frac{c_s^2 - u_z^2}{u_z c_s^2}\frac{\partial u_z}{\partial z}dz = \int_{c_s}^{u_z} \frac{1-(u_z/c_s)^2}{u_z}du_z, \tag{D.17}$$

$$= \int_1^{u_z/c_s} \frac{1-(u_z/c_s)^2}{u_z/c_s}d(u_z/c_s), \tag{D.18}$$

$$= \log\left(\frac{u_z}{c_s}\right) + \frac{1}{2}\left[1-\left(\frac{u_z}{c_s}\right)^2\right], \tag{D.19}$$

and similarly integrating the right-hand side of equation (D.4),

$$\int_0^z \frac{1}{A}\frac{\partial A}{\partial z}dz = \int_{A_0}^{A} \frac{dA}{A} = \log\left(\frac{A}{A_0}\right), \tag{D.20}$$

when combined gives an implicit relationship between the cross-sectional area and velocity profile:

$$\log\left(\frac{u_z}{c_s}\right) + \frac{1}{2}\left[1 - \left(\frac{u_z}{c_s}\right)^2\right] = -\log\left(\frac{A}{A_0}\right). \tag{D.21}$$

This relation assumes that c_s is approximately constant along the nozzle, which is a good approximation for helium-4. This relation, which does not depend on the assumptions made in sections D.1.1 and D.1.2, can be used to approximate $\frac{\partial u_z}{\partial z}$ for any type of nozzle profile near the throat by manipulating the right-hand side and differentiating the result with respect to z.