RMIT UNIVERSITY

# Investigation into Geometric Characteristics of Micro Parabolic Nozzle and Multi-Objective Design Optimisation

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science (Physics) (Honours)


Jack Gale

Bachelor of Space Science, RMIT University

0009-0008-4590-742X

School of Science

College of Science, Technology, Engineering and Maths

RMIT University

Australia


October, 2025

# Declaration of Authorship

I certify that except where due acknowledgement has been made, this research is that of the author alone; the content of this research submission is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

In addition, I certify that this submission contains no material previously submitted for award of any qualification at any other university or institution, unless approved for a joint-award with another institution, and acknowledge that no part of this work will, in the future, be used in a submission in my name, for any other qualification in any university or other tertiary institution without the prior approval of the University, and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of any published works contained within this thesis resides with the copyright holder(s) of those works.

I give permission for the digital version of my research submission to be made available on the web, via the University's digital research repository, unless permission has been granted by the University to restrict access for a period of time.

Jack Gale
October, 2025

# *Acknowledgements*

I would like express my gratitude to many people for their support on this research project, first and foremost of whom is A/Prof Gail Iles. From an offhand comment in a hallway after a lecture, Gail delivered an opportunity overseas in Japan that I will cherish for the rest of my life. Without her support and very, very long list of acquaintances, my honours year may have looked very different. I would like to thank A/Prof. Hideaki Ogawa, for the opportunity provided to work as part of his laboratory and within his MET team, and whose support I am deeply grateful. Assistant Prof. Suk Yeo, whom I am sure is tired of the endless emails received from me about why my simulations weren't working, and whose tireless work got me through the project and code. Keisuke Suenaga, who helped me settle into the lab and got me up and running when I wasn't sure how to operate code, and guided me through the Kyushu code library. A deep appreciation for Prof. Liu Huixin, who gave me a desk and hosted me while I visited Japan (with a brief thanks to A/Prof. Brett Carter who helped us establish contact). Thank you Madeline Farrugia, for reading my thesis, helping me run simulations, and the general overall support given throughout the course of this project. Ben and Sam must be thanked for their computing power and time, which helped me conduct many more simulations than I would have been able to in such a short time. With their help, I can say my project was literally completed with the power of friendship. Lastly, I would like to acknowledge the unwavering support of my parents and grandparents, without whom I wouldn't have been able to pursue my dreams among the stars.

# *Summary*

Evolutionary algorithms, a type of advanced machine learning, are used to optimise a millimetre scale satellite thruster nozzle. Differences between the mathematical *optimal* design and physical reality occur due to performance losses, namely friction along the walls which generates a boundary layer effect. This study aimed to utilise this machine learning to optimise a parabolic nozzle shape at the millimetre scale. Parabolic nozzles, characterised by their curved shape, are regarded as one of the best designs for rocketry at the metre scale. This study provides evidence that the parabolic geometry still outperforms conical geometry at the millimetre scale. The optimised parabolic nozzles excelled in directing exhaust flow out of the nozzle in the correct direction, while also producing high thrust and fuel efficiency.

# Contents

# List of Abbreviations

**EP** Electric Propulsion

**MET** Microwave Electrothermal Thruster

**CD** Converging-Diverging

**CFD** Computational Fluid Dynamics

**FEM** Finite Element Method

**FVM** Finite Volume Method

**MOC** Method of Characteristics

**MDO** Multi-objective Design Optimisation

**EA** Evolutionary Algorithm

**SAEA** Surrogate Assisted Evolutionary Algorithm

**LHS** Latin Hypercube Sampling

**CAD** Computer Aided Design

**LEO** Low Earth Orbit

# List of Symbols

| Symbol | Name | Unit |
|---|---|---|
| $F$ | force | N |
| $m$ | mass | kg |
| $a$ | acceleration | $\mathrm{m/s^2}$ |
| $\vec{V}$ | fluid velocity | $\mathrm{m\,s^{-1}}$ |
| $P$ | pressure | Pa |
| $u, v$ | velocity | $\mathrm{m\,s^{-1}}$ |
| $\vec{n}$ | normal | |
| $Re$ | Reynold number | |
| $L$ | characteristic linear dimension | m |
| $R_i$ | inlet radius | m |
| $R_t, r_t$ | throat radius | m |
| $R_e, r_e$ | exit radius | m |
| $l_c$ | converging length | m |
| $l_t$ | throat length | m |
| $A, A_e$ | exit area | $\mathrm{m^2}$ |
| $A^*$ | throat area | $\mathrm{m^2}$ |
| $L_N$ | divergent length | m |
| $N, Q, E$ | control point | |
| $\dot{m}$ | mass flow rate | $\mathrm{kg\,s^{-1}}$ |
| $V_e$ | exit velocity | $\mathrm{m\,s^{-1}}$ |
| $p_e$ | exit pressure | Pa |
| $p_o$ | ambient pressure | Pa |
| $p_t$ | total pressure | Pa |
| $T_t$ | total temperature | K |
| $R$ | gas constant | $\mathrm{J\,mol^{-1}\,K}$ |
| $M_e$ | mach number | |
| $T_e$ | exit temperature | K |
| $T$ | thrust | N |
| $I_{sp}$ | specific impulse | s |
| $\mu$ | dynamic viscosity | Pa s |
| $\rho$ | density | $\mathrm{kg/m^3}$ |
| $\theta_c$ | converging angle | $^\circ$ |
| $\theta_N$ | initial diverging angle | $^\circ$ |
| $\delta$ | curve factor | |
| $\epsilon$ | expansion ratio | |
| $\varnothing_{c,d}$ | smoothing radius | m |
| $\gamma$ | specific heat ratio | |
| $\eta_D$ | divergence efficiency | |
| $\sigma$ | flow coefficient | $\mathrm{m^3/h/bar}$ |

RMIT UNIVERSITY

# *Abstract*

Bachelor of Science (Physics) (Honours)

**Investigation into Geometric Characteristics of Micro Parabolic Nozzle and Multi-Objective Design Optimisation**

by Jack Gale

Micro-satellite propulsion provides a means for accessible long-term space experimentation and research. One component of propulsion systems is the nozzle, a geometric device which converts enthalpy to thrust. This study aimed to characterise how scale affects key performance metrics of nozzles, namely thrust, divergence efficiency, and specific impulse, which assisted in developing an optimised set of nozzles thereby maximising these metrics. This was accomplished using Computational Fluid Dynamics (CFD) simulations and evolutionary algorithms. A 2-Dimensional axisymmetric mesh was constructed using Rao's parabolic nozzle design, where key geometric values were parameterised in order to investigate how changing different aspects of the nozzle affected the output performance metrics. This flexible design was used to conduct CFD simulations of 432 unique parabolic designs. These were used to construct surrogate models with high confidence, which could predict output performance characteristics given the set of input geometries. These models had errors of less than 0.05 % on the validation data. Thus, surrogate assisted evolutionary algorithms were used to optimise output parameters. A Pareto front was formed and was most defined with respect to the divergence efficiency and specific impulse, which was an inversely correlated relationship. The thrust also displayed Pareto front qualities when analysed against specific impulse and divergence efficiency, but was capped due to parametric bounds. The maximum thrust achieved was 0.362 N, maximum divergence efficiency was 99.68 %, and maximum specific impulse was 110.3 s. Some flow splitting was observed in the high divergence efficiency verification case, but was otherwise not observed in the other verification cases. The parabolic nozzle geometry produced and maintained very good divergence efficiency characteristics, where the lowest non-dominated Pareto solution yielded a divergence efficiency of 99.10 %, which is only a 0.7 % decrease over similar conical divergence efficiency optimised nozzles.

# Chapter 1

# Introduction

## 1.1 Satellites and space

The Wright brothers pioneered flight in 1903, using an understanding of aerodynamics to generate lift. Two world wars accelerated aircraft design and challenged how far our understanding of thrust and lift, which is now more widely grouped into the study of fluid and aerodynamics, could propel aerospace engineering. New forms of propulsion were experimented with as the limitations of propellors and piston-combustion engines were reached. This research culminated in 1942 during the first test of the V2 rocket, which used the latest materials and aerospace engineering to launch the rocket to an altitude of 84.5 km at supersonic speeds. The V2 rocket used a liquid propellant engine to generate thrust, and was the first artificial object to reach space, reaching an altitude of 176 km, well above the conventional 100 km Karman line that is mostly accepted as the boundary of space.

Following the world wars and fuelled by the cold war, the space race began. Engineering and science pushed the boundaries of rocket-propelled aircraft. Designs for intercontinental ballistic missiles were modified for space faring missions, such as the first artificial satellite, Sputnik 1; launched in 1957 atop a modified R-7 missile. Multi-stage rocketry allowed vehicles to exceed previous limitations, with the Saturn V system taking humans to the Moon in 1969. Since then, the number of artificial satellites has risen exponentially and is projected to reach an estimated 100,000 [1] active satellites by 2030.

In-space propulsion has become increasingly relevant with the need for satellite manoeuvrability, station keeping, and efficient deep-space propulsion. While chemical propulsion techniques produce high levels of thrust, they are inefficient with their use of fuel [2]. Alternative propulsion techniques were pioneered in the 1960's, with Gridded Ion engines and Hall Effect Thrusters [3]. These electric propulsion devices are lighter weight, more fuel efficient, and have longer operational lifetimes than their chemical counterparts [4].

### 1.1.1 Electric Propulsion

Electric Propulsion (EP) systems operate using three key subsystems: a propellant system, a power system, and thrust system[5]. These work together to generate

a plasma and accelerate the propellant, which produces low but enduring levels of thrust. EP systems achieve specific impulse in the range of hundreds to a few thousand seconds, while chemical propulsion can only achieve specific impulse of hundreds of seconds [4, 5].

EP can be broadly classified into three classes, which differ in their approach to producing thrust. These classes are: electrostatic, electromagnetic, and electrothermal. Electrostatic propulsion systems apply electric fields on ionised gasses to induce thrust, where an external cathode is then used to neutralise the ionised gas on exit. Electromagnetic propulsion applies an electromagnetic field on ions to induce thrust and does not rely on an external cathode to neutralise the expelled ions. Electrothermal propulsion systems are hybrid chemical and EP systems to convert thermal energy into kinetic energy which induces thrust via an expelled gaseous fuel[3].

The Microwave Electrothermal Thruster (MET) is an electrothermal EP system being developed at Kyushu University [6]. It uses microwave energy to generate plasma in a cavity through collisions between free electrons and heavy particles, which is used to heat a propellant and expelled as thrust[2, 6].

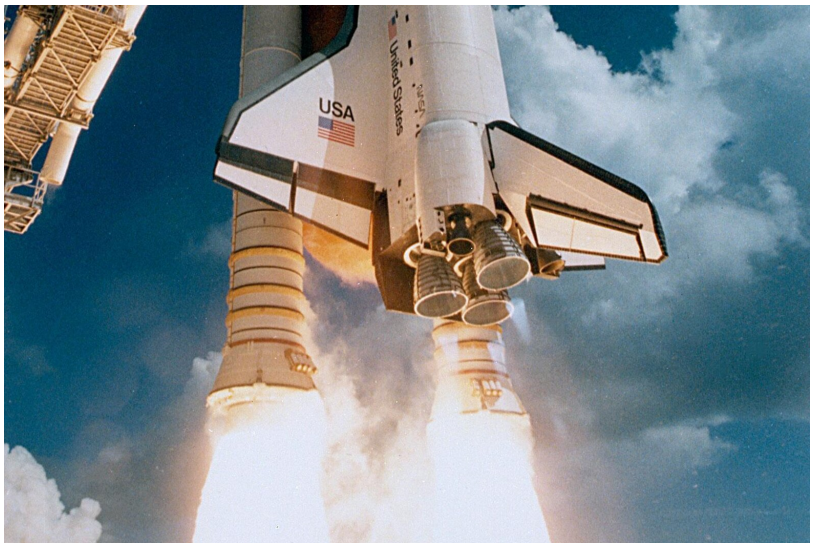


FIGURE 1.1: The Space Shuttle rocket engine during takeoff, with nozzles on boosters and the RS-25 main engines. (Image from: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/Space_Shuttle/Last_Space_Shuttle_mission_launch2

However, all these systems require a way to direct and manage their exhaust. Geometrical devices can be used to direct, compress, and expand fluid flow from the combustion/heating chamber and expel out in a usable fashion. While physically nozzles may vary in design, they all operate on the same basis; nozzles utilise Bernoulli's principle to accelerate and increase the velocity of the fluid that passes through them. This, in turn, leverages the exhaust from chemical or EP systems into usable thrust.

## 1.2 Computational Fluid Dynamics

CFD relies on the mathematical foundations laid down by Euler, whose equations described the flow of an ideal frictionless fluid, Navier, who introduced viscosity to Eulers Equations and generalised them to real fluids, and Stokes, whose equations linked viscosity to strain in a fluid. It is within the world of *Navier-Stokes* Equations that CFD lives. However, these Equations are generally intractable except for idealised and trivial one and two-dimensional flows within very simple geometries. The finite difference methodology for the manual computation of fluid dynamic problems allowed some aspect of real world problems to be analysed, but it was not until electronic computing became available that modern fluid dynamic solvers evolved into what we know today[7].

Many problems that plague CFD are very difficult to faithfully account for within the aforementioned equations, and so different models have been built which aim to approximate and capture the essential workings. While some of the earliest applications of CFD were of an engineering interest, the first was done with a regard to the physics within the Los Alamos National Laboratory (Harlow, 1957)[7]. The needs of each application of CFD have derived different methods of providing a solution, but all CFD problems first require a domain in which the problem be described and solved.

### 1.2.1 Meshing

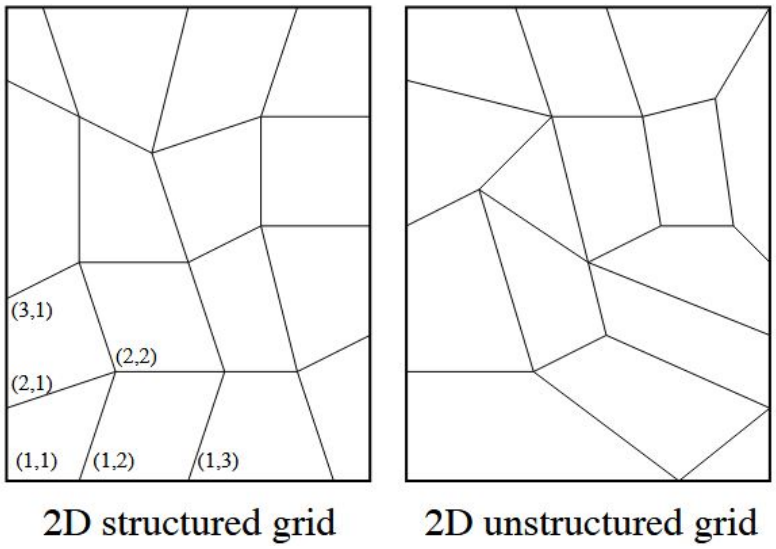


FIGURE 1.2: Different mesh construction techniques. (Image from: [8])

In order to complete any kind of simulation, a mesh must be created that represents the problem which we intend to solve. This *mesh* can be both 2-dimensional and 3-dimensional, and represents where fluid may flow. The mesh can be more simply defined as a grid of nodes, with lines connecting the nodes. The resultant

cell geometry can be structured or unstructured: a structured mesh contains quadrilateral cells and may be referred to as a topological parallelpiped grid, while an unstructured mesh may contain any kind of polygon. While unstructured grids may be appealing and easy to build for complex geometries, structured grids are preferable for their ease of use within the CFD mathematical framework owing to their numerical efficiency[8].

While there are two methods embedded within the CFD system that pertain to how the mesh represents the fluid dynamics problem, the mesh is consistently treated as the fluid flow domain with each cell being described as a control volume. The edges of the mesh, where the cell grid ends, are where we define *boundary conditions*. These may represent a rigid wall or solid surface which the fluid will interact with, an outlet where the fluid is allowed to leave and is no longer simulated, an inlet through which the fluid is input into the simulated space, and sometimes an axis, around which the domain is assumed to be mirrored, but not simulated.

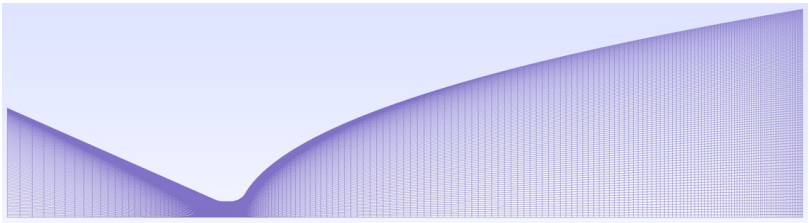

FIGURE 1.3: An example of a 2D structured, axisymmetric Converging-Diverging (CD) nozzle mesh.

By altering the boundary conditions, a single mesh can be used to simulate a variety of external and fluid conditions.

### 1.2.2 Finite Methods

Understanding the physics behind the engine is essential in creating simulated flow fields which approximate real world physics well. The mathematical system of CFD typically uses one of two core concepts to solve the flow domain described by the mesh, these being the Finite Element Method (FEM) or the Finite Volume Method (FVM). Both methods use a mesh to discretize the problem. The working principle of FVM forces conservation of mass, where the mass flow into each cell is equal to the mass flow out. This is done by: estimating the flow velocity, mass flow rate, momentum, pressure, etc, of each cell in the mesh, then calculating the aggregate mass and momentum imbalance across all cells, before solving Taylor Series linearised equations which represent the fluid flow. Only when the aggregate imbalance falls below a tolerance (or threshold) is the solution found. The set of simultaneous equations are derived from the Navier-Stokes Equations. In this way, fluid dynamics behaviour is simulated robustly for well defined problems and meshes[7, 8]. The fundamental concept behind these Equations starts with Newton's second

law, as shown in eq. (1.1).

$$F = ma \tag{1.1}$$

The Net Viscous Force/vol must be defined; $= \mu\nabla^2\vec{V}$
Thus, we can define the governing Equations for CFD as follows:
Differential Form of Momentum Conservation (in the form of eq. (1.1)):

$$\begin{aligned} -\frac{\partial P}{\partial x} + \mu\nabla^2 u &= \rho(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}) \\ -\frac{\partial P}{\partial y} + \mu\nabla^2 v &= \rho(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}) \\ -\nabla P + \mu\nabla^2\vec{V} &= \rho(\vec{V}\cdot\nabla)\vec{V} \end{aligned} \tag{1.2}$$

We can then define an Integral Form of Momentum Conservation:

$$\int_S \rho(\vec{V}\cdot\hat{n})dS = -\int_S P\hat{n}dS + \vec{F}_{visc} \tag{1.3}$$

Following this, we then must define a Differential Form of Mass Conservation:

$$\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1.4}$$

Which allows us to define the Integral Form of Mass Conservation:

$$\int_S (\vec{V}\cdot\hat{n})dS = 0 \tag{1.5}$$

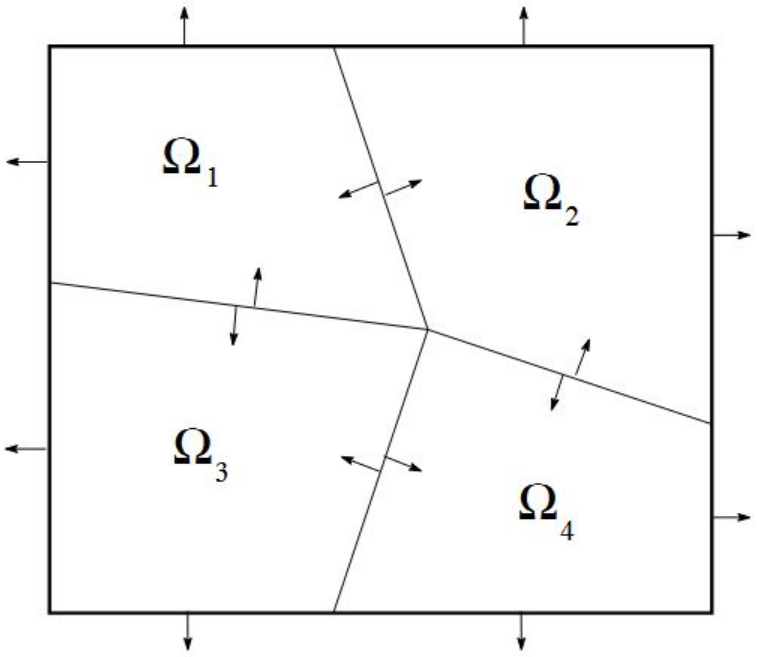


FIGURE 1.4: Conservation of cell properties using FVM on a simple mesh. (Image from: [8])

These are the fundamental equations for CFD fluid flow and are solved for every cell over each iteration. However, these equations are non-linear and are computationally too expensive to solve fully. Some assumptions must be made using this model in order to reduce these non-linear equations. These assumptions depend on the specific software used, but often include a steady flow assumption, incompressibility of the fluid, and Newtonian behaviour. Thus, FVM is solved numerically for each cell. Conservation is built in to FVM, illustrated by Figure 1.4, as each cell is solved by forcing conservation before solving. Error is introduced by interpolating cells to single points of information describing the whole cell, including cell edges and boundaries. Finer resolution of the mesh reduces this induced error but increases the solve time[8].

## 1.3 Multi-Objective Design Optimisation

MDO is fundamentally the sport of balancing competing objectives when designing systems or components. In the context of rocketry and nozzle design, thrust and fuel consumption are competing interests that are desirable to respectively maximise and minimise simultaneously. The problem arises that one conflicts with the other; to generate more thrust, you may increase the amount of fuel you use (per second). In producing solutions which attempt to optimise both objectives simultaneously, a set of solutions which may trade one objective for another will be produced and is known as a set of Pareto-optimal solutions.

Due to the complexity of solving MDO problems of many dimensions, Evolutionary Algorithm (EA)s have been adopted into the method for searching and solving MDO problems; as opposed to single-objective problems in which EAs would be unnecessarily complex for the solving of such problems. EAs are stochastic search techniques that mimic biological processes to find the best solution to a problem and can be thought of as an algorithmic replication of natural selection and genetics. Terms used in EAs are interchangeable with natural biological processes; iterations are commonly referred to as generations, *etc.* As such, EAs use three fundamental operations: selection, crossover, and mutation. Selection guides which individuals in a population are best to preserve into the next iteration, while crossover *breeds* two selected individuals and merges their properties, and mutation may take any individual and randomly change one parameter as it transfers between iterations[9].

Classical calculus-based optimisation algorithms use derivative information to gauge their performance, while EAs use direct search methods to find results. This results in a slower but more robust optimisation method which can be easily translated across disciplines. EAs are further set apart by their population approach, which allows them to search multiple solution fronts simultaneously using multiple individuals during each iteration. Multiple solution fronts are further explored fully by EAs using stochastic operators, as oppose to deterministic ones. Advantages of

this random sampling and search method are varied; most prominently, they are simple to implement and have diverse applicability[10].

#### Surrogate Assisted Evolutionary Algorithms

A Surrogate Assisted Evolutionary Algorithm (Surrogate Assisted Evolutionary Algorithm (SAEA)) is a subset of EAs which aims to bring the computational cost down. Many EAs are conducted on computationally expensive problems, like CFD. This impediment has forced the implementation of computationally cheap, approximate models to be integrated into a standard MDO search. These "surrogates" can be built using data from all previous generations of an EA search, and are used to predict performance characteristics of systems without actually conducting a test *i.e.* conducting a CFD simulation. This must periodically be checked against the true test, with the surrogate models being rebuilt should they fall out of reasonable margins[11].

# Chapter 2

# A Review of Current Literature

## 2.1 Laval Nozzles

Nozzles first saw widespread adoption during the industrial revolution in steam engines. Most famously, Gustaf de Laval designed a CD nozzle which has since become synonymous with his name[12]. CD nozzles work because the mass flow rate of fluids is constant through a constrained path. CD nozzles have three sections which describe them: the convergent and divergent sections, and the throat. As the fluid moves through the converging entrance to the nozzle, it is compressed until it reaches the throat. Speeds of fluid through the convergent portion are subsonic, where the nozzle will accelerate it to keep the mass flow rate constant. At the throat it will locally reach sonic speeds. Once through, it will continue to accelerate at supersonic and evolve through the diverging portion of the nozzle, which shapes and guides the exhaust and is crucial in design for such operations[5]. All stages of the CD nozzle have various parameters which dictate the behaviour of the flow through them, such as length, curvature, and rate.

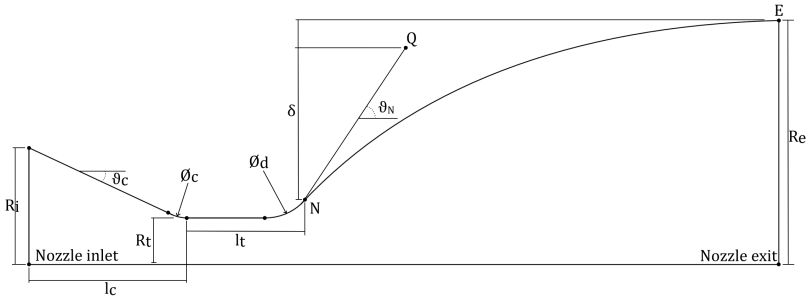


FIGURE 2.1: Parabolic nozzle geometry of the divergent section of a nozzle, indicating the nozzle inlet ($R_i$), throat radius ($R_t$), and the outlet or 'Nozzle exit' is the exit radius ($R_e$).

Previous work as early as 1944 by Kisenko[13] had determined the minimal effect of the convergent section on the development and performance of a nozzle, leaving the throat and divergent portion of CD nozzles as critical components which determine the performance of a rocket engine. Thus, work pertaining to the optimisation of this component has been extensive. Rao developed a method for producing a thrust optimised contour nozzle, which generally followed the profile of a canted parabola[14, 15]. Rao's work focused on the complete expansion of the

exhaust gasses and parallel, uniform, flow at the nozzle exit for the respective ambient operating pressure and achieved this using a Method of Characteristics (MOC) approach.

MOC was used to optimise the fluid flow through the nozzle. MOC is a method for solving a given set of nonlinear equations numerically. First, *characteristic* lines are found in the flow field which have properties that can be modelled using analytical techniques. The characteristics of fluid flow are coincident to the Mach lines. They are used to find the compatibility relation and are key to the computational method[16].

While MOC was used extensively for nozzle design[14, 15], which produced mathematically optimal nozzle profiles, complex fluid dynamic effects and real performance losses are difficult to truly predict and model. Further, only inviscid flow was contemplated, which is an idealised state of fluid dynamics useful for mathematical formulations and models but unrepresentative of real world fluid flows. This is where testing and experimental verification of nozzle design is required; today this is primarily achieved using CFD simulations. Evans *et al* and Candler *et al* contributed to developing CFD simulation processes to incorporate the full Navier Stokes equation sets, introducing a framework which could solve viscous fluid flow in the context of rocket nozzles[17, 18]. From this, Cai *et al* developed a methodology to optimise nozzle contours using CFD and EA[19]. These optimisation processes showed a 1.501% performance improvement in the contour nozzle performance, which they attributed to a decrease in frictional and axial losses. These techniques are still relevant, with nozzle configurations being proven using a combination of CFD and EAs, more broadly as MDO[20, 21].

Different kinds of CD nozzles can be made and each have their own unique advantages which lend them to certain capabilities. The naming of CD nozzles typically refers to the contour and profile when looking at the cross section, often specifically the divergent portion. While conical CD nozzles use exclusively straight lines and are easy to construct, they are fixed and cannot compensate at changing altitude. Dual-bell CD nozzles have unique atmospheric compensating abilities but are typically bigger and heavier as a result. Parabolic nozzles, sometimes referred to as bezier, hybrid, or Rao nozzles, are specialised nozzles that are more difficult to design and construct but offer better performance than that of similar conical designs[5].

## 2.2 Micro Nozzles

Micro-nozzles are defined as nozzles whose throat radii dimensions fall somewhat in the micrometre ($\mu$m) to millimetre (mm) range. Compared to research on regular (metre sized) nozzles, micro-nozzles suffer from viscous boundary layer effects[22, 23]. Viscous boundary layer effects are a key component which drives design of

micro-nozzles and have been shown to deviate the optimal nozzle geometry of conical nozzles from their metre scale counterparts[24]. Suenaga *et al* found that the optimum nozzle divergence half angle is classically 15°, but at the mm scale was found to be in the range of 22°to 29°.

All nozzle profiles are subject to losses which reduce their theoretical output. These can be summarised by: flow divergent losses, expansion ratio losses, boundary layer and wall friction losses, two-phase flow losses, chemical losses, throat erosion, and real gas properties. Many of these are fractional compared to the total power output from rocketry nozzles, but combined can make significant contributions to the difference between a theoretical maximum and the actual performance[5]. Further, on small scale propulsion systems like EP, these losses can be more extreme; boundary layer losses are more pronounced at small scales[24].

$$Re = \frac{\rho\, u\, L}{\mu} \tag{2.1}$$

A measure of flow dynamics which is particularly relevant within the micro nozzle domain is the Reynolds number. First described by Stokes, this number describes whether the flow is turbulent or laminar, and is a ratio of the inertial to viscous forces. This relation is described by eq. (2.1), where $L$ is the characteristic length. Small Reynolds numbers (in the order of $10^2$) may allow for the simplification of eqs. (1.2) and (1.5), where fluid flow is generally described as laminar[25].

## 2.3 Motivations for this Research

There does not exist any work which aims to optimise a micro-nozzle with a parabolic divergent contour. While much work has been done at the metre scale, the increasing demand for small-scale satellite propulsion solutions necessitates an optimised nozzle to maximise the thrust capabilities of the engines used to propel them. An optimised nozzle contour will maximise thrust while minimising the mass flow rate and plume divergence. The difference between an optimal nozzle contour at the metre scale and the millimetre scale is of interest to this study, whereby the aim is to understand the complex multi-modal physics in reduced nozzle dimensions.

This study will therefore produce a parabolic nozzle contour and subsequently use it to find an optimum profile at the millimetre scale. This will be done through use of a MDO framework and advanced EAs. The resultant geometries will then be compared and the key geometric characteristics determined by the scale-dependent physics for the nozzle examined.

# Chapter 3

# Methodology

## 3.1 Analytical Geometry

Defining the geometry of the divergent portion of the CD nozzle (and more broadly the convergent and throat portions) is an important step in the broader process of optimisation. A useful feature of CD nozzle designs is their axisymmetric nature, meaning that they can be defined around some central axis and revolved to create a three-dimensional object. This simplifies the process in designing a nozzle, since only a two-dimensional profile, or contour, can be defined and used to create the whole. Nozzle contours are a popular way of expressing nozzle shapes and dynamics; being two-dimensional, they are simple to construct and can be used by CFD software for simulation purposes and Computer Aided Design (CAD) software for fabrication purposes. With some reasonable constraints and needing to only define a few constant parameters, a system can be devised which allows any number of contours to be made which are related in their profile only. This is the principle behind nozzle contour optimisation; by choosing a few defining geometric features of the mathematical frameworks describing a nozzle contour, different geometries can be produced and tested against one another and the optimum found.

### 3.1.1 Nozzle Geometry

In order to create a nozzle, a mathematical framework must be produced that describes the contour. The contour must be continuous and tangential and may be defined radially or through a piecewise function. The software that will be used to generate the mesh uses a nodal framework, in which points are defined by nodes and lines drawn between them. Considering this piecewise-like approach, the methodology [15] proposed for the construction of a parabolic nozzle may be used. Perhaps the most essential and mathematically dense sections, the parabolic region can be described with reference to only a few key characteristics. Table 3.1 outlines the geometric control parameters for the entirety of the nozzle; the decision variables $\theta_N$, $\theta_E$, and $\epsilon$ are the key characteristics which define the parabolic profile, and describe the nozzle throat angle, the nozzle exit angle, and the expansion ratio between the nozzle throat and exit respectively.

TABLE 3.1: Geometric control parameters for the nozzle contour definition, their symbols, units, and variable type.

| **Geometric Definition** | **Symbols** | **Unit** | Variable Type |
|---|---|---|---|
| Nozzle throat radius | $R_t$ | mm | Decision |
| Expansion ratio | $\epsilon$ | - | Decision |
| Parabolic angle N | $\theta_N$ | ° | Decision |
| Parabolic control ratio | $\delta$ | - | Decision |
| Converging circle radius | $\varnothing_c$, $R_c$ | mm | Decision |
| Diverging circle radius | $\varnothing_d$, $R_c$ | mm | Decision |
| Nozzle throat length | $l_t$ | mm | Constant |
| Nozzle inlet radius | $R_i$ | mm | Constant |
| Nozzle convergence angle | $\theta_c$ | ° | Constant |

Using these definitions, the nozzle geometry may be constrained in useful ways and extract the definitions needed. For example, by defining a throat radius, $R_t$, and expansion ratio, $\epsilon$ (which is typically defined as $\epsilon = \frac{A}{A^*}$), a solution can be found for the exit radius, $R_e$.

$$R_e = \sqrt{\epsilon} R_t \tag{3.1}$$

Given eq. (3.1), the length of the nozzle can be defined. The perfect nozzle (100% nozzle) is only 0.2% more efficient than a 85% nozzle, and given weight limitations on spacecraft, many manufacturers choose an 80 to 85% nozzle instead, such that the nozzle length is 80% the length of the conical equivalent, hence tan 15 represents the optimum conical nozzle length [5, 12]. Thus:

$$L_N = 0.8 \frac{\sqrt{\epsilon} - 1}{\tan 15^\circ} R_t \tag{3.2}$$

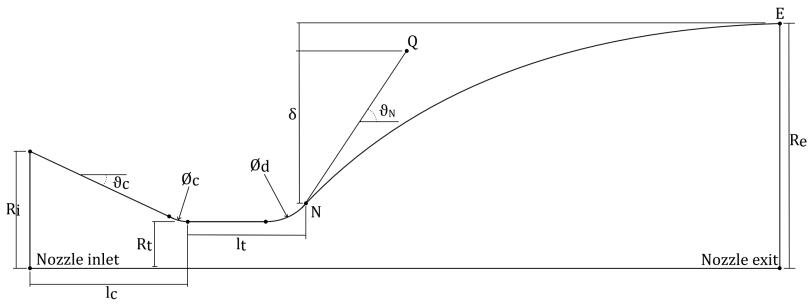


FIGURE 3.1: Parabolic nozzle geometry of the divergent section of a nozzle, where the nozzle inlet is described by ($R_i$), throat radius is described by ($R_t$), and the outlet is described by the exit radius ($R_e$).

Figure 3.1 illustrates the parabolic nozzle geometry. Points N, Q, and E are three nodes which are used to define the bézier curve which describes the parabolic region of the contour. These point's coordinates can be found analytically using the

geometric definitions from Table 3.1, using eqs. (3.1) and (3.2), and a set of parametric Equations defining the radii smoothing function which joins the parabola to the nozzle throat, as given by eq. (3.3). These defining parametric Equations are reliant on the throat radius and the nozzle exit angle, $\theta_N$.

$$\left.\begin{aligned} x_1(\theta_N) &= \varnothing_c R_t \cos\theta_N \\ y_1(\theta_N) &= \varnothing_c R_t \sin\theta_N + \varnothing_c R_t + R_t \end{aligned}\right\} -90 \leq \theta_N \leq (\theta_N - 90) \tag{3.3a, 3.3b}$$

These points are further reliant on the value of $\theta_E$, which is not yet defined. $\theta_E$ is essential in the definition of the parabolic region, describing the angle of depression from the nozzle exit to the point Q. However, it is dynamically constrained by the value of $\theta_N$; should $\theta_E$ be too big, it may cause the location of the point Q to fall past point N, resulting in infeasible geometry. Thus a different control parameter may be described for the geometry, $\delta$. This unitless value describes the ratio between the vertical placement of points N and E, on which point Q is placed upon the ray drawn from point N at the angle $\theta_N$. Figure 3.1 illustrates this relationship well. As such, the point Q can be defined in our system robustly. The control points, $(N_x, N_y)$, $(Q_x, Q_y)$, and $(E_x, E_y)$ are defined as such:

$$\begin{aligned} N_x &= x_1(\theta_N - 90) \\ N_y &= y_1(\theta_N - 90) \\ E_x &= L_N \\ E_y &= R_e \\ Q_x &= \delta\frac{(E_y - N_y)}{\tan\theta_N} + N_x \\ Q_y &= \delta(E_Y - N_y) + N_y \end{aligned} \tag{3.4}$$

The bézier of the parabolic region of the contour is then typically defined using eq. (3.5). However, the nodes are sufficient for the meshing software to produce its own curve.

$$\left.\begin{aligned} x_2(t) &= (1-t)^2 N_x + 2(1-t)tQ_x + t^2 E_x \\ y_2(t) &= (1-t)^2 N_y + 2(1-t)tQ_y + t^2 E_y \end{aligned}\right\} 0 \leq t \leq 1 \tag{3.5a, 3.5b}$$

### 3.1.2 Mesh Generation

The mathematical framework described above was then translated into Python, a general purpose programming language. Python has various packages which can be installed that increase its functionality, many of which are relevant to this study. As such, Python can be utilised to build an integrated system which may accomplish many aspects of the project. One such package is GMSH, an open source meshing

solution[26]. This package connects to a finite element meshing software; Python can be used to calculate key vertex geometric information, and GMSH used to build and construct the lines and mesh around it.

GMSH has the capability to refine the mesh along line elements. Natively referred to as bias, refinement of mesh cells is an important aspect of CFD simulation as it allows for areas of interest or great fluid behaviour change to have more resolution. The nozzle throat, walls, and exit interface are such areas of interest, owing to the more complex physics in those regions. GMSH supports two types of refinement, *progression* and *bump*; these refine the element towards one side, or both sides respectively. The progression refinement follows a simple exponential increase in cell length from start to finish, while the bump refinement is defined by an "ad hoc" function. In order to properly utilise the bump progression when constructing the parabolic mesh, this "ad hoc" function had to be reverse engineered and can be found in appendix A.

By design, the mesh of the nozzle geometry can be constructed with arbitrary values for the decision variables outlined in Table 3.1. However, the geometry must be constrained with some constants at various aspects. These constants and their values are outlined in Table 3.2.

TABLE 3.2: Geometric constant parameters.

| Geometric Constant | Value | Unit |
|---|---|---|
| $l_t$ | 0.51 | mm |
| $R_i$ | 3.8 | mm |
| $\theta_c$ | 21.6 | ° |

The inlet and throat parameters have been constrained, following research suggesting that the nozzle inlet geometry has little influence on the development of the flow from a nozzle[13]. The values chosen reflect previous work on nozzles of the mm scale[24], but remain flexible enough that nozzles of any scale would not be jeopardised.

The code that generates the mesh file can be found in appendix A.

### Geometric Constraints

Part of designing the geometry is constraining the values which drive it. The decision variables which will drive the optimisation of the nozzle geometry must be constrained in order to ensure feasible geometries are always produced. While some values are constrained to ensure the nozzle contour is always continuous, not self-intersecting, and otherwise physically viable, others are constrained for their practicality; $\epsilon$ is constrained to ensure the geometry does not grow excessively in order to increase thrust which ensures that the nozzle does not get exceptionally large. $\varnothing_{c,d}$ is constrained given the initialisation and positioning is contained within the nozzle

throat definition, which is capped and outlined in Table 3.3. Table 3.3 outlines the decision variables and their initial respective ranges. The ranges define the search space for the geometric optimisation and have been chosen (if not for their inherit geometric feasibility) for their broad appeal to make a multitude of feasible parabolic nozzle geometries.

TABLE 3.3: Initial geometric constraint parameters and their upper and lower limits used for the parametric study.

| Geometric Definition | Decision Variable | Range | Unit |
|---|---|---|---|
| Throat radius | $R_t$ | 0.5 - 2.0 | mm |
| Expansion ratio | $\epsilon$ | 50 - 200 | - |
| Parabolic angle N | $\theta_N$ | 20 - 40 | $^\circ$ |
| Parabolic control ratio | $\delta$ | 0.7 - 1.0 | - |
| Throat smoothing radius | $\varnothing_c$ | 0.6 - 1.0 | mm |
| Throat smoothing radius | $\varnothing_d$ | 0.6 - 1.0 | mm |

The parameters and their limits described by Table 3.4 have been chosen in order to produce consistent simulations. Two methods of testing the sensitivity of geometric parameters on the solution outcomes were utilised. Appendix B outlines the methods and their results in greater detail. This analysis was critical in producing consistent and realistic fluid flow and behaviour through the nozzle in the simulations, as good simulated fluid behaviour is a strong indicator of high confidence in the produced results. The first method combined a sensitivity analysis using a set of cases generated using LHS (see section 3.2.2) and a random forest regression model to predict geometric bounds which would guarantee consistent and realistic fluid flow behaviour. The second method conducted a brief sensitivity analysis on a stable geometric configuration (one that had proven to generate desirable fluid flow behaviour), where all parameters were held constant and one varied such that safe simulation bounds for the design space were found. It was found that $\theta_N$ and $\epsilon$ had a strong correlation with the convergence and success of a simulation. In particular, $\theta_N$ was found to not allow simulations to converge when too large; $\theta_N$ values above 25.5° impeded simulation convergence. Very small values of $\varnothing_{c,d}$, around 0.1 mm also impeded convergence. Further, the implicit value of $\theta_E$ (the angle of depression from the exit to the $\delta$ control node, see Figure 3.1) was also found to impede convergence. Values below 6.5° resulted in failed convergence and were therefore measured and cut from the study.

TABLE 3.4: Finalised geometric constraint parameters and their upper and lower limits used for the parametric study.

| Decision Variable | Range | Unit |
|---|---|---|
| $R_t$ | 0.5 - 1.2 | mm |
| $\epsilon$ | 60 - 120 | - |
| $\theta_N$ | 20 - 25.5 | $^\circ$ |
| $\delta$ | 0.7 - 0.95 | - |
| $\varnothing_c$ | 0.85 - 1.0 | mm |
| $\varnothing_d$ | 0.85 - 1.0 | mm |

## 3.2 Parametric Study

A parametric study can be conducted to analyse the effect of different design variables (or parameters) have on resultant properties. In the case of nozzle design, a parametric study may analyse the effect of changing geometric parameters have on the resultant thrust produced. In conducting a parametric study, different elements can be inspected and thoroughly investigated prior to the optimisation. The parametric study may reveal a parameter which has no effect on the result; this would thus allow for the complexity of the design optimisation to be reduced, ensuring that the scope of the optimisation remains grounded and realistic, whilst also reducing solve time. Thus, the sensitivity of the parameters to the result can be identified.

The parametric study for the parabolic nozzle geometry will be conducted by varying the elements outlined in Table 3.4. Resultant geometries will be analysed using ANSYS Fluent, a CFD software, with thrust, specific impulse, and divergence of the exhaust plume investigated. Geometries which maximise the thrust, specific impulse, and the efficiency of the plume divergence will be rated highly.

### 3.2.1 CFD Settings

In order to obtain these results, the nozzle geometries and meshes must be analysed. ANSYS Fluent is the CFD software which will be used, because: proven history in completing accurate simulations of nozzles of various scales, powerful fluid flow models that integrate into the finite element analysis methods prescribed to cell based mesh simulations, ability to simulate different fuels and ambient pressure profiles which match the operational use-case of the nozzle being designed, capability to monitor relevant output data, and expansive automation library integrated into the Python framework.

#### Flow Conditions

The CFD simulations will be conducted in a two-dimensional, axisymmetric plane to reduce complexity. This can be achieved because the nozzle is an axisymmetric

object, meaning that it can be described in three-dimensions by using only the profile, with all flow fields being similar around the axis. Further, the simulation will be conducted using a density based implicit simulation scheme. This means that the simulation will be calculated using the density of the fluid moving through cells to achieve convergence, as opposed to using the pressure at different cells. Density based simulations are typically faster to calculate, owing to the reduced complexity of density based particle physics compared to pressure based. The simulation will also have energy Equations enabled, which ensures that heat convection and diffusion are incorporated. Another assumption that can be made is that of its viscous model; due to the supersonic nature of fluid flow through nozzles, laminar flow can be enforced safely by definition. Thus, the cell Reynolds Number (described using eq. (2.1)) is in the order of $10^1$ to $10^2$ within the nozzle domain.

The MET engine, which the nozzle will be optimised for, is flexible in choice of fuels and may be easily adapted to use water vapour or Argon. Water vapour is appealing for its abundance and ease of transport, but is chemically complex for CFD simulations[27]. Therefore, in order to minimise this uncertainty, Argon gas, a neutral, noble gas, may be used. The simulation assumes that the Argon acts as an ideal gas, using the Sutherland viscosity model and a thermal conductivity kinetic theory to approximate how it might behave.

### Boundary Conditions

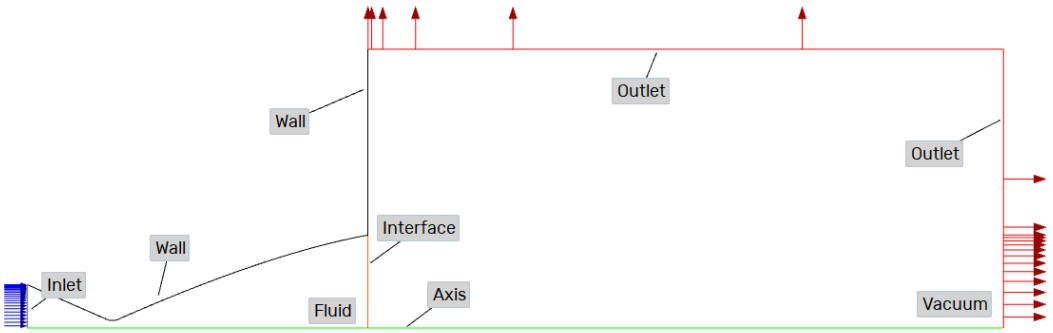


FIGURE 3.2: Parabolic nozzle geometry mesh with boundary names established and fluid/vacuum regions described.

To conduct CFD, boundary conditions must be established. These represent the environment that the simulation is placed in and how it connects to it. Boundary conditions used in the simulation of a nozzle include: an axis, a wall, an inlet, and an outlet. The axis has already been established as the point at which the simulation is assumed to be mirrored around. The wall boundary condition represents a surface on which the fluid may interact. The wall is assumed to be made of aluminium material. The nozzle inlet is situated at the entrance to the nozzle, with the conditions here describing the interior of the engine which the nozzle is connected to. The inlet passes in the fluid to the simulation, the Argon fuel, at a temperature of 1000 K, and

an initial gauge pressure of 50,000 Pa. The outlet is situated at the end of the nozzle, where the domain is assumed to be the ambient outside environment conditions. For this nozzle, the operating conditions will be in Low Earth Orbit (LEO) and so a near vacuum state will be assumed. A gauge pressure value of 13 Pa has been set in the vacuum region.

### Convergence Conditions

Convergence of a CFD simulation is not an easily measurable quantity. Residual values describe the average change of the governing CFD equations (see eq. (1.2)) between iterations and generally guide whether a solution is stable. While providing a good reference for simulation quality, more must be done to ensure a simulation exhibits expected and real behaviour. An important quantity of interest for the nozzle simulation is the flux of the mass flow through the inlet and outlet. This quantity should be constant, given the constraints imposed by the governing equations to ensure constant mass flow through all cells. Thus, by measuring the difference between the inlet and outlet we can determine when a simulation is behaving in a steady-state and may be deemed converged. An arbitrary value of 0.1 % may be enforced to ensure the convergence of a simulation.

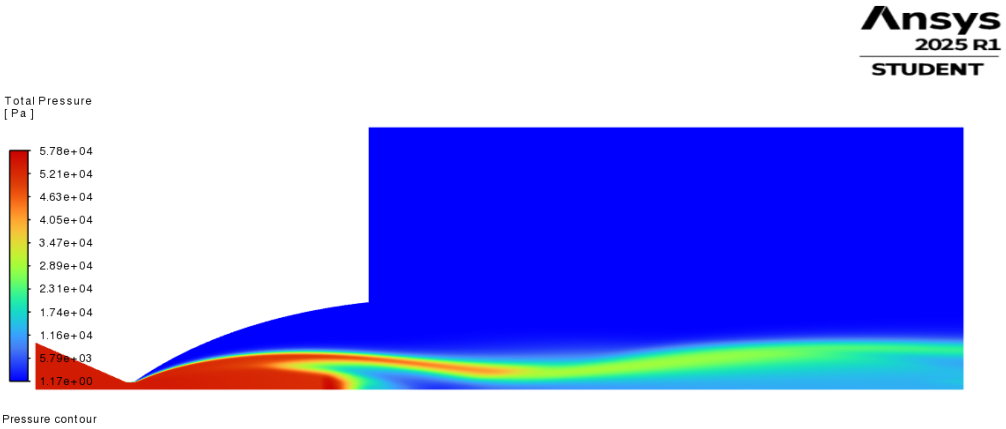


FIGURE 3.3: An example of flow splitting, a symptom of a poorly converged simulation which is best identified via the pressure contour.

Further, it is important to inspect the flow fields when a simulation has converged to ensure realistic and expected flow behaviour. CFD software may find local minimums of the convergence conditions and governing Equations that do not represent realistic flow, which may have been achieved following approximations and assumptions. By inspecting the pressure, Mach number, total velocity, and temperature flow fields using contour plots, convergence can be confirmed or challenged. Flow splitting, illustrated in Figure 3.3, is a common symptom of a poorly converged simulation and is unrealistic fluid behaviour. Whilst similar to what may

be observed during a shock, flow splitting does not display characteristic pressure profiles which have characteristic shape, or have corresponding Mach number flow fields which indicate shock formation.

#### Pyfluent Implementation

Appendix A describes the Pyfluent code which performs the ANSYS Fluent simulation. Pyfluent is a Python package which allows seamless Pythonic integration into the ANSYS Fluent workspace. This package allows for a repeatable, scriptable workflow surrounding similar CFD simulations. Despite similarities to the native Fluent journal system, Pyfluent allows external data to be imported and changed, allowing for many simulations to be ran in succession with desired changes (*i.e.* changing the mesh to be imported, controlling desired parameters like inlet temperature, pressure, determining the output file names) in one cohesive script. As such, the simulation execution can be integrated into the existing mesh-generation workflow.

### 3.2.2 Base Case Validation

Prior to completing the sensitivity analysis, a crucial step is ensuring that the constructed geometry and simulation conditions produce consistent simulations. This can be done via a set of base case validations. By producing a limited set of geometries and passing them into Fluent for a CFD simulation, a systematic performance check can be produced which, should the entire set pass, should ensure a high success rate of simulations when conducting the sensitivity analysis. This check is important in ensuring confidence in the new mesh geometry.

The base case validations may also be used to refine geometric constraints. Should simulations consistently fail to converge despite satisfactory meshing (proven by some geometries converging well) an analysis may be performed on which elements are hindering convergence. It was found that a high value of $\theta_N$ (values above 25.5°) hindered convergence. This analysis may be performed using the set of base cases, which may be constructed using geometric constraints chosen which cover approximately the entire search domain. This may be done through the use of LHS.

#### Latin Hypercube Sampling

Latin Hypercube Sampling LHS[28] is a random number generation system which ensures each column exhibits values across its entire range of values. A LHS is a $M \times N$ matrix, where $N$ is the number of input parameters used to construct the geometry of the nozzle (there are five decision variables, thus $N = 5$), and $M$ is the number of geometries to be constructed. LHS ensures the entire search space is covered, as opposed to a generic random number generation which has no awareness of other values being produced across other dimensions (columns). During the base case

validation, an $M$ value of five, then twenty, was chosen to validate the mesh and simulation convergence. All geometric constraints (see Table 3.4) were represented across their entire range of values within these five and twenty cases. To conduct the sensitivity analysis, $M = 400$ was chosen to ensure appropriate coverage of all geometric combinations and nuance.

### 3.2.3 Sensitivity Analysis

A sensitivity analysis is conducted to assess the capacity each geometric parameter has on a measured outcome. The Sobol variance decomposition model [29] requires the input parameters to be generated via LHS, which ensures every configuration of values has even distribution. This variance based global sensitivity analysis calculates first-order and total effect indices, $S_i$ and $S_{Ti}$ respectively. $S_i$ is a measure of an inputs individual influence to the measured outcome and how the variance of the measured outcome is tied to this, while $S_{Ti}$ measures the overall variance of the measured outcome as evaluated by the parameter and its combined effect with all other joint geometric values. This in essence is a measure of the weight a geometric parameter has on the measured outcome and how it interacts with the other geometric parameters.

The Sobol variance model and sensitivity analysis can be conducted using geometries and outputs generated by a surrogate model. These models will be trained on the data produced by the geometries constructed and tested by the LHS of dimension $M = 400$. 90% of the produced cases will be used to train the surrogate model, while the remaining 10% will be used to validate and test the model for its ability to predict measured outcomes. The sensitivity analysis may be conducted across three measured outcomes of interest: the thrust generated, the divergence efficiency, and the specific impulse of the system. Each analysis must be run separately, and will each have a unique makeup of parameters which influence the outcomes.

The sensitivity analysis may also be run on the raw data produced from the LHS geometries. This data, being unstructured, may not be tested via Sobol's method, and must instead be investigated via a local sensitivity function and reduced global analysis. This analysis using the Delta sensitivity method can be conducted using Python's SALib package, which requires a matrix with the input geometric parameters and an evaluation of each quantity being measured. This may be compared to the Sobol variance sensitivity analysis, which will be conducted within a MATLAB program provided by Kyushu University and the STSEL laboratory.

#### Performance Measurement Technique

Laval nozzles are well understood systems in the classical sense, and thus have many convenient Equations which may be used to characterise quantities of interest, like thrust, mass flow rate, or exit velocity. Thrust is described by eq. (3.6), where $\dot{m}$

is the mass flow rate, $V_e$ is the exit velocity, $p_{e,o}$ is the exhaust and ambient pressure respectively, and $A_e$ is the exit area.

$$F = \dot{m}\, V_e + (p_e - p_o) A_e \tag{3.6}$$

The mass flow rate may be described separately, using eq. (3.7). $A^*$ represents the area of the throat, $p_t$ is the total pressure, $T_t$ is the total temperature, $\gamma$ is the specific heat ratio of the exhaust, and $R$ is the gas constant.

$$\dot{m} = \frac{A^*\, p_t}{\sqrt{T_t}} \sqrt{\frac{\gamma}{R}} \left( \frac{\gamma + 1}{2} \right)^{-\frac{\gamma+1}{2(\gamma+1)}} \tag{3.7}$$

Finally, the exit velocity of the exhaust can be described using eq. (3.8). $M_e$ is the exit Mach number and $T_e$ is the exit temperature.

$$V_e = M_e \sqrt{\gamma\, R\, T_e} \tag{3.8}$$

These three Equations are generally useful for describing exhaust flow through a Laval nozzle and may be used to analyse, verify, and explain results obtained from fluid simulations of exhaust fluid flow.

While ANSYS Fluent is capable of providing estimates of some physical values, such as the mass flow rate at the inlet and interface, more advanced quantities must be calculated post process. ANSYS Fluent can provide cell-based values for quantities such as pressure, density, axial and radial velocity, and temperature. While individually these values hold little value, the integration of them across the boundary of interest results in the effective value of interest. Thus, thrust, axial efficiency, and the specific impulse may be derived from eqs. (3.6) to (3.8) and then calculated, alongside a more accurate value for the mass flow rate of a system, using eq. (3.9):

$$\begin{aligned} \dot{m} &= 2\pi \int_0^{r_e} \rho (v \cdot \vec{n})\, dA \\ T &= 2\pi \int_0^{r_e} (\rho u^2 + p) r\, dr \\ T_{axial} &= 2\pi \int_0^{r_e} (\rho u_{axial}^2 + p) r\, dr \\ \eta_D &= \frac{T_{axial}}{T} \\ I_{sp} &= \frac{T}{\dot{m}} \end{aligned} \tag{3.9}$$

where $\vec{n}$ is the normal of the plane through which the fluid is passing, $p$ is the static pressure, $u$ is the fluid velocity, and $r$ is the radius at which the point is calculated. In general, it is useful to solve these integrals across the discrete data produced by the cell based simulation method via an application of Simpson's integration, defined by eq. (3.10):

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right] \tag{3.10}$$

where $a$ and $b$ represent cells, with $f(x)$ being the value of the quantity in question at each of those cells.

## 3.3 MDO

The primary MDO process will be completed on existing architecture provided by Kyushu University, which is built in MATLAB. This system is built following the principles of the NSGA-II algorithm, which is a SAEA. These surrogate models are trained and built on a library of existing simulation data, which in this study will be the library produced within the sensitivity analysis. From this, predictive values for values of interest may be produced based on any configuration of geometric input parameters. The architecture built by Kyushu University trains multiple models, among which the model with the highest accuracy when measured against existing input parameters and predictive outputs is adopted. Through this, further simulation of systems during the optimisation process is substituted for the surrogate model prediction, cutting computation time and cost.

### 3.3.1 Objective Functions

MDO relies on objective functions. These are characteristic of a system, such as the thrust produced by a nozzle, and are often desirable to maximise or minimise. In practice, these can all be solved as a minimisation problem, by use of a negative value on a maximising function. Thus, standard mathematical procedures can be applied for solving minimisation problems. Single objective problems have only one such function in question, and are generally trivial to solve using basic Newtonian-like methods[10]. Multi-objective functions are more difficult to solve, since they rarely follow functions which can be solved analytically using calculus based methods. These are instead solved using an EA, direct search approach. The objective functions used in this study are summarised as follows:

*Maximise:*
(1) Thrust, $T$
(2) Divergence Efficiency, $\eta_D$
(3) Specific Impulse, $I_{sp}$

These are all directly related to desirable performance characteristics of the nozzle. Objective function (1) and (3) are conflicting parameters, indicating a Pareto front will develop, describing many different nozzle geometries which have a trade-off relationship between the specific impulse of the system and the thrust produced.

Confidence in the produced results will be ensured from the simulation and the MDO process using objective function constraints, such that if any cases display unrealistic (*i.e.* not physically possible) behaviour, they are discarded.

$$\begin{aligned} \textit{Subject to:} \quad & (1)\ \text{Specific Impulse, } I_{sp} \leq I_{sp,theory} \\ & (2)\ \text{Divergence Efficiency, } \eta_D \leq 1 \end{aligned}$$

The $I_{sp}$ of a simulated system must not exceed the theoretic maximum, just as the system may not be more than 100 % efficient. The theoretic maximum $I_{sp}$ of a system may be derived using eqs. (3.6), (3.7) and (3.9), and described using eq. (3.11), where the inlet and ambient pressure, inlet temperature, flow coefficient, and nozzle geometric information is known.

$$\begin{aligned} \dot{m}_{\text{ideal}} &= \frac{p_0\, A^* \sigma}{\sqrt{R\, T_0}} \\ F_{\text{ideal}} &= \dot{m}_{\text{ideal}}\, V_e + p_e\, A_e \\ I_{sp\,\text{ideal}} &= \frac{F_{\text{ideal}}}{\dot{m}_{\text{ideal}}} \end{aligned} \tag{3.11}$$

# Chapter 4

# Parametric Study

A sensitivity analysis of the parabolic micro-nozzle has been conducted, as has a parametric analysis which varies throat radius, smoothing radius, expansion ratio, curve factor and divergent angle to maximise thrust, exhaust divergence efficiency, and specific impulse.

## 4.1 Sensitivity Analysis of the Parabolic Micro-Nozzle

Since the data was not produced via a LHS technique, a Sobol analysis was not possible. Instead, a Delta analysis was conducted; this analysis held very little confidence, so the results were discarded. The random forest method was still applicable and was generated using the sci-kit learn Python package. The random forest algorithm is at its core a best-split decision-tree classifier, which may be used to estimate and predict outcomes based on model data. The core principles are those of regression analysis.

### 4.1.1 Static Study (50 Case)

The "case" or "instance" is a single set of values for all variables, where each value is randomly chosen from its respective interval without repetition in any given dimension.

This analysis was conducted by first identifying a case which produced desirable results, where the produced pressure contours were without discrepancies and illustrated realistic fluid flow, simulation residuals were below desired targets, and convergence criteria were met. Then, all parameters were held constant while one was varied between a maximum and minimum bound. Appendix B details the input geometric parameters, alongside the output parameter values as calculated using eqs. (3.9) and (3.10).

Figure 4.1 illustrates the relationships revealed via the random forest. The random forest analysis yielded consistent results which suggested high dependence on $R_t$ on all output parameters, but most significantly on thrust and specific impulse. It was found that the model could make very accurate predictions of all output parameters given a change in the $r_c$, $\epsilon$ and $\delta$ values.

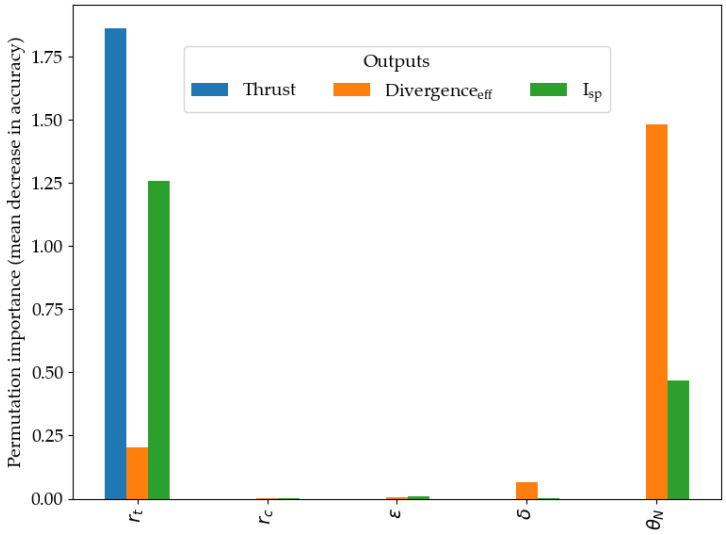


FIGURE 4.1: Random forest prediction of most important parameters for solution outcome.

$\theta_N$ was found to reduce the reliability of the models to predict the divergent efficiency and specific impulse values. As such, the values with high permutation importance as described by Figure 4.1 indicate a strong correlation between the relevant input parameter with the output parameter. These parameters may be described as complex parameters, which may yield strong global sensitivities which cannot be identified via a local analysis. This is further justified via the Pearson bearing analysis.

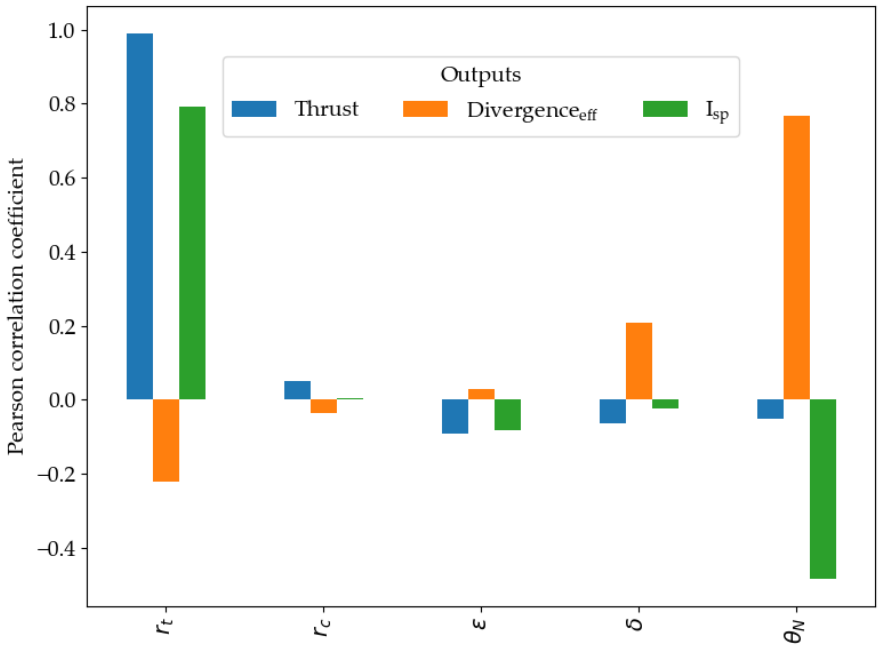


FIGURE 4.2: Pearson analysis on the input parameters bearings on the output parameters.

Figure 4.2 illustrates the results of the Pearson bearing analysis. Positive values indicate that as the input parameter in question increases, so does the output parameter, while negative coefficients indicate an inverse relationship. It should be noted that the values with strongest correlations are closely tied to those predicted by the random forest method. As expected, $\delta$ and $\theta_N$ values had the greatest effect on the divergence efficiency of the nozzle, while $R_t$ had an inverse relationship. The analysis suggests that the throat region is the most important for developing the thrust of a nozzle. However, the specific impulse is a better indicator of the ability of a nozzle to convert the mass-flow into thrust, and is best developed by the $\theta_N$, $\delta$, and $R_t$ values.

### 4.1.2 LHS Study (500 Case)

494 cases were generated using a LHS of dimension $M = 494$. This number was chosen such that an estimated 10% of cases could be cut due to $\theta_E$ values being too small (which yielded poor convergence within CFD simulations). Another 10% of cases may be non-converging during the batch simulation process, leaving $\sim$ 400 cases with adequate convergence and reliable results/data. In total, 432 cases were used for the sensitivity analysis and further for the parametric analysis and surrogate based MDO.

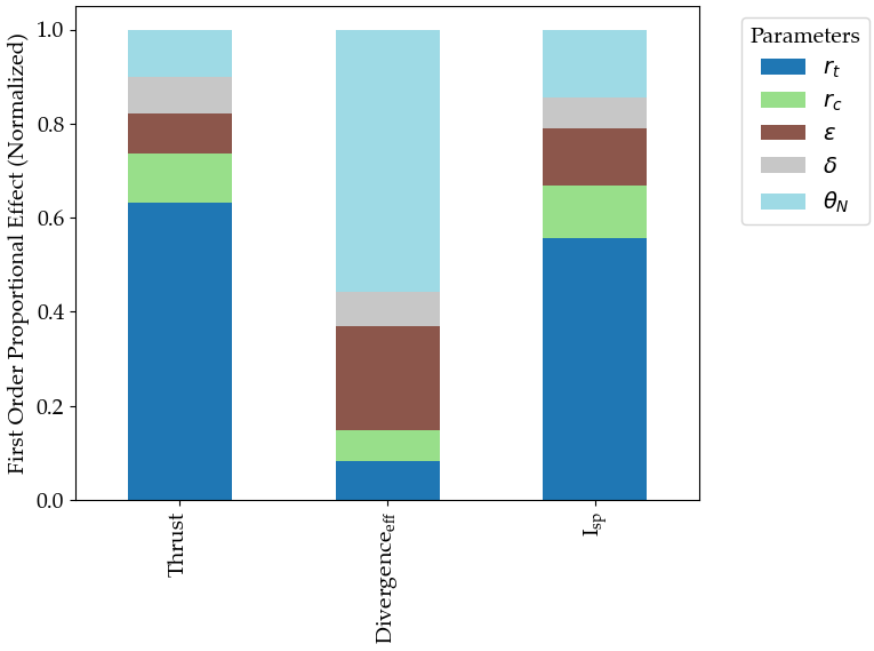


FIGURE 4.3: Delta sensitivity analysis first order indices conducted on the input parameters with respect to the output parameters.

Figure 4.3 shows the first order indices for the thrust, divergence efficiency, and specific impulse of the nozzles broken down into the individual proportional effect of each geometric parameter. Table 4.1 further breaks down these indices, including

the models confidence value associated with each index. This confidence value is proportional to the index it is associated with; if the confidence value is too large with respect to the index, the confidence is, therefore, low. It is evident that the confidence is almost universally low, except in the case of the effect of $R_t$ on the thrust and specific impulse of a system, and $\theta_N$ on the divergence efficiency and specific impulse.

TABLE 4.1: Delta sensitivity analysis first order indices for each parameter, with respective confidence bounds.

| **Output** | **Parameter** | $S_i$ | **Confidence** |
|---|---|---|---|
| Thrust | $R_t$ | 0.709286 | 0.008319 |
| | $\varnothing_c$ | 0.069979 | 0.025085 |
| | $\epsilon$ | 0.070659 | 0.022645 |
| | $\delta$ | 0.076502 | 0.020176 |
| | $\theta_N$ | 0.057644 | 0.021182 |
| Divergence efficiency | $R_t$ | 0.077200 | 0.020989 |
| | $\varnothing_c$ | 0.087264 | 0.023706 |
| | $\epsilon$ | 0.101654 | 0.032558 |
| | $\delta$ | 0.078644 | 0.019131 |
| | $\theta_N$ | 0.535009 | 0.024437 |
| $I_{sp}$ | $R_t$ | 0.501112 | 0.026917 |
| | $\varnothing_c$ | 0.091105 | 0.024481 |
| | $\epsilon$ | 0.136545 | 0.027120 |
| | $\delta$ | 0.082273 | 0.023260 |
| | $\theta_N$ | 0.149531 | 0.026649 |

While the confidence was low, the results were consistent when analysed multiple times. To understand why it was self-consistent, a Pearson analysis was also conducted to observe the local correlation. The Pearson analysis may be used to understand how a variable affects the outcome locally and whether it can be used as an indicator of what the outcome may be.

Figure 4.4 illustrates the Pearson correlation coefficients. These coefficients show remarkable similarity to those evaluated in Figure 4.2. Whilst assuring the validity of the static studies results, this simultaneously illustrates why the delta first-order indices were consistent; high local correlation ensures that despite low confidence, results of the analysis remain largely consistent at least in terms of their respective proportion to each other. Figure 4.4 is different from Figure 4.2 in only two significant ways: $\epsilon$ has a much higher effect on the divergence efficiency of a system than previously indicated, while $\delta$ has a much smaller effect.

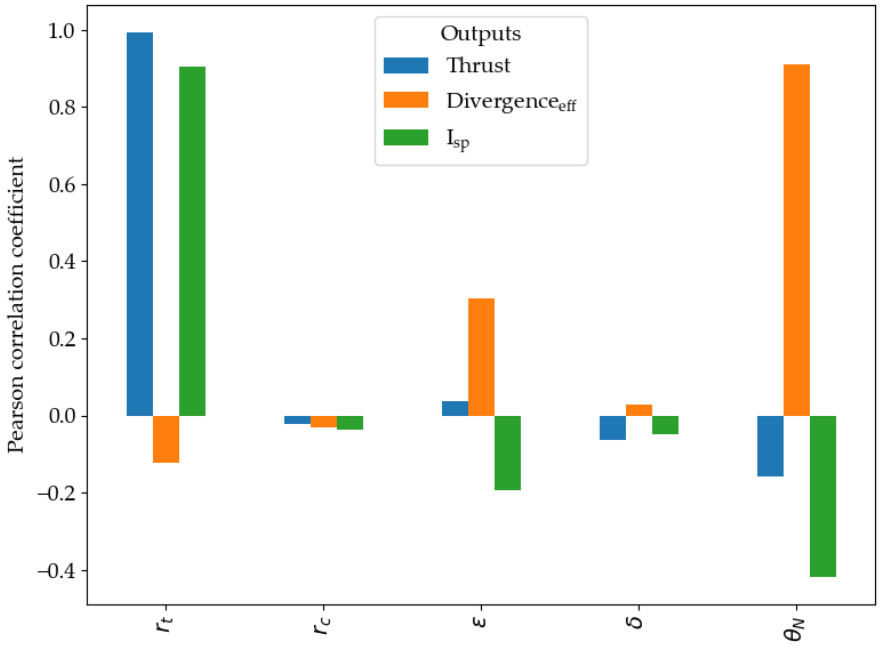


FIGURE 4.4: Pearson analysis on the input parameters bearings on the output parameters.

### 4.1.3 Sensitivity Analysis using the Surrogate Model

A surrogate model was trained on each output parameter, using 90% of the training data, whilst the remaining 10% was used as validation of the models. 5 different surrogate models were trained for each output parameter, but only the best performing was used. The model that had the best performance for thrust was found to be the orthogonal response surface methodology[30], divergence efficiency was the multi-layer perception method[31], and specific impulse was found to be the orthogonal response surface methodology. These had errors of 0.0005%, 0.0450%, and 0.0203% respectively on the validation data.

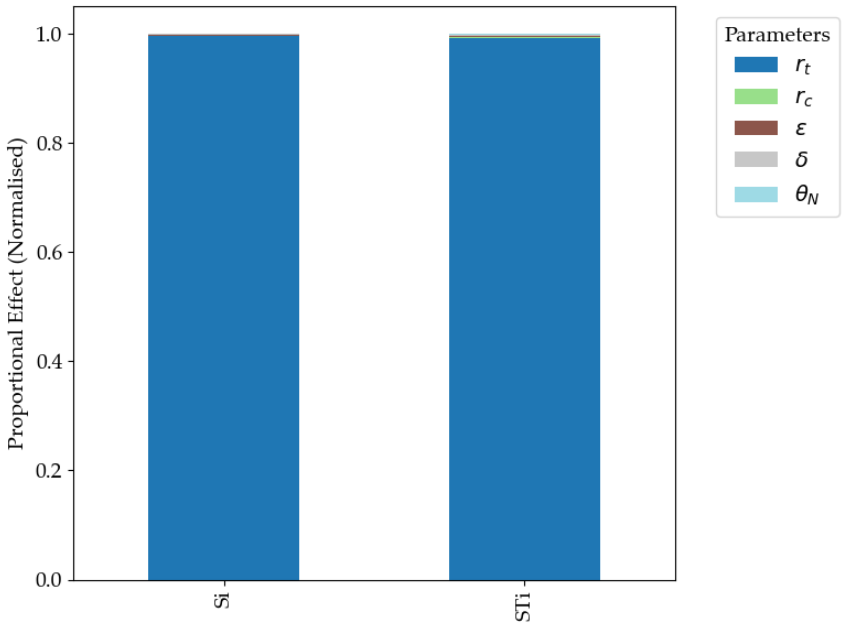


FIGURE 4.5: Sensitivity indices of the thrust with respect to the input parameters using the surrogate model.

The surrogate models could then be used to produce a set of data that conforms to the required Sobol analysis structure. Figure 4.5 illustrates the Sobol sensitivity indices for each geometric input parameter on the thrust generated. These indices are normalised against each other to illustrate and compare their proportional effect. The thrust is completely dominated by the $R_t$ parameter, both locally and globally. It is generally expected that the expansion ratio would have a large influence on the thrust, but this is not the case.

$$F \propto \pi R_t^2 (1 + \epsilon) \tag{4.1}$$

This discrepancy was unexpected, and investigated further. From eq. (3.7), a direct correlation to the throat area, $A^*$, is found, which is itself exponentially proportional to the throat radius. Whilst eq. (3.6) illustrates that the thrust is both proportional to the mass flow rate and the exit area (which is tied to $\epsilon$), the effect of $R_t$ is stronger. Using eqs. (3.6) and (3.7), eq. (4.1) may be derived, which illustrates the proportional effect both the throat radius and the expansion ratio have on the thrust of a system. As such, the throat radius was disproportionally represented in the sensitivity analysis. The other geometric parameters were completely dominated and thus insufficient analysis could be performed on their sensitivity indices as they were all equally small.

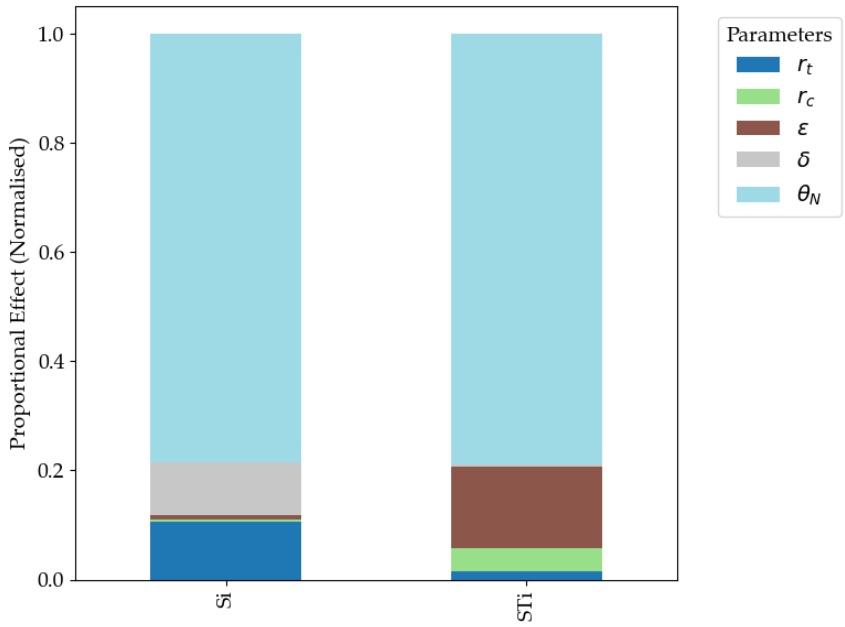


FIGURE 4.6: Sensitivity indices of the divergence efficiency with respect to the input parameters using the surrogate model.

The divergence efficiency was more interesting to examine, as Figure 4.6 illustrates how the geometric parameters combine to influence the divergence efficiency result. As expected the value of $\theta_N$, the initial divergent angle, had a large bearing on the divergent efficiency of the nozzle. The curve factor, $\delta$, also influences the nozzles divergent efficiency capacity, although only locally. The $S_{Ti}$ index had very little representation of the $\delta$ factor, indicating that the combined effect was insignificant when compared to the combined global effects of the other geometric parameters.

The $\delta$ factor was expected to influence the divergence efficiency, since it plays a role in the shaping of the divergent portion and directly relates to the near-wall effects. The indication of the limited effect $\delta$ has that Figure 4.6 highlights is indicative of the changed fluid dynamics at the millimetre scale. Viscous wall-effects and the boundary layer produced at this scale is significant and is the leading cause of this discrepancy. Further, the $\delta$ value was severely limited due to simulation convergence issues. Alongside a limited $\theta_N$ value, the range of nozzle shapes that were applicable and available for analysis in this study was reduced and may provide further insight into the small effect the $\delta$ value had on the divergence efficiency.

Locally, $\epsilon$ had very little influence on the divergent efficiency of the system, but had a large effect on the global index. This is perhaps indicative of the construction of the nozzle, through which larger values of $\epsilon$ resulted in longer nozzles. This suggests that the nozzle length had a greater impact on the divergent efficiency, and thus the flow needed time to develop within the nozzle to feel the impacts of the nozzles geometric design on divergent efficiency.

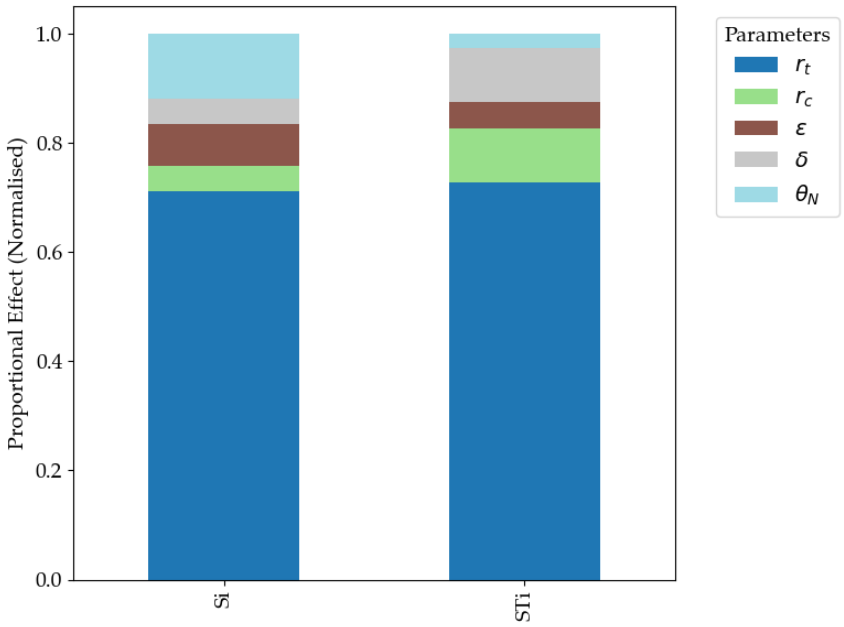


FIGURE 4.7: Sensitivity indices of the specific impulse with respect to the input parameters using the surrogate model.

Figure 4.7 illustrates the specific impulse Sobol indices for each geometric parameter. The specific impulse is also dominated by the throat radius, $R_t$. However, the other geometric parameters also play a fair part in developing the specific impulse of the system, both locally and globally. The value of $\theta_N$ has a greater influence locally, while a much smaller effect globally when combined with all other geometric parameters. Inversely, $\delta$ and the smoothing radius ($r_c$) had a greater effect globally than locally. This indicates that the specific impulse is sensitive to the shape and flow of the nozzle, rather than the scale or broad shape of it, as shown by the greater effect of the smoothing radius and $\delta$ indices over the $\theta_N$ and $\epsilon$ indices.

## 4.2 Parametric Analysis of the Parabolic Micro-Nozzle

The parabolic nozzle geometry was analysed using three distinct methods, each offering unique insights into the subtle variations of nozzle output from small changes to nozzle geometry. Figure 4.2 illustrates the results from a targeted local analysis, while Figure 4.3 illustrates to what proportion each input parameter has on the outputs locally. Figures 4.5, 4.6, and 4.7 offer insight into the local and global effect, and may be used as validation of the surrogate models when comparing the local sensitivity to Figures 4.2 and 4.4.

It was found that $\epsilon$ had more influence when the dataset was expanded from the static to the LHS study with respect to the divergent efficiency, which suggests that the value of $\epsilon$ may have varying effect on different geometries and thus must have a unique global effect. Figure 4.6 confirms this theory, where the local influence

is small, but the global more dominant. The inverse is true of $\delta$, where its local influence is greater than its global influence.

#### The throat radius, $R_t$

It was found that by varying the throat radius, $R_t$, the thrust was greatly increased. This is tied to the construction of the geometry; the exit radius was tied directly to both the expansion ratio, $\epsilon$, and the throat radius, $R_t$. In doing so, the thrust was dominated by the throat radius parameter. Further, the throat radius proportionally increased the mass flow rate. Thus, the radius of the throat had a direct and exponential correlation to the nozzle thrust generated, as shown using eqs. (3.6), (3.7) and (4.1).

#### The smoothing radius, $r_c$

Almost universally, all sensitivity analyses found that the throat smoothing radius, $r_c$ had very little effect on any measured output. Table 4.1 illustrates that while the delta analysis showed a non-insignificant effect on thrust, the confidence was low. The throat smoothing radius does have some effect on the divergence efficiency and specific impulse of a system when investigated globally, as shown in Figures 4.6 and 4.7, but is generally overshadowed by other geometric parameters. While the throat region does play a significant role in the generation and development of exhaust through a nozzle, the smoothing between the throat region and the parabolic region is not significant at the relative scales tested in this study.

#### The expansion ratio, $\epsilon$

The expansion ratio's biggest contribution to the nozzle outputs was in the divergent efficiency, where it can be concluded that the global effect of the expansion ratio - preserving nozzle shape while extending nozzle length and exit area - helped mature the exhaust flow and yield more direct exhaust flow. Although there was a local inverse effect on the specific impulse, as illustrated by Figures 4.2 and 4.4, the surrogate analysis found the overall impact to be diminishing in comparison to the other geometric values. Previous research on nozzle dynamics, both at the metre and millimetre scale[24], highlight a distinct relationship between the expansion ratio and thrust, which is broadly absent. This can be attributed to the overwhelming influence of the throat radius on the thrust output of nozzle geometries tested. Alongside the low confidence reported by Table 4.1, Figure 4.2 suggests a negative correlation, further establishing the grounds for dismissing the marginal results of the small-population studies.

**The curve factor, $\delta$**

As shown in Figure 4.7, the $\delta$ value had its most significant contribution to the output parameters in the global effect on specific impulse, indicative of a sensitivity to the shape of the nozzle on the ability to convert exhaust flow to thrust. This highlights how even at the millimetre scale, even small changes to the broad shape of a nozzle may change the flow dynamics. This underlines how the boundary layer effects may be influenced by minimal changes in nozzle shape and are not an overpowering force at this scale, merely a consideration that may be leveraged in favour of specific nozzle performance characteristics.

**The initial divergent angle, $\theta_N$**

The divergence efficiency was highly correlated to the value of $\theta_N$, both locally and globally as shown in Figure 4.6. When analysing the global effect, $r_c$, $\epsilon$, and $\theta_N$ are all contributing factors which influence both the nozzle shape, as well as the initial expansion region immediately exiting the throat of the nozzle. This highlights how the divergence efficiency is strongly influenced by the initial stages of exhaust expansion in the nozzle at the millimetre scale. This finding is consistent with previous work[24, 27] and suggests a relationship that transcends the nozzle shape (*i.e.* conical, parabolic *etc.*).

The sensitivity analysis indicates that in order to better understand the behaviour and influence of the geometric parameters which shape the divergent parabolic region of the nozzle, more constraints need to be applied. The throat radius should remain fixed, to ensure consistent exhaust inlet behaviour; to understand the complex behaviour surrounding the mass flow rate at the millimetre scale, the throat radius should remain constant to instead focus on the divergent region while mass flow rate is kept constant. Further, the smoothing radius is largely underrepresented in its global and local effect on measured exhaust outputs; and thus could be kept constant to further reduce research complexity with little consequence provided a suitable optimal value is agreed upon.

# Chapter 5

# Multi-Objective Design Optimisation

After completing the parametric study, the surrogate models can be used for the MDO of the nozzle geometry. The raw input data (*i.e.* the 432 cases simulated) achieved ranges of the objective functions as shown in Table 5.1; while some nozzles achieved very high values for one or two of the objective functions, the MDO will maximise all objectives simultaneously.

TABLE 5.1: Ranges of the objective functions from the simulated, LHS dataset (before MDO).

| Output | Minimum Value | Maximum Value | Unit |
|---|---|---|---|
| Thrust | 0.059 | 0.360 | N |
| Divergence Efficiency | 99.14 | 99.74 | % |
| Specific Impulse | 106.2 | 110.0 | s |

A brief investigation into a SAEA MDO process was conducted using Python and is presented in Appendix C. However, due to time constraints, the EA model and MDO process could not be refined to produce results of sufficient calibre. As such, they are excluded from the main body of work.

## 5.1 MDO Results

The MDO process requires all objective functions to be minimisation evaluations, where the objectives being optimised in this study are maximisation functions. The conversion is simple, where the functions are given a negative value when evaluated and are thus converted from a maximisation to a minimisation. When interpreting the raw results from the MDO process of the nozzle geometry, it is useful to remember that due to the minimisation conversion, axes may be inverted.

The MDO was run using a MATLAB library composed by Kyushu University. The SAEA was initialised using a standard setting which has been used in previous micro-nozzle optimisation work[24, 27]. The EA operation probabilities were as follows: the crossover probability of individuals across generations was set to 100%,

while the mutation probability of individuals was set to 10%. 200 individuals were used and evaluated over 200 generations, with 40,000 total evaluations completed via surrogate models.

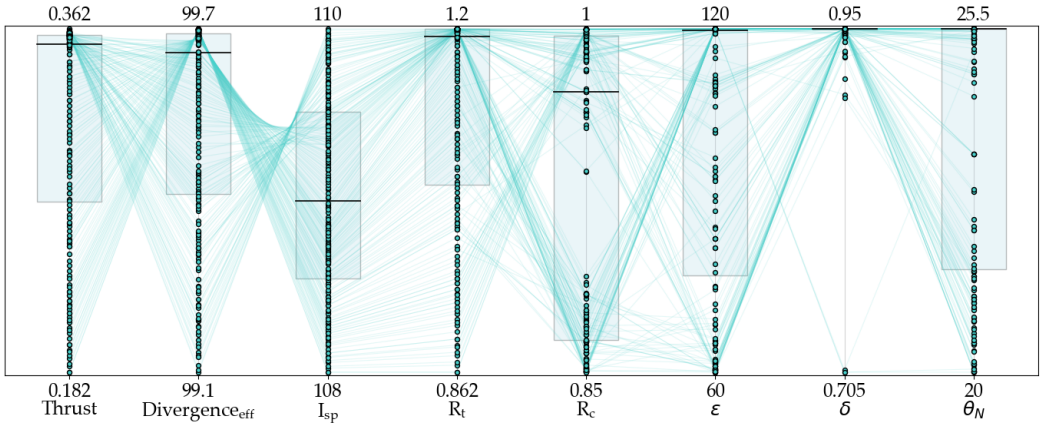


FIGURE 5.1: MDO parallel coordinate plot for non-dominated solutions, linking the objective functions to the input parameters which produce them.

Noticeably, most input parameters exhibit a wide range of values; having a range of values provides confidence that the surrogate model, alongside the design and definition of the nozzle, provides enough complexity and range of nozzle designs which may produce sufficiently different nozzle characteristics. In optimising the objective functions, some trends have arisen in the input geometric parameters.

$R_t$ tends towards larger values, through which the sensitivity analysis illustrated plays an important role in increasing thrust and specific impulse. The range illustrated can be justified by the divergence efficiency competing with the increase in thrust and specific impulse, as Figure 4.4 illustrates. $R_c$ exhibits a range of values which either completely maximise or minimise its value, suggesting that the bounds for $R_c$ may be expanded to further extract performance in the specific impulse or divergence efficiency outputs (given these are both mildly affected by the value of $R_c$ as found in Figures 4.7 and 4.6 during the sensitivity analysis). $\epsilon$ exhibits a wide range of values covering the entire spectrum of values, illustrating very good diversity among the Pareto optimal designs. However, the majority of them sit at the maximum value of the available range, which indicates that the range could be expanded further to extract greater performance metrics from a larger value of $\epsilon$. The $\delta$ value is perhaps that most universally similar value, where many nozzles maximise the value to improve the divergence efficiency. Some designs have minimised the value, producing a more conical design, but these are few when compared to the amount of more traditionally parabolic designs produced via the MDO. Finally, $\theta_N$ exhibits a range of values, with many maximising the angle to further maximise the divergence efficiency.

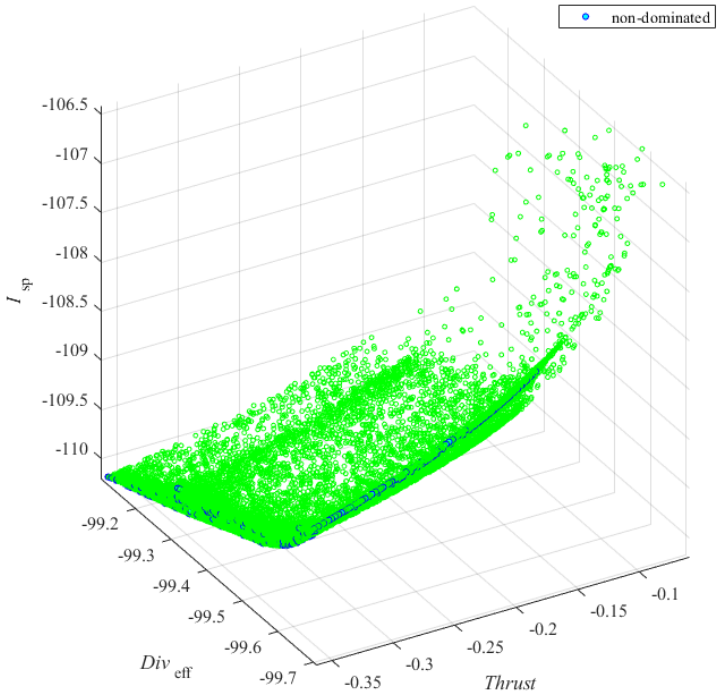


FIGURE 5.2: The final MDO generation plotted against the objective functions. Cyan dots represent non-dominated solutions while green dots represent feasible, albeit dominated, solutions.

The objective functions as displayed in Figure 5.2 suffer from the axes inversion as mentioned previously. The final generation of the MDO process using surrogate models is illustrated in Figure 5.2. The non-dominated solutions are coloured in cyan; non-dominated solutions satisfy the vector condition $f_i(\vec{x}) < f_i(\vec{x}')\quad i \subset (1,2,3)$, where $(1,2,3)$ indicate one of the objective functions.

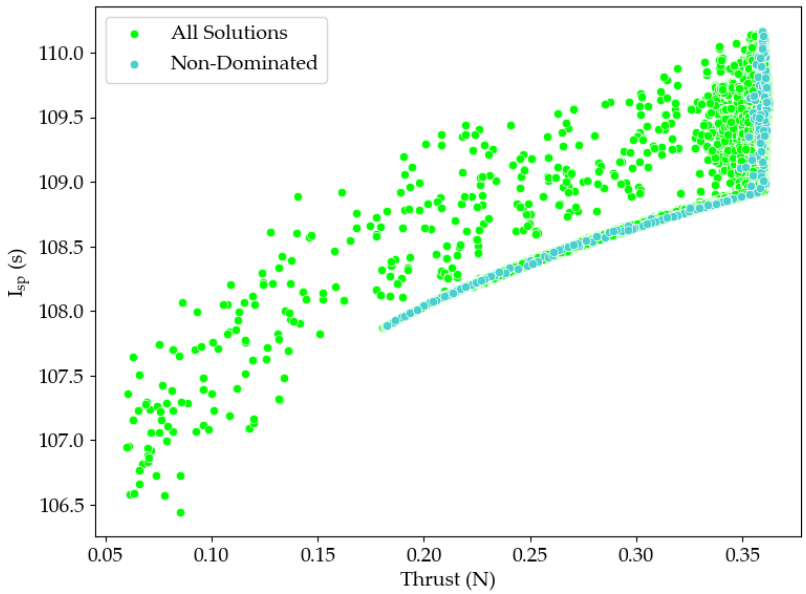


FIGURE 5.3: Final generation of the MDO, plotted against thrust and specific impulse.

A clear dependence between the thrust and specific impulse of the system can be identified in Figure 5.3. The is a direct correlation, where an increase in thrust also increases the specific impulse of the system. The upper right region is indicative of some restriction on thrust, where the specific impulse may be varied without impacting thrust.

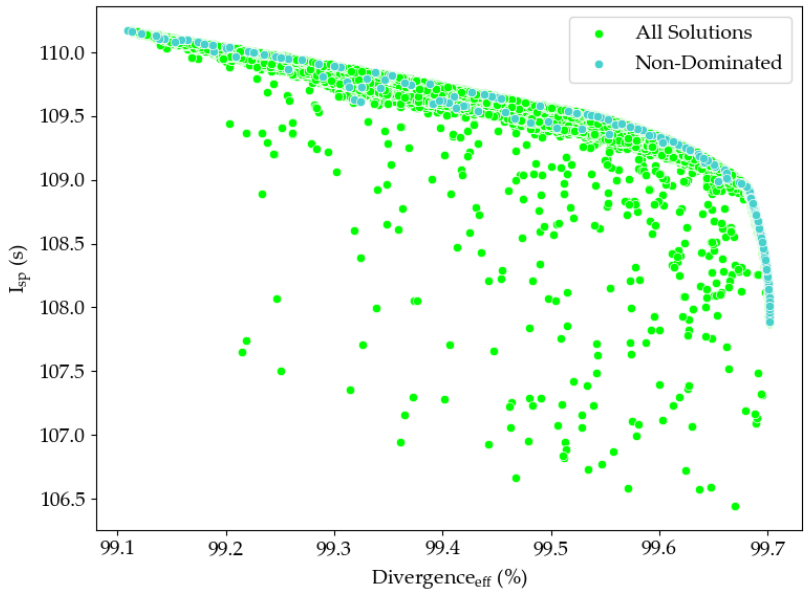


FIGURE 5.4: Final generation of the MDO, plotted against specific impulse and divergent efficiency.

A trade-off relationship between the specific impulse and the divergence efficiency can be seen in Figure 5.4, where an increase in specific impulse results in a decrease in the divergence efficiency of the system.

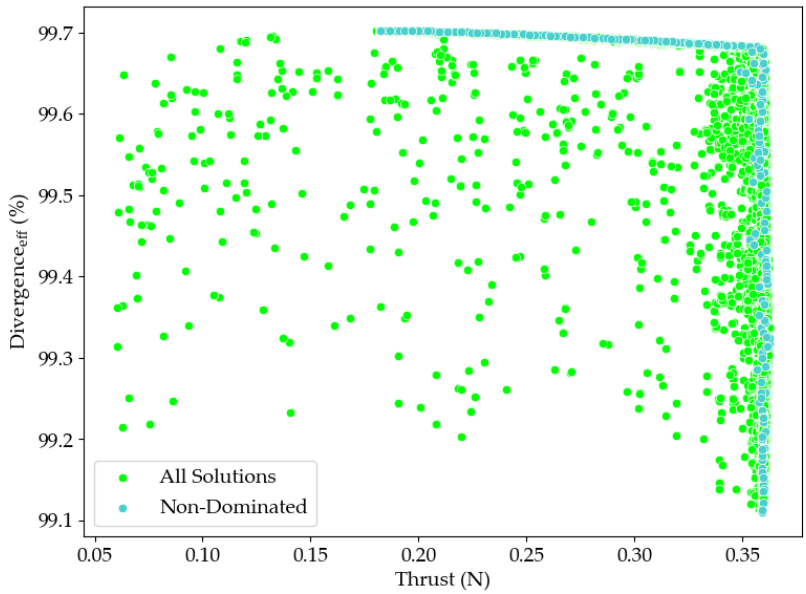


FIGURE 5.5: Final generation of the MDO, plotted against divergent efficiency and thrust.

Finally, the general relationship between the divergence efficiency and the thrust of a system is unstructured, but a Pareto front has developed at high divergent efficiencies. Above 99.68%, a strong trade-off relationship develops where a small increase in the divergence efficiency dramatically reduces the thrust of a system. Thus, geometries which produce divergent efficiencies above 99.68% may generally be excluded from consideration when choosing a nozzle geometry for a system, given the large reduction in thrust for a small increase in efficiency. Compared to conical micro nozzles[24, 27], the divergence efficiency of the optimised parabolic nozzle geometry is much improved across the board.

## 5.2 Verification of Results

In order to examine the Pareto front generated by the MDO process, a small selection of cases may be extracted and examined further. These validation cases were used to ensure that the MDO and surrogate models have yielded high quality results that are representative of the nozzle geometry and performance characteristics. Three cases were examined, one for each performance metric, where the highest performing case was chosen. Thus, a high thrust case, a high divergence efficiency case, and a high specific impulse case have been chosen for validation.

### 5.2.1 High Thrust Case

TABLE 5.2: High thrust geometric parameters derived through MDO

| Decision Variable | Value | Unit |
|---|---|---|
| $R_t$ | 1.200 | mm |
| $\epsilon$ | 120.0 | - |
| $\theta_N$ | 20.00 | $^\circ$ |
| $\delta$ | 0.705 | - |
| $\varnothing_c$ | 1.000 | mm |

The highest thrust developed during the MDO was 0.362 N, which is a 5.6 % increase over the unprocessed data as shown in Table 5.1. Table 5.2 outlines the geometric inputs which produced this nozzle. $R_t$, $\epsilon$, and $\varnothing_c$ have all been maximised, while $\theta_N$ and $\delta$ have been minimised. Notably, $\delta$ has not been pushed to its absolute limit, which suggests that the produced value is at some local minimum; this is in contrast to the other parameters, where the values at the bounds suggest minimums outside the bounded ranges. Increasing or decreasing these values may allow for further performance increase.

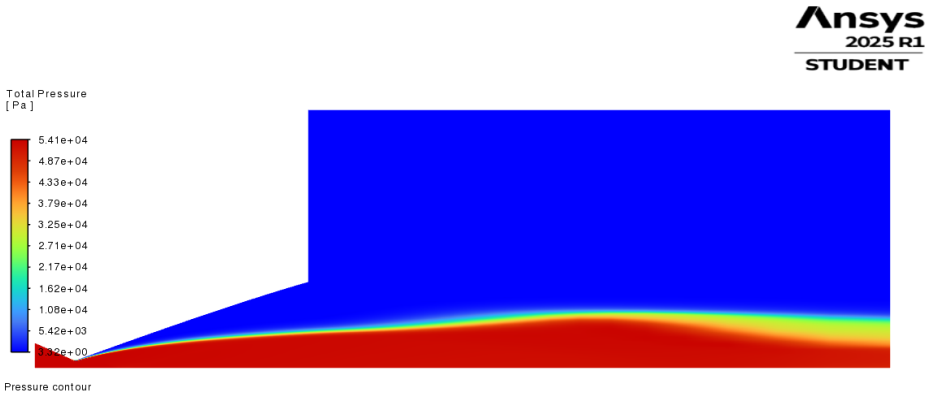


FIGURE 5.6: CFD pressure contour of the high thrust case.

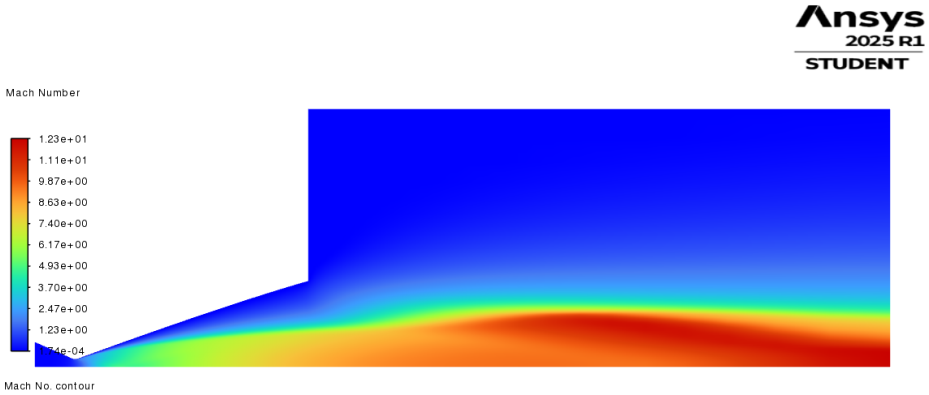


FIGURE 5.7: CFD Mach number contour of the high thrust case.

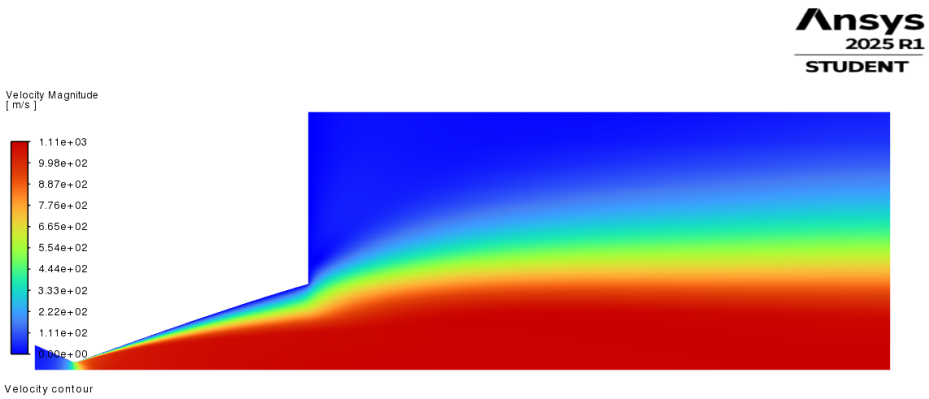


FIGURE 5.8: CFD velocity contour of the high thrust case.

TABLE 5.3: High thrust output parameters derived through MDO

| Output | Estimated Value | Simulated Value | Unit | Difference (%) |
|---|---|---|---|---|
| Thrust | 0.362 | 0.362 | N | -0.102 |
| Divergence Efficiency | 99.32 | 99.32 | % | 5.9e-4 |
| Specific Impulse | 109.6 | 109.7 | s | 0.048 |

The high thrust nozzle is very conical in profile, but has the characteristic higher divergence efficiency provided by the minimal parabolic profile. While the divergence efficiency is low with respect to the parabolic nozzles developed in the MDO, they exceed those produced using conical designs. Conical nozzles fail to deliver high divergence efficiency when also producing high thrust; the divergence efficiency fails to achieve more than 97 % when producing high thrust[27].

The difference between the MDO estimated values and the CFD simulated values shows very good agreement, as shown in Table 5.3. This can be attributed to the high confidence in surrogate models. Further, the flow fields produced by the CFD are of excellent quality, as illustrated in Figures 5.6, 5.7, and 5.8. All flow fields are continuous and do not display any flow splitting.

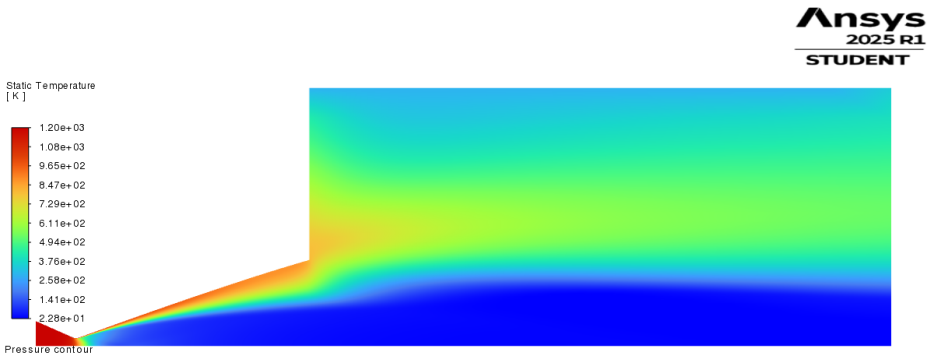


FIGURE 5.9: CFD temperature contour of the high thrust case.

The Mach number and the velocity contours are typically closely correlated, given the direct relation between the two values. However, Figures 5.7 and 5.8 have discrepancies which may only be explained when examining the temperature within the nozzle. Figure 5.9 illustrates the static temperature within the high thrust case nozzle, but may be representative of all cases. The high temperature at the nozzle wall is resultant from the large boundary layer effect at the micro scale, and the wall friction. This change in temperature changes the speed of sound and thus the local Mach number. Therefore, the boundary layer is both confirmed and its effect on exhaust flow quantified.

### 5.2.2 High Divergence Efficiency Case

TABLE 5.4: High divergence efficiency geometric parameters derived through MDO

| Decision Variable | Value | Unit |
|---|---|---|
| $R_t$ | 0.862 | mm |
| $\epsilon$ | 120.0 | - |
| $\theta_N$ | 25.50 | ° |
| $\delta$ | 0.950 | - |
| $\varnothing_c$ | 1.000 | mm |

The highest divergence efficiency developed during the MDO was 99.70 %, which is a 0.04 % decrease over the unprocessed data as shown in Table 5.1. Figures 5.4 and 5.5 suggest that the divergence efficiency is generally a conflicting parameter to both thrust and specific impulse. Thus, it can be seen that the MDO has found a maximum for the divergence efficiency while maintaining the conflicting thrust and specific impulse values.

Table 5.2 outlines the geometric inputs which produced this nozzle. The highest divergence efficiency produced via the MDO has maximum values for $\epsilon$, $\theta_N$, $\delta$, and $\varnothing_c$, while had a mid-range value for $R_t$. This suggests that the value found for $R_t$ is a local minimum, where the other parameters may benefit from further extended bounds.

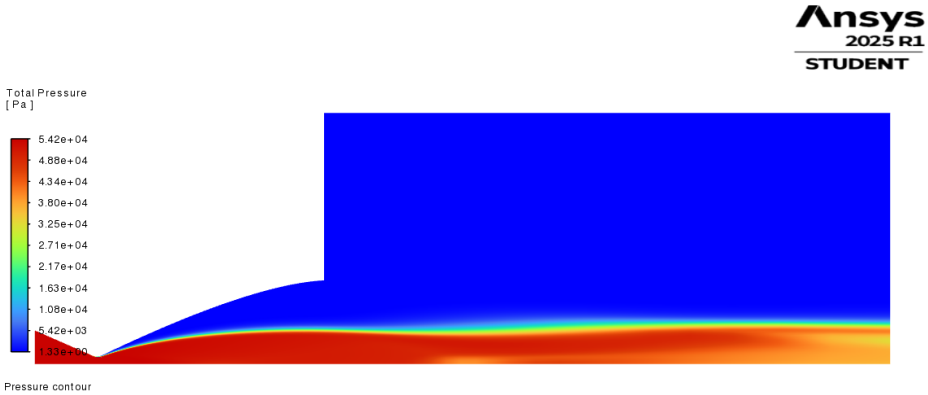


FIGURE 5.10: CFD pressure contour of the high divergence efficiency case.

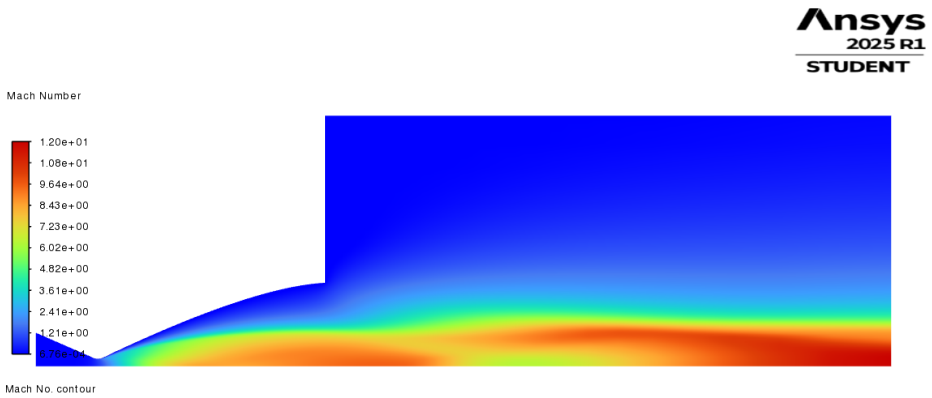


FIGURE 5.11: CFD Mach number contour of the high divergence efficiency case.

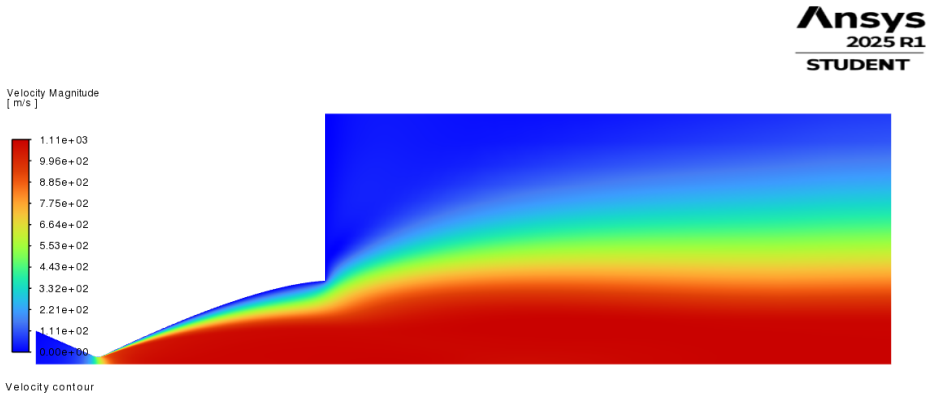


FIGURE 5.12: CFD velocity contour of the high divergence efficiency case.

TABLE 5.5: High divergence efficiency output parameters derived through MDO

| Output | Estimated Value | Simulated Value | Unit | Difference (%) |
|---|---|---|---|---|
| Thrust | 0.182 | 0.181 | N | -0.923 |
| Divergence Efficiency | 99.70 | 99.68 | % | -0.022 |
| Specific Impulse | 107.9 | 108.8 | s | 0.836 |

The high divergence efficiency nozzle has a much more pronounced curve when compared to the high thrust case; this assists in directing flow out of the nozzle in the axial direction. This example highlights how the parabolic shape improves performance at the millimetre scale, similar to the usual metre scale, despite the increased boundary layer effects.

Some minor flow splitting can be identified in the downstream region of Figure 5.10 along the axis. This does bring some level of uncertainty in the results and is evident in the larger differences between the estimated values and simulated values as shown in Table 5.5 when compared to the high thrust case (see Table 5.5).

### 5.2.3 High Specific Impulse Case

TABLE 5.6: High specific impulse geometric parameters derived through MDO

| Decision Variable | Value | Unit |
|---|---|---|
| $R_t$ | 1.200 | mm |
| $\epsilon$ | 60.00 | - |
| $\theta_N$ | 20.01 | ° |
| $\delta$ | 0.950 | - |
| $\varnothing_c$ | 0.851 | mm |

The highest specific impulse developed during the MDO was 110.2 s, which is a 0.18 % increase over the unprocessed data as shown in Table 5.1. Table 5.6 outlines the geometric inputs which produced this nozzle. The geometric values of $R_t$ and $\delta$ have been maximised, while $\epsilon$ has been minimised. The value of $\theta_N$ was very close to its lower bound, but may have found a minimum just above this bound. It is also possible that this near-boundary value is a characteristic trait of the EA process, where it has been mutated in the final generation. Interestingly, $\varnothing_c$ found a local minimum at 0.851 mm, where the high thrust and specific impulse nozzles both had this value maximised.

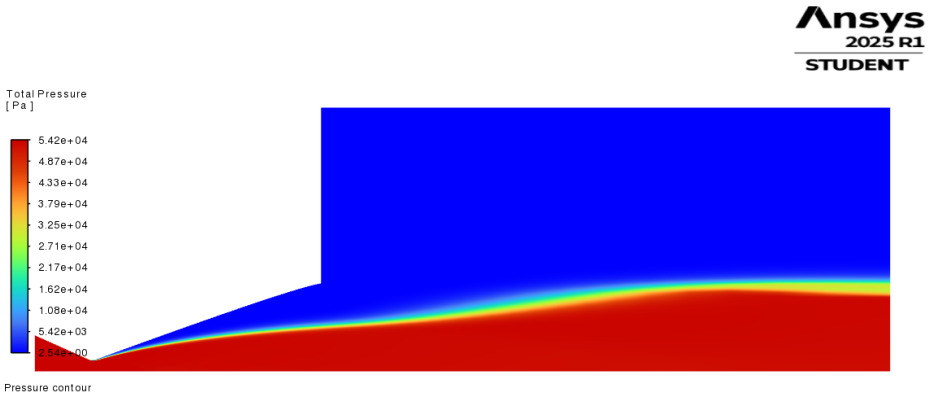


FIGURE 5.13: CFD pressure contour of the high specific impulse case.

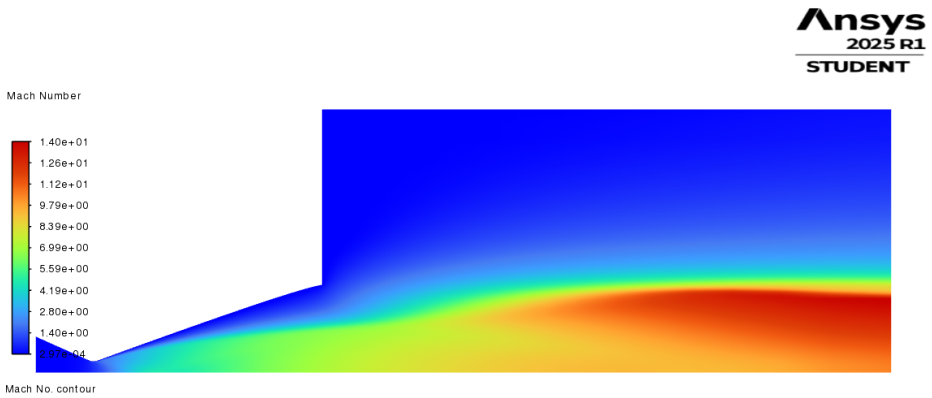


FIGURE 5.14: CFD Mach number contour of the high specific impulse case.

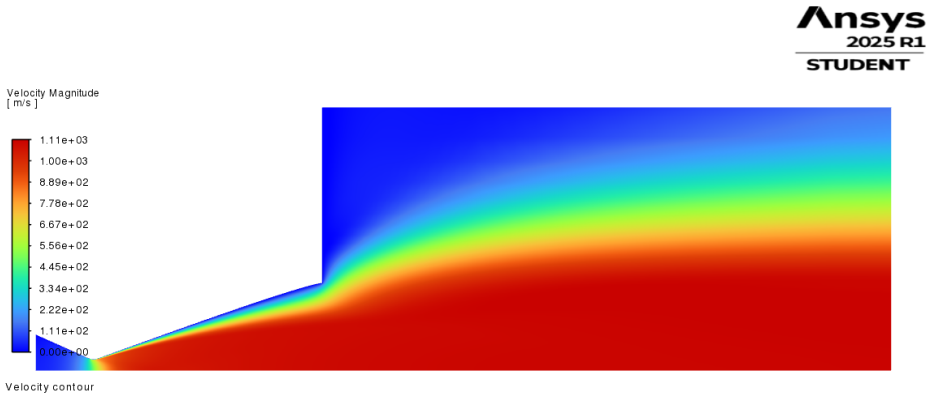


FIGURE 5.15: CFD velocity contour of the high specific impulse case.

TABLE 5.7: High specific impulse output parameters derived through MDO

| Output | Estimated Value | Simulated Value | Unit | Difference (%) |
|---|---|---|---|---|
| Thrust | 0.359 | 0.359 | N | -0.121 |
| Divergence Efficiency | 99.11 | 99.08 | % | -0.026 |
| Specific Impulse | 110.2 | 110.3 | s | 0.126 |

The high specific impulse nozzle had more distinct curvature than the high thrust case, due to the increased $\delta$ value. The nozzle achieved both very high specific impulse value and thrust value, while displaying the worst divergence efficiency of the three verification cases. Similarly to the high thrust nozzle, the parabolic nature of the nozzle retains the overall high divergence efficiency when compare to that of purely conical nozzles.

The difference between the estimated MDO values and simulated CFD values was insignificant, as shown in Table 5.7, with the estimation being under-representative of the actual simulated value. This can be attributed to the high confidence in surrogate models. The flow fields, as shown in Figures 5.13, 5.14, and 5.15, are of excellent quality as they are all continuous and no flow splitting is present.

The MDO process, using EA, successfully maximised the performance characteristics of the nozzles, where local minimums were found for various geometric attributes which defined some of the highest performing nozzles. Each high performing nozzle found one geometric parameter that helped achieved the unique attributes of that design, namely $\delta$, $R_t$, and $\varnothing_c$, for the highest thrust, divergence efficiency, and specific impulse respectively. Interestingly, the derived values of $\varnothing_c$ are also higher than Rao's formula suggested value, where $\varnothing_c = 0.382\, R_t$.

Overall, all three cases showed very good correlation with the estimations produced via the surrogate models and the MDO process. This gives high confidence in the set of solutions contained along the Pareto front, from which any configuration may be chosen to provide the desired performance characteristics. The slight flow splitting observed in the high divergence efficiency case (see Figure 5.10) is problematic, but may be resolved with further simulation work. Whilst both the high thrust and divergence efficiency cases both saw losses from their estimated performance metric, the specific impulse saw an increase; this can be attributed to the best simulation convergence, where the specific impulse case saw the lowest mass flow rate difference between the inlet and interface of $4 \times 10^{-4}$ %.

# Chapter 6

# Conclusion

This study produced a parabolic nozzle contour and found a set of Pareto-optimum profiles at the millimetre scale. The resultant geometries were compared and the key geometric characteristics determined by the scale-dependent physics for the nozzle examined. Investigations into the nozzle geometry prior to the parametric study and MDO process revealed parametric dependence on CFD simulation convergence. Specifically, values of $\theta_N$ over 25.5° impeded simulation convergence, where flow splitting occurred. Further sensitivity analysis was performed to characterise geometric parameters with performance outputs.

The nozzle geometry had characteristic geometric parameters which drove each of the output performance metrics. The throat radius, $R_t$, had an effective 100 % correlation to the thrust produced, as a consequence of the exponential correlation derived. Similar to the conical nozzle, the initial divergent angle, $\theta_N$, had the largest share of the global correlation to the divergence efficiency of the nozzle at 88.1 %. The $\delta$ factor, which controls the curve of the parabolic region, was shown to correlate to the divergence efficiency only locally, while having more of an effect on the specific impulse of the nozzle. This behaviour may be indicative of a relationship to the boundary layer effects.

The surrogate models performed very well, where they allowed for in depth analysis of parameter sensitivities and computationally inexpensive EA processes using SAEA. Verification of resultant geometries was performed, where the biggest difference between surrogate estimation and simulation output in resultant performance was 0.923 %. Through the MDO process, both thrust and specific impulse saw increased performance when compared to the best performing nozzle in each characteristic during the LHS raw data.

The parabolic nozzle geometry performed well with respect to the output parameters, especially the divergence efficiency. After the MDO process, the lowest divergence efficiency recorded was 99.10 %, which is only a 0.7 % decrease over a similar, best performing conical nozzle geometry[27]. The MDO Pareto front was well defined, except in the high thrust region, where it can be extrapolated that further performance may be gained should the parameter bounds be expanded. While the maximisation of most parameters during the MDO process is indicative of further performance gains outside of the parameter bounds, the maximisation of $\varnothing_c$

is indicative of key fluid dynamic behavioural differences between the millimetre scale and metre scale. The change in behaviour due to boundary layer effects is the increase in the throat smoothing radius, $\varnothing_c$.

Novel findings produced by this study are as follows:

- Micro parabolic nozzles achieved much higher divergence efficiencies than similar conical nozzles, providing evidence that metre scale research may still be relevant despite the significant boundary layer effects.
- Parabolic nozzle expansion ratios ($60 \leq \epsilon \leq 120$) have little effect on thrust production at the micro scale, but nozzle length is an important quantity with respect to exhaust maturity and divergence efficiency.
- Micro parabolic nozzle geometries differ from their metre scale counterparts, where historical geometric parameters, like $R_c$ (the smoothing radius), are significantly larger.

## 6.1 Future work and recommendations

Further work on expanded parameter bounds while retaining simulation convergence and quality can be undertaken, which would further expand and understand the Pareto front, especially in the thrust aspect. Further, optimal values may be found by expanding these bounds in the form of local minimums over more geometric parameter values for the high performance metric optimised cases. Geometric values, like $R_t$, may be fixed in future studies to investigate other geometric properties, and their effect on the thrust and specific impulse of a system more in depth.

More in depth analysis of the parabolic nozzle geometry compared to the conical nozzle geometry can be initiated, where more direct comparisons may be completed. These may include comparisons with similar convergent and throat geometry, expansion ratio, and inlet conditions, where the difference lies exclusively in the divergent region geometry.

Investigations into parabolic nozzles at different scales, specifically in respect to the changing conditions, may be conducted. This would work to understand scale dependant effects like the boundary layer effect. These losses may be characteristic with respect to some geometric parameters, where an understanding of how sensitive they each may be to the scale would be ultimately understood.

Finally, verification of results through construction and experimentation of a nozzle using a supersonic wind tunnel or unit testing on a physical system (such as the MET) would complete the development of the nozzle and further validate the MDO process used to optimise the performance.

Considering the novel nature of the work in this thesis, an article is in preparation for publication to present Figures 4.5 to 4.7, and 5.2 to 5.9, alongside the optimised values of the objective functions and nozzles geometric configuration.

# Bibliography


[1] E. S. Agency, *Space debris: Is it a crisis?* https://www.esa.int/ESA_Multimedia/Videos/2025/04/Space_Debris_Is_it_a_Crisis, [Accessed 29-10-2025], 2025.

[2] L. Conde and p. Institute of Physics (Great Britain), *An introduction to plasma physics and its space applications. Volume 2, Basic equations and applications* (IOP ebooks. [2020 collection]), eng. Bristol [England] (Temple Circus, Temple Way, Bristol BS1 6HG, UK): IOP Publishing, 2020, ISBN: 0-7503-3542-4.

[3] S. H. Yeo, H. Ogawa, D. Kahnfeld, and R. Schneider, "Miniaturization perspectives of electrostatic propulsion for small spacecraft platforms," *Progress in Aerospace Sciences*, vol. 126, p. 100 742, 2021, ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2021.100742.

[4] F. G. Hey, *Micro Newton Thruster Development*. Springer Fachmedien Wiesbaden, 2018, ISBN: 9783658212094. DOI: 10.1007/978-3-658-21209-4.

[5] A. Chandler Karp and E. T. Jens, *Hybrid Rocket Propulsion Design Handbook*, eng, First edition. London, England: Academic Press, 2024, ISBN: 9780128165935.

[6] S. H. Yeo, D. Gadisa, H. Ogawa, and H. Bang, "Multi-objective design optimization and physics-based sensitivity analysis of field emission electric propulsion for cubesat platforms," *Aerospace Science and Technology*, vol. 154, p. 109 516, 2024, ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2024.109516.

[7] A. Runchal and S. ( service), *50 Years of CFD in Engineering Sciences A Commemorative Volume in Memory of D. Brian Spalding*, eng, 1st ed. 2020. Singapore: Springer Singapore, 2020, ISBN: 9789811526701.

[8] F. Magoules, *Computational fluid dynamics* (Chapman and Hall/CRC numerical analysis and scientific computation series), eng, 1st ed. Boca Raton: Chapman and Hall/CRC, 2012, ISBN: 0-429-10757-9.

[9] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, en. Boston, MA: Addison Wesley, Jan. 1989.

[10] K. D. Lihui Wang Amos H. C. Ng, *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*. Springer London, 2011, ISBN: 9780857296528. DOI: 10.1007/978-0-85729-652-8.

[11] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA Journal*, vol. 41, no. 4, pp. 687–696, 2003. DOI: 10.2514/2.1999.

[12] A. S. Pillai, *Introduction to rocket science and space exploration*, eng. Palm Bay, Florida ; Apple Academic Press, 2023, ISBN: 9781003323396.

[13] M. S. Kisenko, "Comparative results of tests on several different types of nozzles," in *National Advisory Committee for Aeronautics*, vol. 1066, Central Aero-Hydrodynamical Inistitue, 1944.

[14] G. V. R. RAO, "Exhaust nozzle contour for optimum thrust," *Journal of Jet Propulsion*, vol. 28, no. 6, pp. 377–382, 1958. DOI: 10.2514/8.7324.

[15] G. V. R. RAO, "Approximation of optimum thrust nozzle contour," *ARS Journal*, vol. 30, no. 6, pp. 561–563, 1960. DOI: 10.2514/8.5151.

[16] E. Rathakrishnan and E. Rathakrishnan, "Method of characteristics," eng, in *Applied Gas Dynamics*, United Kingdom: John Wiley  Sons, Incorporated, 2019, pp. 309–328, ISBN: 9781119500452.

[17] J. S. Evans and C. J. Schexnayder Jr., "Influence of Chemical Kinetics and Unmixedness on Burning in Supersonic Hydrogen Flames," *AIAA Journal*, vol. 18, no. 2, pp. 188–193, Feb. 1980. DOI: 10.2514/3.50747.

[18] G. V. Candler and J. N. Perkins, "Effects of vibrational nonequilibrium on axisymmetric hypersonic nozzle design," in *AIAA PAPAER 91-0297*, 1991.

[19] G. Cai, J. Fang, X. Xu, and M. Liu, "Performance prediction and optimization for liquid rocket engine nozzle," *Aerospace Science and Technology*, vol. 11, no. 2, pp. 155–162, 2007, ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2006.07.002.

[20] J. Bahamon and M. Martinez, "Study of fluid-dynamic behavior in a convergent–divergent nozzle by shape optimization using evolutionary strategies algorithms," *Proceedings of the Institution of Mechanical Engineers, Part G*, vol. 237, no. 12, pp. 2844–2862, 2023. DOI: 10.1177/09544100231163372.

[21] M. Matsunaga, C. Fujio, H. Ogawa, Y. Higa, and T. Handa, "Nozzle design optimization for supersonic wind tunnel by using surrogate-assisted evolutionary algorithms," *Aerospace Science and Technology*, vol. 130, p. 107 879, 2022, ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2022.107879.

[22] G. P. Sutton and O. Biblarz, *Rocket Propulsion Elements*. New Delhi, India: Wiley, Jan. 2010, ISBN: 9788126525775.

[23] F. L. Torre, "Gas flow in miniaturized nozzles for micro-thrusters," Ph.D. dissertation, Delft University of Technology, 2011.

[24] K. Suenaga, S. H. Yeo, T. Ozawa, and H. Ogawa, "Physical insights into downsized nozzle geometry for microwave electrothermal thruster via evolutionary algorithms," in *AIAA SCITECH 2024 Forum*, 2024. DOI: 10.2514/6.2024-2707.

[25] F. Durst and S. ( service), *Fluid Mechanics An Introduction to the Theory of Fluid Flows* (Graduate Texts in Physics), eng, 2nd ed. 2022. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, ISBN: 9783662639153.

[26] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009. DOI: https://doi.org/10.1002/nme.2579.

[27] M. Farrugia, K. Suenaga, S. H. Yeo, J. Gale, G. N. Iles, and H. Ogawa, "Multi-point design optimisation of microwave electrothermal thruster nozzles for robust in-space operation," in *IAF Space Propulsion Symposium*, International Astronautical Federation (IAF), 2025, p. x99999.

[28] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of Monte Carlo Methods* (Wiley series in probability and statistics), eng, 1. Aufl. Hoboken, N.J: Wiley, 2011, vol. 706, ISBN: 0470177934.

[29] I. Sobol, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Mathematics and Computers in Simulation*, vol. 55, no. 1, pp. 271–280, 2001, The Second IMACS Seminar on Monte Carlo Methods, ISSN: 0378-4754. DOI: https://doi.org/10.1016/S0378-4754(00)00270-6.

[30] A. B. Owen, "Orthogonal arrays for computer experiments, integration and visualization," *Statistica Sinica*, vol. 2, no. 2, pp. 439–452, 1992, ISSN: 10170405, 19968507. [Online]. Available: http://www.jstor.org/stable/24304869.

[31] "Multilayer perceptrons," in *Neural Networks in a Softcomputing Framework*. London: Springer London, 2006, pp. 57–139, ISBN: 978-1-84628-303-1. DOI: 10.1007/1-84628-303-5_3.

[32] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020. DOI: 10.1109/ACCESS.2020.2990567.

# Appendix A

# Code Extracts

## A.1 Python Code

### A.1.1 Mesh Generation Code

LISTING A.1: GMSH Bump Progression Solver

```
def f_transfinite(t, coef, length, nbpt):
    # coef < 1 case (bump)
    sqA = math.sqrt(1.0 - coef)
    a = (2.0 * sqA * math.log(abs((1.0 + 1.0 / sqA) / (1.0 - 1.0
        / sqA)))) / (nbpt * length)
    b = -a * length**2 / (4.0 * (coef - 1.0))  # note coef - 1.0
    d = length
    return d / (-a * (t * length - 0.5 * length)**2 + b)

def integrate_val(val_func, length, coef, npts, samples=2000):
    ts = np.linspace(0, 1, samples)
    vals = np.array([val_func(ti, coef, length, npts) for ti in
       ts])
    p = np.cumsum((vals[:-1] + vals[1:]) * (ts[1] - ts[0]) /
       2.0)
    p = np.insert(p, 0, 0.0)
    return ts, p

def compute_node_ts(ts, p, N):
    total_p = p[-1]
    b = total_p / (N - 1)
    target_ps = [k * b for k in range(N)]
    return np.interp(target_ps, p, ts)

def first_cell_size_bump(coef, length, n):
    ts, p = integrate_val(f_transfinite, length, coef, n)
    t_nodes = compute_node_ts(ts, p, n)
    return t_nodes[1]*length
```

LISTING A.2: Nozzle geometric definition

```
unit_scale = 0.001 # units are default in metres

# Geometry inputs
rt      = X[1] * unit_scale # throat radius
rc      = X[2] * unit_scale # radius of throat circle
exsp    = X[3] # expansion ratio
qy      = X[4] # Q y position as % of vertical dist from N to E
thetaN = X[5] # theta N

# Geometry constants
thetaE = 21.6 # Converging entrance angle
Rent = 3.8 * unit_scale # Converging entrance radius
El = 0.51 * unit_scale # Entry length (ie how long the throat is
    b/w converging and diverging section)

# Entrance circle point
tc = (thetaE - 90) * math.pi / 180
cex = -El + rc * math.cos(tc)
cey = rc * math.sin(tc) + rc + rt

# Entry section
es = 0
ee = (Rent - cey) / math.sin(thetaE * math.pi / 180)
cx = -ee * math.cos(thetaE * math.pi / 180) - cex
cy = cey + ee * math.sin(thetaE * math.pi / 180)

# Throat
te = (thetaN - 90) * math.pi / 180
nx = rc * math.cos(te)
ny = rc * math.sin(te) + rc + rt

# Exit
Re = math.sqrt(exsp) * rt
Ln = (0.8 * (math.sqrt(exsp) - 1) * rt) / math.tan(15 * math.pi
   / 180)
ex = Ln
ey = Re

# Q point
qx = qy * (ey - ny) / math.tan(thetaN * math.pi / 180) + nx
qy = qy * (ey - ny) + ny

# Constant / defined points of the nozzle's parabola
Nx, Ny = nx, ny
Qx, Qy = qx, qy
Ex, Ey = ex, ey
```

### A.1.2 Automatic Fluent Control and Execution

LISTING A.3: Pyfluent implementation to run ANSYS Fluent

```
if not os.getenv('FLUENT_PROD_DIR'):
        import ansys.fluent.core as pyfluent
        try:
            flglobals = pyfluent.setup_for_fluent(
                product_version="25.2.0", mode="solver",
                dimension=2, precision="double", processor_count
                =4, graphics_driver="dx11", ui_mode="gui")
        except:
            flglobals = pyfluent.setup_for_fluent(
                product_version="25.1.0", mode="solver",
                dimension=2, precision="double", processor_count
                =4, graphics_driver="dx11", ui_mode="gui")
        globals().update(flglobals)

    # Bring in case file
    cur_dir  = os.getcwd()
    #sim_file = cur_dir + "/case_files/" + my_file_name + "/
        Gmsh_output/case_1.cas.h5"
    sim_file = cur_dir + "/case_files/" + my_file_name + ".cas.
        h5"
    solver.settings.file.read_case(file_name = sim_file)

    save_folder = "sim_results/" + my_file_name
    try: os.mkdir(save_folder)
    except Exception as e: pass

    # Define settings
    solver.settings.setup.general.solver.type = "density-based-
        implicit"
    solver.settings.setup.general.solver.two_dim_space = "
        axisymmetric"
    solver.settings.setup.models.energy = {"enabled" : True}
    solver.settings.setup.models.viscous.model = "laminar"
    solver.settings.setup.materials.database.copy_by_name(type =
        "fluid", name = "argon")
```

```
25    solver.settings.setup.materials.fluid['argon'] = {"density"
          : {"option" : "ideal-gas"}, "viscosity" : {"sutherland"
          : {"effective_temperature" : 144.4, "
          reference_temperature" : 273.11, "reference_viscosity" :
           2.125e-05, "option" : "three-coefficient-method"}, "
          option" : "sutherland"}, "thermal_conductivity" : {"
          option" : "kinetic-theory"}, "molecular_weight" : {"
          value" : 39.948, "option" : "constant"}, "
          lennard_jones_length" : {"value" : 4, "option" : "
          constant"}, "lennard_jones_energy" : {"value" : 100, "
          option" : "constant"}}
26    solver.settings.setup.cell_zone_conditions.fluid["fluid"].
          general.material = ("argon")
27    solver.settings.setup.cell_zone_conditions.fluid["vacuum"].
          general.material = ("argon")
28    solver.settings.setup.boundary_conditions.set_zone_type(
          zone_list = ["inlet"], new_type = "pressure-inlet")
29    solver.settings.setup.boundary_conditions.set_zone_type(
          zone_list = ["outlet"], new_type = "pressure-outlet")
30    solver.settings.setup.boundary_conditions.set_zone_type(
          zone_list = ["wall-58", "wall-59", "wall-60", "wall-61",
           "wall-62", "wall-63"], new_type = "axis")
31    solver.execute_tui("/mesh/check")
32    solver.execute_tui("/mesh/repair-improve/repair")
33    # Note: supersonic_or_initial_gauge_pressure is the value to
           change, gauge_total_pressure stays constant
34    solver.settings.setup.boundary_conditions.pressure_inlet['
          inlet'] = {"momentum" : {"
          supersonic_or_initial_gauge_pressure" : {"value" :
          _settings_master.inlet_pressure}, "gauge_total_pressure"
           : {"value" : 54000.}}, "thermal" : {"total_temperature"
           : {"value" : _settings_master.inlet_temperature}}}
35    solver.settings.setup.boundary_conditions.pressure_outlet['
          outlet'] = {"momentum" : {"gauge_pressure" : {"value" :
          _settings_master.outlet_pressure}}}
36    solver.settings.setup.general.operating_conditions.
          operating_pressure = 0.
37    solver.settings.setup.reference_values.zone = "vacuum"
38    solver.settings.setup.reference_values.area = 1.
39    solver.settings.setup.reference_values.density = 1.225
40    solver.settings.setup.reference_values.enthalpy = 0.
41    solver.settings.setup.reference_values.length = 1.
42    solver.settings.setup.reference_values.pressure = 0.
43    solver.settings.setup.reference_values.temperature = 288.16
44    solver.settings.setup.reference_values.velocity = 1.
45    solver.settings.setup.reference_values.viscosity = 1.7894e-5
46    solver.settings.setup.reference_values.
          ratio_of_specific_heats = 1.4
```

```
47      solver.settings.setup.reference_values.yplus = 300.
48      solver.settings.setup.reference_values.compute(
            from_zone_type="pressure-inlet")
49      solver.settings.solution.methods.high_speed_numerics.enable
            = True
50      solver.settings.solution.methods.spatial_discretization.
            gradient_scheme = "green-gauss-cell-based"
51      solver.settings.solution.initialization.initialization_type
            = "hybrid"
52      solver.settings.solution.initialization.hybrid_init_options.
            general_settings = {"iter_count" : 200}
53      solver.settings.solution.initialization.hybrid_initialize()
54      solver.settings.file.read_journal(file_name_list=["
            solution_steering.jou"])
55      solver.settings.solution.monitor.residual.equations['
            continuity'].absolute_criteria = 1e-04
56      solver.settings.solution.monitor.residual.equations['x-
            velocity'].absolute_criteria = 1e-05
57      solver.settings.solution.monitor.residual.equations['y-
            velocity'].absolute_criteria = 1e-06
58      solver.settings.solution.monitor.residual.equations['energy'
            ].absolute_criteria = _settings_master.energy_limit
59      solver.settings.solution.monitor.residual.equations['
            continuity'].check_convergence = False
60      solver.settings.solution.monitor.residual.equations['x-
            velocity'].check_convergence = False # Keisuke doesn't
            check x-velocity
61      solver.settings.solution.monitor.residual.equations['y-
            velocity'].check_convergence = False
62      solver.settings.solution.monitor.residual.equations['energy'
            ].check_convergence = True
63      solver.settings.file.read_journal(file_name_list=["residual.
            jou"])
64
65      residuals_report = "plot/residuals-set/plot-to-file " +
            save_folder + "/residuals.csv"
66      solver.execute_tui(residuals_report)
67      solver.execute_tui("/plot/residuals")
68      solver.settings.solution.monitor.residual.options.n_save =
            iterations
69
70      # Run simulation
71      solver.settings.solution.run_calculation.iterate(iter_count
            = iterations)
72
73      # Results
74      pressure_contour = solver.settings.results.graphics.contour.
            create(name = "Pressure contour")
```

```
75      pressure_contour.field = "total-pressure"
76      pressure_contour.display()
77      solver.settings.results.graphics.views.auto_scale()
78      pressure_pic = save_folder + "/total-pressure-contour.png"
79      solver.settings.results.graphics.picture.save_picture(
            file_name=pressure_pic)
80      mach_no_contour  = solver.settings.results.graphics.contour.
            create(name = "Mach No. contour")
81      mach_no_contour.field  = "mach-number"
82      mach_no_contour.display()
83      solver.settings.results.graphics.views.auto_scale()
84      machno_pic = save_folder + "/mach-no-contour.png"
85      solver.settings.results.graphics.picture.save_picture(
            file_name=machno_pic)
86      velocity_contour = solver.settings.results.graphics.contour.
            create(name = "Velocity contour")
87      velocity_contour.field = "velocity-magnitude"
88      velocity_contour.display()
89      solver.settings.results.graphics.views.auto_scale()
90      velocity_pic = save_folder + "/velocity-magnitude-contour.
            png"
91      solver.settings.results.graphics.picture.save_picture(
            file_name=velocity_pic)
92
93      # Save data
94      case_data = save_folder + "/" + my_file_name + "_done.cas.h5
            "
95      solver.settings.file.write(file_type="case-data", file_name=
            case_data)
96
97      # Write mass flow rate data to .txt file
98      fluxes_report = save_folder + "/mass_flow_rate.txt"
99      solver.settings.results.report.fluxes.mass_flow(zones=["
            inlet", "interface-interface"],write_to_file=True,
            file_name=fluxes_report)
100
101     profiles_report = "file/write-profile " + save_folder + "/
            profiles.csv inlet interface-interface () pressure total
            -pressure density velocity-magnitude axial-velocity
            radial-velocity mach-number lambda2-criterion
            temperature total-temperature wall-temp cell-partition-
            active cell-partition-stored partition-neighbors"
102     partition-active cell-partition-stored partition-neighbors"
103     solver.execute_tui(profiles_report)
104
```

```
105     pressure_report = "file/write-profile " + save_folder + "/
            pressure.csv wall-58 wall-59 wall-60 wall-61 wall-62
            wall-63 () dynamic-pressure absolute-pressure total-
            pressure"
106     solver.execute_tui(pressure_report)
107
108     # End session
109     solver.exit()
```

## A.2 MATLAB Code

### A.2.1 Convergence Criteria

LISTING A.4: Ensuring specific impulse does not exceed theortical maximum value

```
1  function [T_ideal,Isp_ideal,mdot_ideal,pe] = cal_ideal(a_r)
2      pa = 13.0;
3      sigma = 0.7262;
4      R = 208.11;
5      A_thr = pi * rt^2 * 10^-6; % millimetres to metres
           conversion
6      T0 = 1200;
7      p0 = 150000;
8      mdot_ideal = p0 * A_thr * sigma / (sqrt(R * T0));
9      kappa = 1.66667;
10
11     Ae = pi * re^2 * 10^-6;
12
13     fun = @(Me) root3d(a_r,kappa,Me);
14     Me = fsolve(fun,2);
15
16     pe_p0 = 1/(1 + (kappa - 1)/2.0 * Me^2)^(kappa/(kappa-1.0));
17
18     pe = pe_p0 * p0;
19
20     ue = sqrt(2.0*kappa*R*T0/(kappa-1.0) * (1-pe_p0)^((kappa-1)/
           kappa));
21     T_ideal = mdot_ideal * ue + (pe) * Ae;
22     Isp_ideal = T_ideal / (mdot_ideal * 9.8);
23 end
```

# Appendix B

# Initial Parameter Simulation Sensitivity Analysis

The sensitivity of simulation convergence with respect to geometric parameters was conducted in two parts. The ranges of values for each parameter in the parametric study and MDO were initially set to those outlined in table 3.3, with the final values used as found stable by the following analyses outlined in table 3.4.

## B.1 Part 1 - Using a LHS

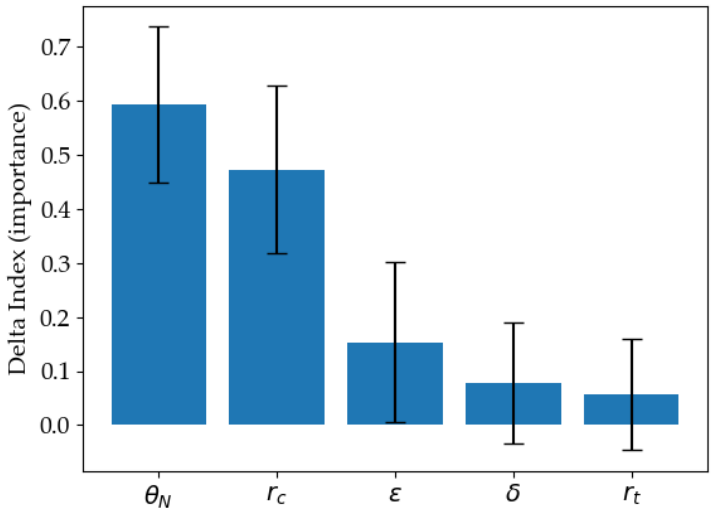


FIGURE B.1: The sensitivity of each parameter with respect to solution outcome.

This analysis was conducted using a LHS of dimension $M = 20$. The initial ranges of values was tied to those displayed in table 3.3. A grade was given to each case with respect to three simulation outputs: the mass flow rate difference between the inlet and interface, the pressure contour of the exhaust flow through the nozzle, and the pressure plot along the axis. High scoring nozzles performed well in all three aspects, while low scoring nozzles performed poorly or outright failed in all

aspects. Figure B.2 illustrates the pressure of a converged case and a diverged case, where convergence is evident by the smooth and continuous pressure distribution, alongside only a small drop in pressure along the entirety of the axis; the diverged case has a sharp drop in pressure, which is discontinuous. This was then fed into the Python SALib packages delta function, which is a tool used to correlate a sensitivity analysis and link parameters to outcomes. This package works best with data that was initially generated using a LHS type structure.

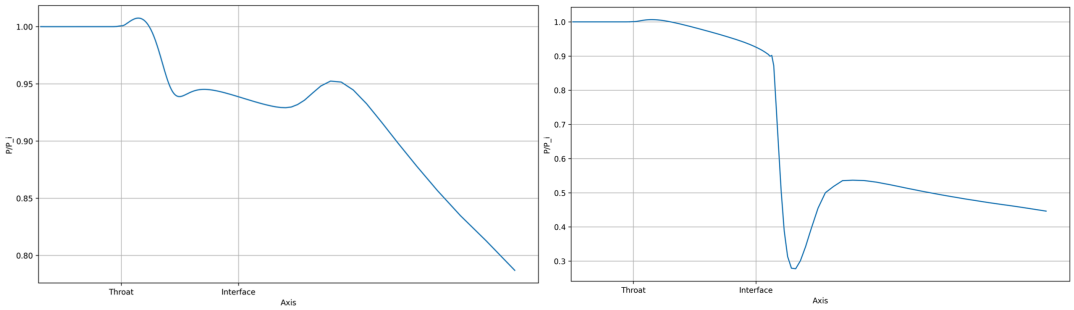


FIGURE B.2: Pressure plot along nozzle central axis of (Left) Converged case vs (Right) Diverged case.

The small population of the study limited the accuracy of the regression model, as is illustrated by figure B.1. While all values had relatively low confidence, $\theta_N$ was found to be most important to the outcome. Figure B.3 illustrates the range of values present in cases which produced good and bad outcomes (good indicative of an outcome with a score higher than $^{0.85}/_{1}$).

A random forest was also generated using the sci-kit learn Python package. The random forest algorithm is at its core a best-split decision-tree classifier, which may be used to estimate and predict outcomes based on model data. The core principles are those of regression analysis. The outcomes of this analysis on the model and data provided are illustrated in figure B.4.

In the context of this analysis, the random forest gives a prediction as to the sensitivity of a value to the convergence of a solution. As such, it confirms that $\theta_N$ and $\epsilon$ are very sensitive with respect to a solution achieving a realistic fluid flow.

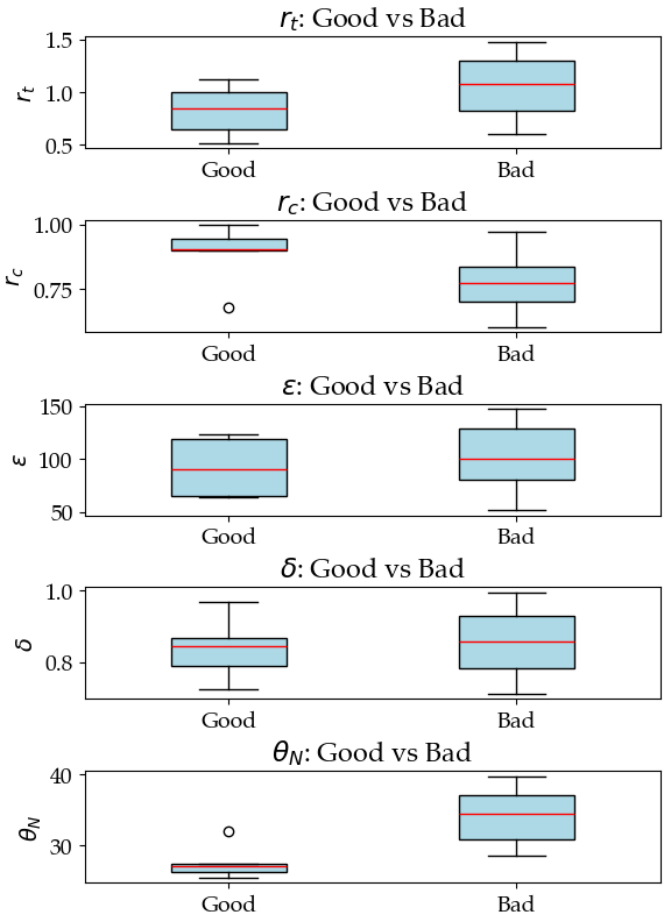


FIGURE B.3: The range of values prescribed to each parameter in good vs bad case outcomes.

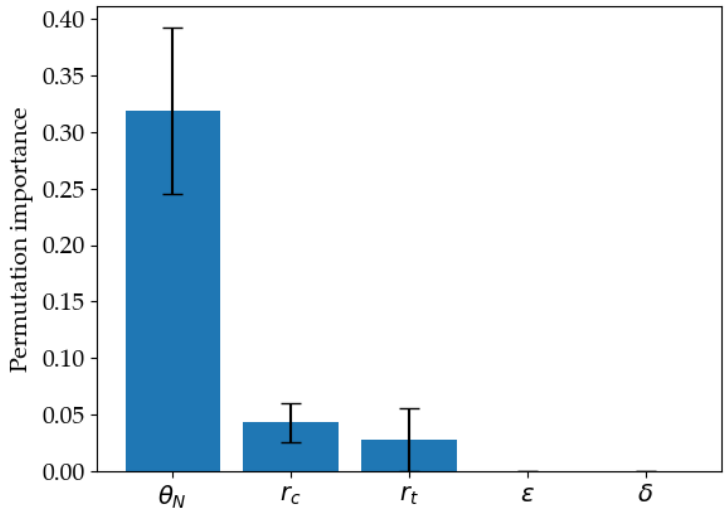


FIGURE B.4: Random forest prediction of most important parameters for solution outcome.

## B.2 Part 2 - Static confirmation

The static confirmation was conducted in parallel to the sensitivity analysis described in chapter 4, by first identifying a case which produced desirable results. Then, all parameters were held constant while one was varied between a maximum and minimum bounds. Table B.1 shows all case geometric parameters, along with their output parameters. Cases 21 and 51, both with $\theta_N$ values above 28.2°, failed to converge with respect to their pressure contour plots. This affirms the method described above to identify safe bounds. The study also provided an opportunity to tentatively test expanding the other bounds which showed little correlation to convergence failure. These bounds were further tested in an expanded parameter test using a LHS of dimension $M = 20$, of which 14 passed. The implicit value of $\theta_E$ was then calculated for each case, which had very strong correlation to the remaining non-convergence cases. Removing cases which generated with values of $\theta_E$ below 6.5° further improved convergence. Out of a test using a LHS of dimension $M = 20$, 19 were deemed to have converged.

TABLE B.1: Case input parameters and result parameters of the second sensitivity analysis.

| Case No. | $R_t$ | $r_c$ | $\epsilon$ | $\delta$ | $\theta_N$ | $T$ | $T_{axial}$ | $\eta_D$ | $I_{sp}$ | Converged |
|---|---|---|---|---|---|---|---|---|---|---|
| Unit | mm | mm | | | ° | N | N | % | s | |
| 1 | 0.67 | 0.86 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.45 | 1 |
| 2 | 0.50 | 0.86 | 71.39 | 0.78 | 23.79 | 0.06 | 0.06 | 99.47 | 107.57 | 1 |
| 3 | 0.83 | 0.86 | 71.39 | 0.78 | 23.79 | 0.17 | 0.17 | 99.53 | 109.05 | 1 |
| 4 | 1.16 | 0.86 | 71.39 | 0.78 | 23.79 | 0.33 | 0.33 | 99.48 | 109.83 | 1 |
| 5 | 1.50 | 0.86 | 71.39 | 0.78 | 23.79 | 0.56 | 0.56 | 99.40 | 110.36 | 1 |
| 6 | 0.67 | 0.60 | 71.39 | 0.78 | 23.79 | 0.10 | 0.10 | 99.53 | 108.42 | 1 |
| 7 | 0.67 | 0.73 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.43 | 1 |
| 8 | 0.67 | 0.93 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.45 | 1 |
| 9 | 0.67 | 1.00 | 71.39 | 0.78 | 23.79 | 0.11 | 0.11 | 99.52 | 108.46 | 1 |
| 10 | 0.67 | 0.86 | 50.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.49 | 108.46 | 1 |
| 11 | 0.67 | 0.86 | 83.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.42 | 1 |
| 12 | 0.67 | 0.86 | 116.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.31 | 1 |
| 13 | 0.67 | 0.86 | 150.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.51 | 108.21 | 1 |
| 14 | 0.67 | 0.86 | 71.39 | 0.70 | 23.79 | 0.11 | 0.10 | 99.49 | 108.48 | 1 |
| 15 | 0.67 | 0.86 | 71.39 | 0.85 | 23.79 | 0.11 | 0.10 | 99.54 | 108.42 | 1 |
| 16 | 0.67 | 0.86 | 71.39 | 0.92 | 23.79 | 0.11 | 0.10 | 99.56 | 108.40 | 1 |
| 17 | 0.67 | 0.86 | 71.39 | 1.00 | 23.79 | 0.11 | 0.10 | 99.58 | 108.38 | 1 |
| 18 | 0.67 | 0.86 | 71.39 | 0.78 | 20.00 | 0.11 | 0.10 | 99.14 | 108.97 | 1 |
| 19 | 0.67 | 0.86 | 71.39 | 0.78 | 26.00 | 0.11 | 0.10 | 99.59 | 108.21 | 1 |
| 20 | 0.67 | 0.86 | 71.39 | 0.78 | 23.00 | 0.11 | 0.10 | 99.48 | 108.54 | 1 |
| 21 | 0.67 | 0.86 | 71.39 | 0.78 | 29.00 | 0.10 | 0.10 | 99.64 | 107.88 | 0 |
| 22 | 0.40 | 0.86 | 71.39 | 0.78 | 23.79 | 0.04 | 0.04 | 99.40 | 106.76 | 1 |
| 23 | 0.64 | 0.86 | 71.39 | 0.78 | 23.79 | 0.10 | 0.10 | 99.52 | 108.33 | 1 |
| 24 | 0.88 | 0.86 | 71.39 | 0.78 | 23.79 | 0.19 | 0.19 | 99.52 | 109.19 | 1 |
| 25 | 1.12 | 0.86 | 71.39 | 0.78 | 23.79 | 0.31 | 0.31 | 99.52 | 109.18 | 1 |
| 26 | 1.36 | 0.86 | 71.39 | 0.78 | 23.79 | 0.46 | 0.46 | 99.47 | 109.63 | 1 |
| 27 | 1.60 | 0.86 | 71.39 | 0.78 | 23.79 | 0.65 | 0.64 | 99.41 | 110.00 | 1 |
| 28 | 0.67 | 0.63 | 71.39 | 0.78 | 23.79 | 0.10 | 0.10 | 99.53 | 108.42 | 1 |
| 29 | 0.67 | 0.70 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.53 | 108.44 | 1 |
| 30 | 0.67 | 0.76 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.44 | 1 |
| 31 | 0.67 | 0.83 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.44 | 1 |
| 32 | 0.67 | 0.89 | 71.39 | 0.78 | 23.79 | 0.11 | 0.10 | 99.53 | 108.46 | 1 |
| 33 | 0.67 | 0.96 | 71.39 | 0.78 | 23.79 | 0.11 | 0.11 | 99.52 | 108.45 | 1 |
| 34 | 0.67 | 0.86 | 55.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.50 | 108.46 | 1 |
| 35 | 0.67 | 0.86 | 73.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.44 | 1 |
| 36 | 0.67 | 0.86 | 91.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.53 | 108.39 | 1 |
| 37 | 0.67 | 0.86 | 109.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.53 | 108.35 | 1 |
| 38 | 0.67 | 0.86 | 127.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.52 | 108.27 | 1 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 39 | 0.67 | 0.86 | 145.00 | 0.78 | 23.79 | 0.11 | 0.10 | 99.51 | 108.20 | 1 |
| 40 | 0.67 | 0.86 | 71.39 | 0.68 | 23.79 | 0.11 | 0.10 | 99.49 | 108.49 | 1 |
| 41 | 0.67 | 0.86 | 71.39 | 0.73 | 23.79 | 0.11 | 0.10 | 99.50 | 108.47 | 1 |
| 42 | 0.67 | 0.86 | 71.39 | 0.79 | 23.79 | 0.11 | 0.10 | 99.52 | 108.44 | 1 |
| 43 | 0.67 | 0.86 | 71.39 | 0.83 | 23.79 | 0.11 | 0.10 | 99.53 | 108.43 | 1 |
| 44 | 0.67 | 0.86 | 71.39 | 0.91 | 23.79 | 0.11 | 0.10 | 99.56 | 108.40 | 1 |
| 45 | 0.67 | 0.86 | 71.39 | 0.97 | 23.79 | 0.11 | 0.10 | 99.57 | 108.39 | 1 |
| 46 | 0.67 | 0.86 | 71.39 | 0.78 | 21.00 | 0.11 | 0.10 | 99.28 | 108.80 | 1 |
| 47 | 0.67 | 0.86 | 71.39 | 0.78 | 22.80 | 0.11 | 0.11 | 99.53 | 107.83 | 1 |
| 48 | 0.67 | 0.86 | 71.39 | 0.78 | 24.60 | 0.11 | 0.11 | 99.60 | 107.62 | 1 |
| 49 | 0.67 | 0.86 | 71.39 | 0.78 | 26.40 | 0.11 | 0.11 | 99.64 | 107.32 | 1 |
| 50 | 0.67 | 0.86 | 71.39 | 0.78 | 28.20 | 0.11 | 0.11 | 99.68 | 106.96 | 1 |
| 51 | 0.67 | 0.86 | 71.39 | 0.78 | 30.00 | 0.11 | 0.11 | 99.67 | 107.04 | 0 |

# Appendix C

# Alternative MDO Results

A brief investigation into an alternative MDO solution was conducted using python and the *pymoo* package[32]. A NSGA-II algorithm was used, with surrogate models trained on the parametric studies set of data. 200 individuals were processed through 100 generations of a SAEA.

## C.1 Results

All output parameters were trained using a hybrid surrogate model, using both kriging and radial basis functions. The accuracy of these models was determined to be very high, with the root mean square error of the models being less than 0.01%. Given high confidence, the model was allowed to estimate output parameters exceeding the input data range by up to 20%.

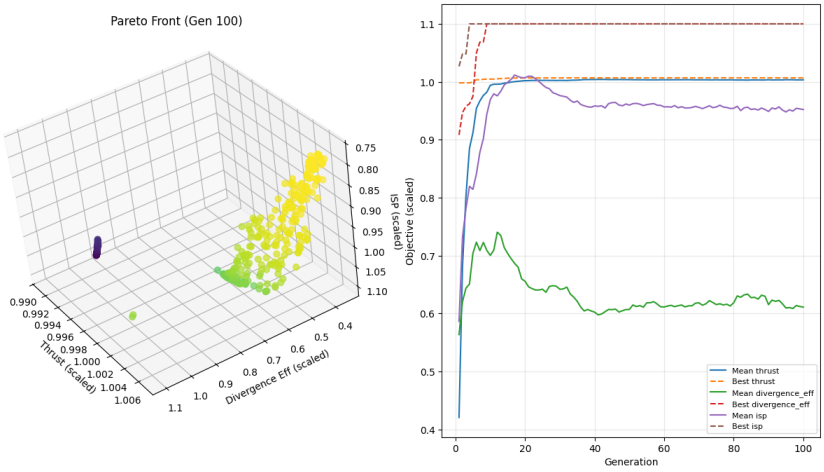


FIGURE C.1: MDO progress visualised, alongside the active generation plotted against the objective functions.

Figure C.1 illustrates the final output parameters of the 200$^{th}$ generation, alongside the training residuals over the MDO algorithm. The fitness of the generations

improved rapidly during the initial 20 iterations, and found balance by the $40^{th}$ iteration. Past 50 iterations, little improvement was found, as the pareto front was generally well resolved by this point.

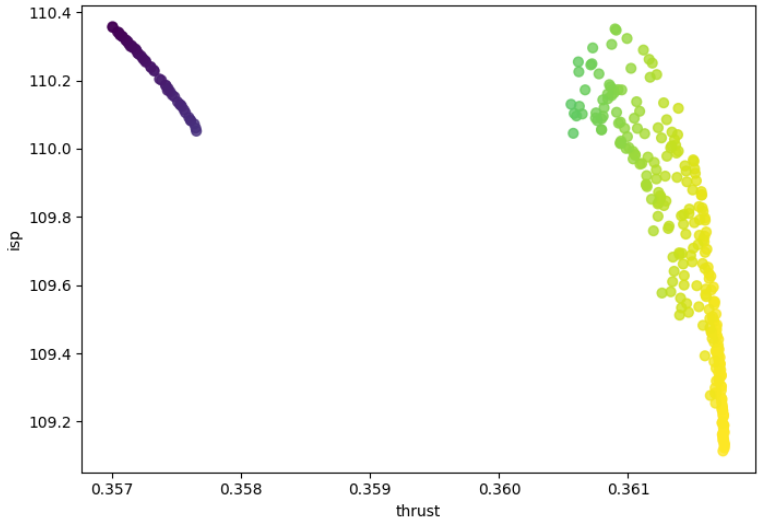


FIGURE C.2: Final generation of the MDO, plotted against their output parameters and coloured by thrust value (thrust vs specific impulse).

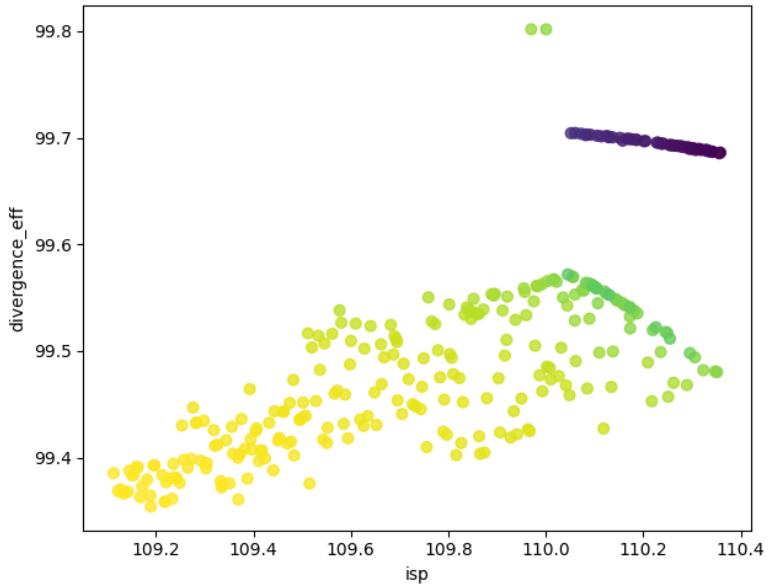


FIGURE C.3: Final generation of the MDO, plotted against their output parameters and coloured by thrust value (specific impulse vs divergent efficiency).

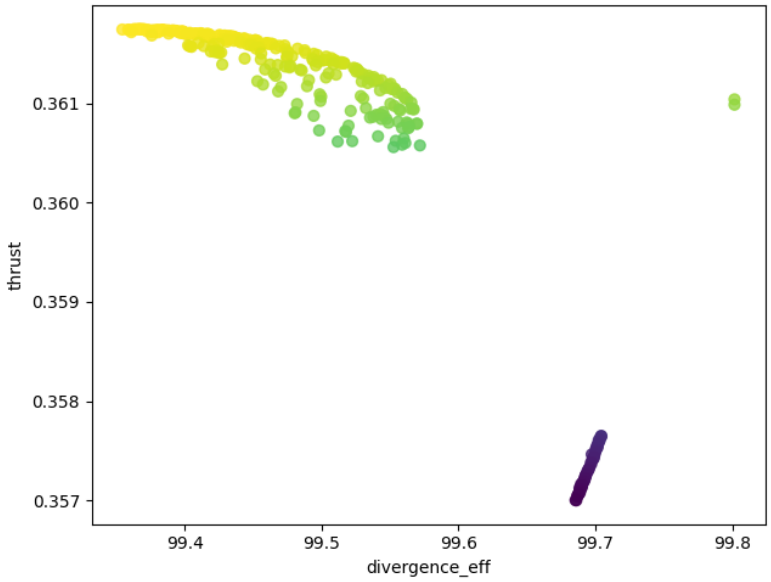


FIGURE C.4: Final generation of the MDO, plotted against their output parameters and coloured by thrust value (divergent efficiency vs thrust).

Figures C.2, C.3, and C.4 illustrate the final iteration of the MDO process using python. All points represent non-dominated individuals, and are colour coded by the estimated thrust of the individual. While clear relationships are present, the isolated high $I_{sp}$ low thrust group is indicative of local maximums being found. Since the pareto front is non-continuous, the validity of the result is pulled into question.

## C.2 Verification of Results

Three distinct cases were extracted from the final generation to examine their output parameters and verify the MDO results by conducting CFD simulations within ANSYS. The three cases chosen to be examined are each of the top performing output parameter cases: highest thrust, divergence efficiency, and specific impulse respectively.

### C.2.1 High Thrust Case

TABLE C.1: High thrust geometric parameters derived through python MDO

| Decision Variable | Value | Unit |
|---|---|---|
| $R_t$ | 1.199 | mm |
| $\epsilon$ | 119.9 | - |
| $\theta_N$ | 20.47 | ° |
| $\delta$ | 0.700 | - |
| $\varnothing_c$ | 0.938 | mm |
| $\varnothing_d$ | 0.938 | mm |

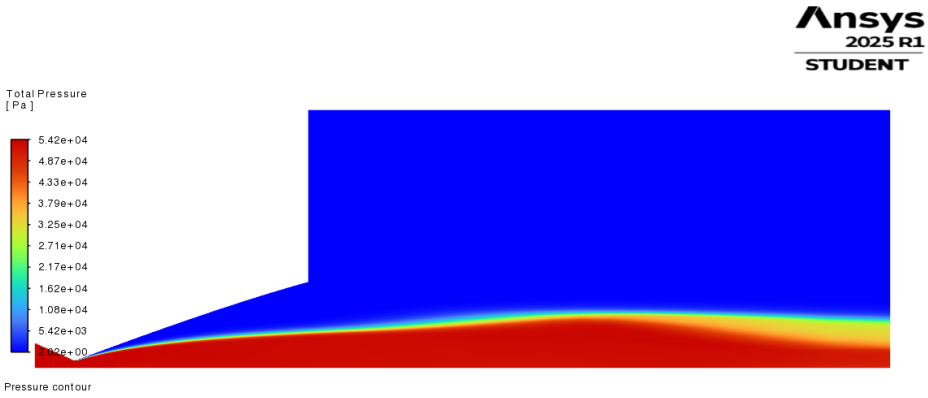


FIGURE C.5: CFD pressure contour of the high thrust case.

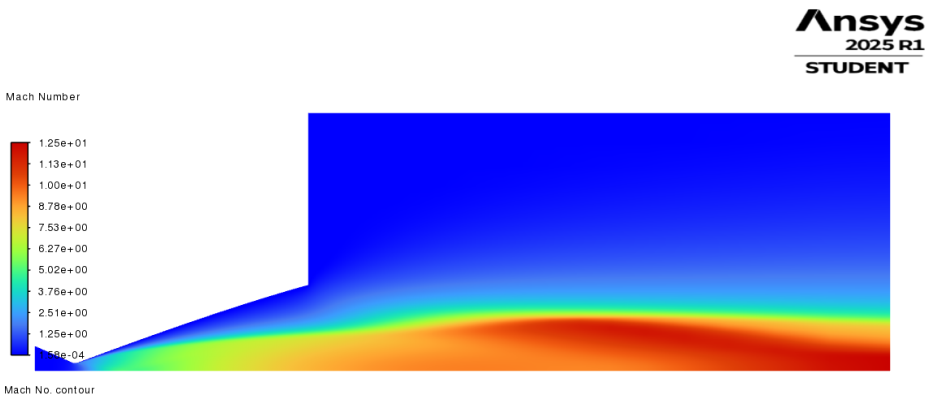


FIGURE C.6: CFD mach number contour of the high thrust case.

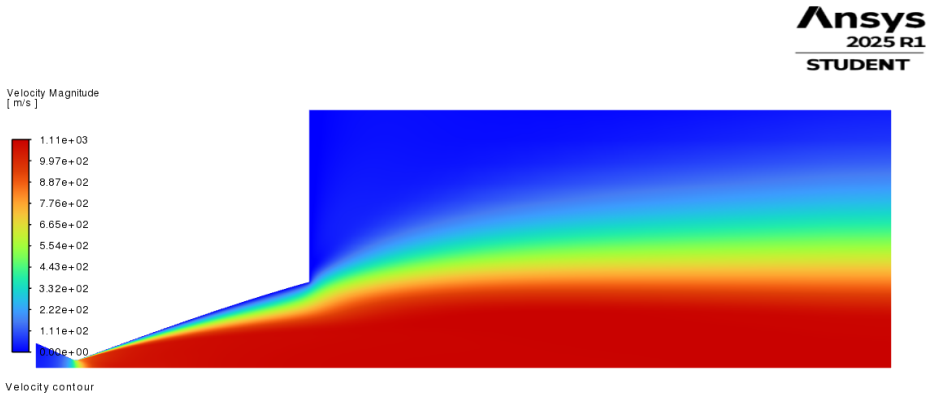


FIGURE C.7: CFD velocity contour of the high thrust case.

TABLE C.2: High thrust output parameters derived through python MDO

| Output | Estimated Value | Simulated Value | Unit | Difference (%) |
|---|---|---|---|---|
| Thrust | 0.362 | 0.358 | N | -1.105 |
| Divergence Efficiency | 99.35 | 99.28 | % | -0.070 |
| Specific Impulse | 109.1 | 110.3 | s | 1.100 |

The surrogate models and MDO process was validated using a CFD simulation of the nozzle geometry using the parameters generated. This case was simulated for 100000 iterations and converged with good residual values and excellent flow contour fields, as illustrated in Figures C.5, C.6, and C.7. The simulated values are compared to the estimated values in Table C.2. The thrust and divergence efficiency was overestimated, while the specific impulse was underestimated; all values are similar to their estimated counterparts with the largest difference being only 1.105%.

### C.2.2 High Specific Impulse Case

TABLE C.3: High specific impulse geometric parameters derived through python MDO

| Decision Variable | Value | Unit |
|---|---|---|
| $R_t$ | 1.199 | mm |
| $\epsilon$ | 119.9 | - |
| $\theta_N$ | 25.49 | ° |
| $\delta$ | 0.884 | - |
| $\varnothing_c$ | 0.850 | mm |
| $\varnothing_d$ | 0.850 | mm |

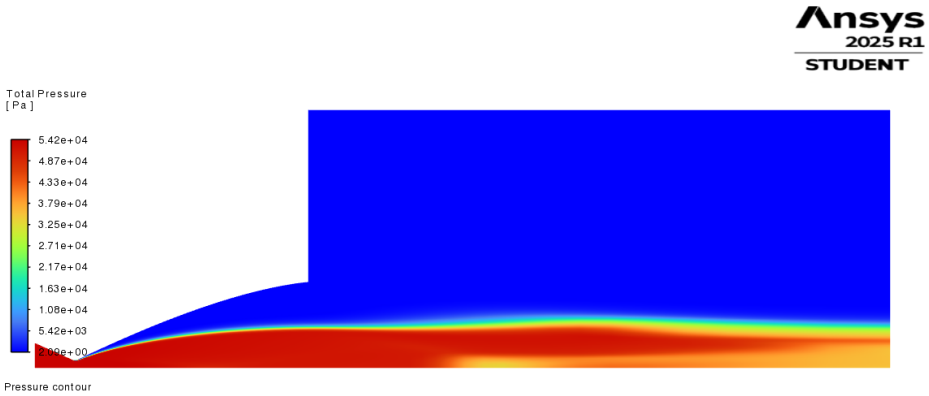


FIGURE C.8: CFD pressure contour of the high specific impulse case.

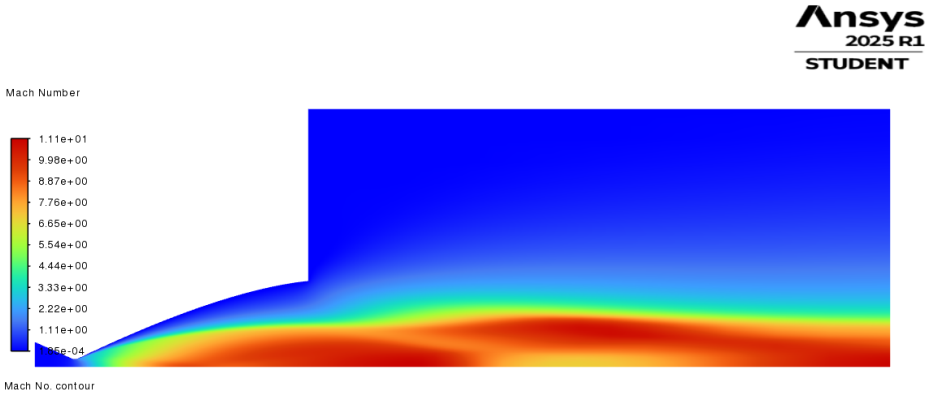


FIGURE C.9: CFD mach number contour of the high specific impulse case.

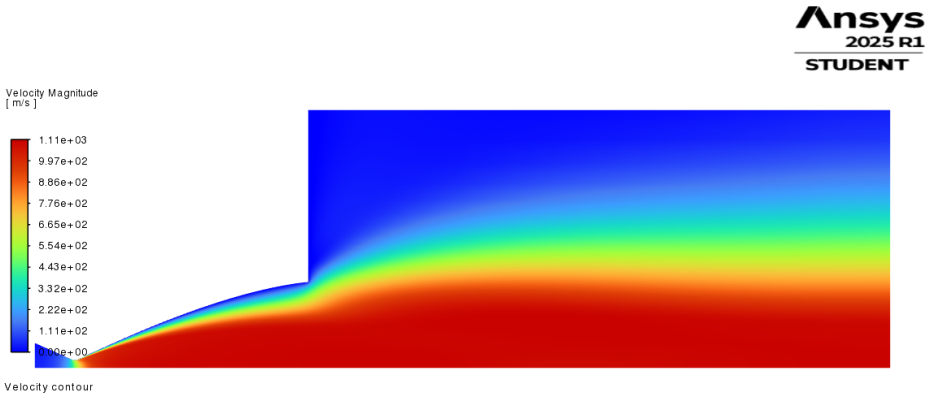


FIGURE C.10: CFD velocity contour of the high specific impulse case.

TABLE C.4: High specific impulse output parameters derived through python MDO

| Output | Estimated Value | Simulated Value | Unit | Difference (%) |
|---|---|---|---|---|
| Thrust | 0.357 | 0.356 | N | -0.280 |
| Divergence Efficiency | 99.69 | 99.64 | % | -0.050 |
| Specific Impulse | 110.4 | 109.7 | s | -0.634 |

The high specific impulse nozzle was simulated for 100000 iterations and converged with good residual values but very poor flow contour fields, as illustrated prominently in Figure C.5. Severe flow splitting can be observed downstream of the nozzle exit, which effects nozzle convergence. Thus, confidence in the results produced via this simulation are very low. While the estimated nozzle outputs are very similar to the actual simulated value from CFD simulation (albeit universally underestimated) the actual values undermine the key performance metric of this nozzle. The actual specific impulse value is lower than that of the actual specific impulse of the high thrust nozzle. This can be attributed to the bad convergence of the flow fields via flow splitting.

### C.2.3 High Divergence Efficiency Case

TABLE C.5: High divergence efficiency geometric parameters derived through python MDO

| Decision Variable | Value | Unit |
|---|---|---|
| $R_t$ | 1.199 | mm |
| $\epsilon$ | 119.9 | - |
| $\theta_N$ | 25.49 | ° |
| $\delta$ | 0.700 | - |
| $\varnothing_c$ | 0.997 | mm |
| $\varnothing_d$ | 0.997 | mm |

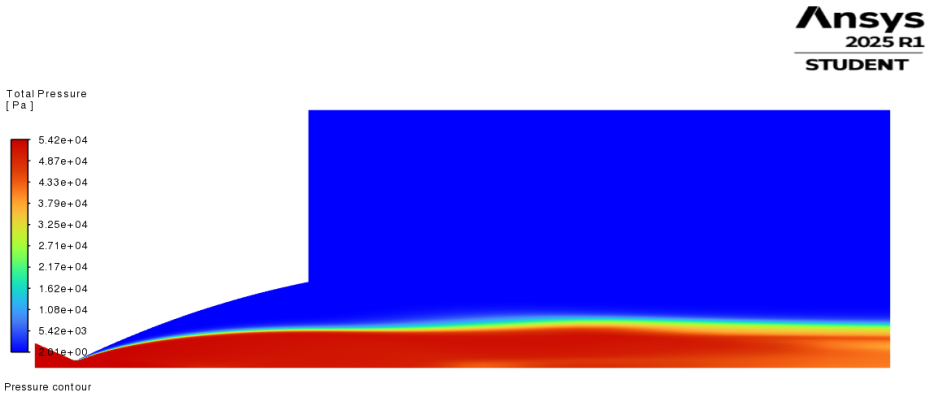


FIGURE C.11: CFD pressure contour of the high divergence efficiency case.

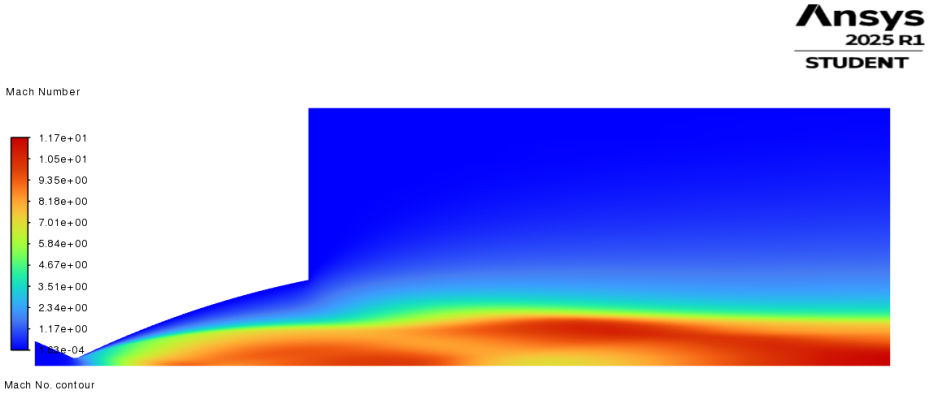


FIGURE C.12: CFD mach number contour of the high divergence efficiency case.

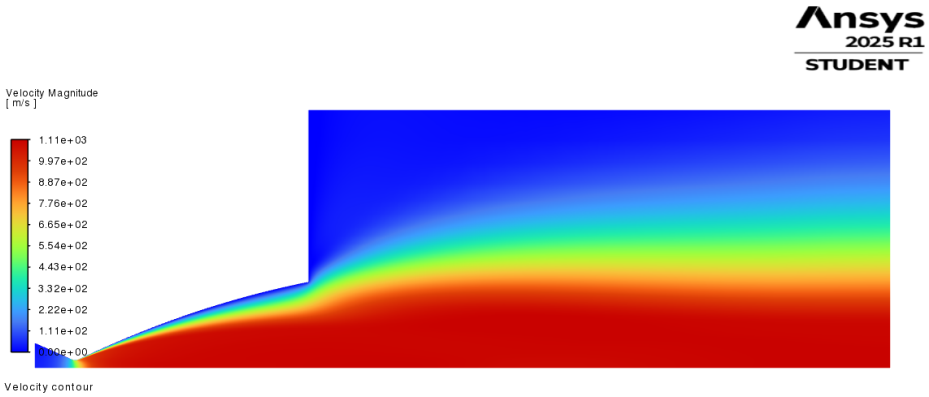


FIGURE C.13: CFD velocity contour of the high divergence efficiency case.

TABLE C.6: High divergence efficiency geometric parameters derived through python MDO

| Output | Estimated Value | Simulated Value | Unit | Difference (%) |
|---|---|---|---|---|
| Thrust | 0.361 | 0.357 | N | -1.108 |
| Divergence Efficiency | 99.80 | 99.61 | % | -0.190 |
| Specific Impulse | 110.0 | 109.8 | s | -0.182 |

The high divergence efficiency nozzle was simulated for 100000 iterations and converged with good residual values but poor flow contour fields, as illustrated prominently in Figure C.5. Flow splitting can be observed downstream of the nozzle exit, which effects nozzle convergence. Thus, confidence in the results produced via this simulation are low. However, the thrust and divergence efficiency accuracy was similar to those of the high thrust case, as shown in Table C.6. However, the specific impulse was overestimated, which can be linked to the flow splitting. Further, the actual divergence efficiency was less than the actual divergence efficiency of the high specific impulse nozzle, further undermining the key performance metric of this nozzle.

All three nozzles maximised the $R_t$ and $\epsilon$ input parameters to maximise the thrust of the nozzles, while $\theta_N$, $\delta$, and $\varnothing_{c,d}$ were varied to induce more specific impulse or divergence efficiency of the nozzle (as indicated by the sensitivity analysis). However, due to the insufficient confidence in the results due to the flow splitting effects, the nozzle geometries produced via the SAEA and MDO conducted using python cannot be used nor recommended with good confidence.