Response Surface Based Design Optimization of a Low-Pressure Microresistojet's Heat Exchanger

MSc Thesis

Anant Chandra

Delft University of Technology

TUDelft

Response Surface Based Design Optimization of a Low-Pressure Microresistojet's Heat Exchanger

by

Anant Chandra

Student number:	4997972	
Project duration:	January, 2021 - July, 2021	
Thesis supervisors:	Prof. Angelo Cervone	TU Delft (supervisor)
	Dr.-Ing. Dejan Petkow	SPARC Industries SARL (co-supervisor)
Thesis committee:	Prof. Angelo Cervone	TU Delft
	Dr.-Ing. Dejan Petkow	SPARC Industries SARL
	Prof. Eberhard Gill	TU Delft (Chair)
	Prof. Saullo Giovani Pereira Castro	TU Delft (External examiner)
Institution:	Delft University of Technology	
Place:	Faculty of Aerospace Engineering, Delft	

Cover Image: Cross-section of microchannels generated using 8 control points positioned using Sobol sequences and joined by a NURBS curve.

TU Delft

Acknowledgement

I would like to thank my thesis supervisor, Prof. Angelo Cervone, for this opportunity to work on a satellite propulsion system, which is something that I have wanted to do for a long time. Your guidance and quick responses are much appreciated as is giving me the freedom to explore and implement, new and novel concepts. I received a lot of help and guidance in the field of DSMC simulations and on general aspects of the thesis from my co-supervisor, Dejan Petkow, and from his team at SPARC Industries SARL that included Sander Rouwette, Fa Zhu, Sawssen Chibani, and Ullrich Siems.

I also thank two of my close friends for their help. Anirudh Bharath helped me understand the NURBS formulation, offered troubleshooting help with the Gmsh software and also proofread this report. Sushrut Deshpande explained the concept of low discrepancy sequences, guided me through the derivation of the Sobol sequence algorithm and introduced me to the relation between neural networks and Hilbert's 13^{th} problem.

Finally, I thank my parents who supported my decision to study Space Engineering at TU Delft, a field that has enraptured me since early childhood.

Anant Chandra
Delft, July 2021

Summary

A Low-Pressure Micro-resistojet (LPM) is a type of in-space electrothermal propulsion system for satellites that works by heating low-pressure (50 to 300 Pa) fluid flowing through microchannels/slots (typically <1 mm diameter) using resistive heating elements like thin-film Molybdenum. This thesis delineates a response surface based method to explore and optimize the cross-sectional design of microchannels, which, as documented in literature, have so far only been either rectangular or circular in shape.

Experimental studies place the efficiency of LPM systems at 25% which can be increased to 30% by reducing heat losses from the heat exchanger to the plenum. The concept is still in its early stages of development and therefore offers plenty of opportunities in the improvement of the propulsion system whether it be regarding the choice of propellant, design of propellant management system, choice of thruster materials or the design of the microchannels, which is the focus of this thesis. Usage of Response Surface Methodology (RSM) to optimize the design of a microchannel for greater thrust efficiency requires the following steps: Design of Experiment (DoE) run to select samples from the design space, performing experiments to evaluate the performance of selected samples, building a response surface (or surrogate model) from the generated data and finally using an optimization algorithm to find a high-efficiency design. In this thesis, these steps were performed using (respectively): Sobol Sequences, Direct Simulation Monte Carlo (DSMC) method using Sparc Industries' VSTRAP software, Convolutional Neural Network (CNN) and Genetic Algorithm (GA).

As a case study, the optimization was performed for Argon as the propellant, inlet pressure of 50 Pa, inlet temperature of 300 K and microchannel wall temperature of 600 K. The number of design parameters were set as 8 though the base geometry supports higher numbers as well. Using Sobol sequences to determine parametric values, 512 designs (termed polar designs) were generated. After DSMC simulations (validated with an experimental study in literature), the best performing design was found to have a thrust efficiency of 0.7019. 512 Rectangular designs were generated as well among which the best performing design had a thrust efficiency of 0.6566. From this dataset, the main takeaway is that the thrust efficiency peaks at a certain slot area and that small areal rectangular slots, common in literature, are not as efficient as broader slots with higher cross-sectional area (up to a limit). The CNN was trained on the polar dataset (CNN1) as well as the combination of both datasets (CNN2) to predict thrust efficiency from slot images. The predictions were, on average, within 10% from the simulation results. A GA run with 8-parameter geometry and with CNN1 as the response surface resulted in an optimum design with a thrust efficiency of 0.732. With CNN2, the optimized efficiency

for 8-parameter and 15-parameter geometry was 0.7281 and 0.7455 respectively. However, on simulating the GA results it was found that CNN1 and CNN2 had overestimated the thrust efficiency by $\approx 11\%$. Thus in this case study, the most thrust efficient design is the best one from the polar dataset. The conclusion drawn is that a high number of designs generated using Sobol Sequences are enough to find the best performing design for a given set of input conditions i.e. choice of propellant, inlet pressure and temperature, and heaterchip temperature. As it seems that there is a strong correlation between slot area and performance, the number of designs to be generated can be based on whether it spans the allowable minimum to the allowable maximum area set by the designer. However, if the correlation between slot area and performance is not found after a DoE run for a different set of input conditions, then a CNN would have to be trained and used by the GA for design optimization.

The thesis concludes with a design recommendation for the shape of a microchannel that may on experimental testing result in a high performing thruster. Improvement in LPM technology will further the capabilities of PocketQube satellites (5 x 5 x 5 cm^3 form factor) and also of large satellites for which it could function as precise attitude control thrusters. Low pressure operation allows for a compact and lightweight propellant management system and the thruster's high thrust-to-power ratio gives it a major advantage over competing propulsion systems such as hall-effect and electrospray thrusters.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
BGK	Bhatnagar-Gross-Krook model
CFD	Computational Fluid Dynamics
CNN	Convolution Neural Network
DoE	Design of Experiments
DRIE	Deep Reactive Ion Etching
DSMC	Direct Simulation Monte Carlo
FMMR	Free Molecular Micro Resistojet
GA	Genetic Algorithm
I_{sp}	Specific Impulse
Kn	Knudsen Number
LBM	Lattice Boltzmann method
LPM	Low Pressure Microresistojet
MEMS	Micro-electromechanical systems
MHX	Micro Heat Exchanger
NURBS	Non-uniform rational basis spline
px	pixels
ReLU	Rectified Linear Unit
RSM	Response Surface Methodology
SDF	Signed Distance Function
VSTRAP	Versatile Software Tool for RArefied Plasmas

Constants

Symbol	Definition	value
g_0	Gravitational acceleration at earth sea level	$9.8066\ m/s^2$
σ	Stefan-Boltzmann constant	$5.670374419 \times 10^{-8}\ Wm^{-2}K^{-4}$
k_B	Boltzmann constant	$1.38E\text{-}23\ m^2kgs^{-2}K^{-1}$

List of Figures

List of Tables

List of Listings

1

Introduction

Resistojets are high thrust–to–power propulsion systems that generate thrust by transferring energy to the fluid using resistive heating elements. These systems are typically employed in CubeSat (and smaller satellite) missions which have strict mass, volume, and power constraints. A subclass of Resistojets is the 'Free Molecular Micro Resistojet' (FMMR) / 'Low-Pressure Micro-Resistojet' (LPM) propulsion system in which the fluid flow is, typically, in the rarefied regime, i.e. when the Knudsen number (Kn, eq. 1.1) is greater than 0.1. Depending on the operating conditions, the fluid flow in the thruster, or sections of the thruster, could also fall into the slip-flow regime or even the continuum regime. The flow regimes are categorized as follows (Colin 2013): Continuum flow $\rightarrow$ Kn $< 10^{-3}$, Slip flow $\rightarrow 10^{-3} <$ Kn $< 10^{-1}$, Transition flow $\rightarrow 10^{-1} <$ Kn < 10, Free molecular flow $\rightarrow 10 <$ Kn.

$$\begin{aligned} \text{Kn} &= \frac{\textit{Mean free path of the molecule}, \lambda}{\textit{Characteristic length of geometry}, L} \\ &= k_2\sqrt{\gamma}\frac{\text{Ma}}{\text{Re}} \text{ (Colin 2013)} \end{aligned} \tag{1.1}$$

Where Ma is Mach number, Re is Reynolds number, γ is the heat capacity ratio, and k_2 is constant whose value depends on the collision model being considered for the problem. Its values are tabulated in table 2.1 of ref. Colin 2013.

A typical LPM system works as follows (refer fig. 1.1): Low-pressure gas, in the order of a few hundred Pascals, enters the plenum and then encounter slots, the sides of which are heated using resistive heaters. The gas particles gain energy from the slot walls on collision. The reflected particles can either move towards the plenum or towards space (depends on surface roughness). The latter direction is preferable for thrust generation. Low-pressure inflow allows the usage of solid (or dense liquid) propellants (like ice) which can sustain the propulsion system with just their vapour pressure (pressure can be increased using heaters). The result is

a low mass and low volume propellant feed system. In this text, the silicon wafer along with the resistance heater and the silicon nitride electrical insulation layer is collectively referred to as the 'heaterchip' which is the micro-heat exchanger (MHX) part of the LPM system. The heaterchip could be further coated with a layer of gold to reduce thermal losses due to radiation. For example, a 5 x 5 cm^2 surface at 600 K radiates to space (using eq. 1.2): 7.3 W (silicon nitride layer) and 0.367 W (thin-film gold layer).

$$P_{radiation} = \epsilon A \sigma T_c^4 \tag{1.2}$$

where A is the surface area, σ is the Stefan-Boltzmann constant = $5.670374419 \times 10^{-8}\ Wm^{-2}K^{-4}$, emissivity, ϵ, is 0.4 for silicon nitride (Zhang et al. 2020) and 0.02 for thin-film gold (Wong, Reed, and Ketsdever 2001, Van Zwol et al. 2015).

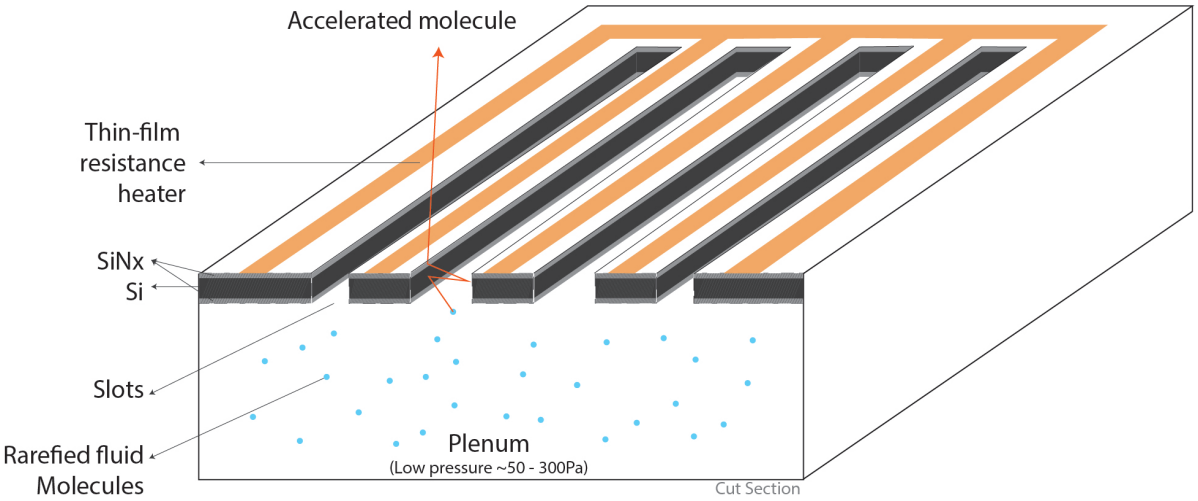


Figure 1.1: Schematic of an LPM propulsion system.

The need for LPM systems arises out of a need for a low thrust, low power and small form factor propulsion system for PocketQube (coined by Twiggs 2009) satellites which typically have a volume equal to one or more units of a 5 cm sided cube. Details about PocketQube applications can be found in Bouwmeester et al. 2020. Delft University's Delfi-PQ mission, which is currently in development, will feature an LPM propulsion system, the requirements for which are: thrust between 0.2 to 3 mN, peak power consumption $<$ 4 W, volume $<$ 42 x 42 x 30 mm and mass $<$ 75 g (Pallichadath et al. 2019).

The performance of an LPM system has a dependency on the design of its heaterchip due to its effect on the transmission coefficient (α) of the rarefied fluid flow (Daduí C. Guerrieri et al. 2018; Mancas, Cervone, and Zandbergen 2014). In an ideal geometry, all molecules will exit in the intended direction of thrust [1] and that all exiting molecules would have the same temperature as the wall. The ratio of the number of particles entering a system from the entrance plane to the

1. It is possible due to the surface roughness of the wall, that the particle will reflect back towards the plenum

number of particles exiting the system from the exit plane is known as *transmission coefficient* or *transmission probability* (Mancas, Cervone, and Zandbergen 2014; Binder et al. 2016). Analytically, Daduí C. Guerrieri et al. 2018 showed that for the same geometrical aspect ratio (depth of microchannel/smaller cross-sectional dimension), a rectangular and circular cross-sectional slot had a transmission coefficient of 0.36 and 0.19 respectively. Direct Simulation Monte Carlo (DSMC) simulations revealed that the transmission coefficient is also dependent on the wall temperature and the plenum pressure (Daduí C. Guerrieri et al. 2018). LPM performance can be quantified using 3 parameters: thrust, specific impulse and power required to maintain the heaterchip temperature. Analytically, Daduí C. Guerrieri et al. 2018 derived the following equations for thrust, specific impulse and power (eq. 1.3).

$$\begin{aligned} T &= \alpha P_0 A_e \frac{(\pi+2)}{2\pi}\sqrt{\frac{T_w}{T_0}\left(\frac{6\gamma}{\pi+6\gamma}\right)} \\ I_{sp} &= \frac{(\pi+2)}{g_0}\sqrt{\frac{k_B T_w}{2\pi m_a}\left(\frac{6\gamma}{\pi+6\gamma}\right)} \\ P &= \dot{m} C_p \left(T_w - T_0\right) \textit{where } C_p = \frac{(\zeta+2)}{2}\frac{k_B}{m_a} \end{aligned} \tag{1.3}$$

where α is the transmission coefficient, P_0 is plenum pressure, A_e is exit area, T_w is the microchannel wall temperature, T_0 is plenum temperature, γ is the specific heat ratio, g_0 is the gravitational acceleration at earth sea level ($= 9.8066\ m/s^2$), m_a is molecular mass, $\dot{m}$ is mass flow rate, C_p is the specific heat at constant pressure, ζ is the thermodynamic degree of freedom, and k_B is the Boltzmann constant.

Note that the specific impulse in eq. 1.3 is independent of the transmission coefficient, α, but this is due to the assumption that all molecules exiting the thruster are attaining the wall temperature. It is possible that some molecules will exit the thruster without striking the hot wall and thus not contribute significantly to thrust. Hence, specific impulse, which can be written as $I_{sp} = \frac{T}{\dot{m} g_o}$, will be less. Furthermore, power consumption will decrease as well because the particle has not taken energy from the wall. It is thus important to analyse heaterchip designs with DSMC simulations.

Designs can be evaluated by running Computational Fluid Dynamics (CFD) simulations. Typically, the Navier-Stokes equation forms the basis of Computational Fluid Dynamics (CFD) codes that simulate fluid flow in the continuum regime. However, in the rarefied regime, the Navier-Stokes equations are not valid (Hadjiconstantinou 2006) and instead equations operating at the molecular level are needed, such as the Boltzmann equation. However, evaluating the collision integral in the Boltzmann equation is computationally expensive. A realistic alternative is to use the Direct Simulation Monte Carlo (DSMC) method. The DSMC method is a stochastic model designed by G. A. Bird in the 1960's (G. A. Bird 1970), which tracks the motion of every molecule in the system deterministically and allows for inter-atomic

collisions in a stochastic manner. It has been mathematically shown that the DSMC method, for a limiting case of infinite particles, converges to the solution of the Boltzmann equation (Wagner 1992). The method has been extensively tested in literature and is, so far, the preferred method for analysing fluid flow in LPM designs. In this work, SPARC Industries' ("SPARC Industries SARL," n.d.) DSMC solver, called Versatile Software Tool for RArefied Plasmas (VSTRAP), has been used. An open-source alternative is the dsmcFoam+ solver (White et al. 2018).

The design space of microchannel cross-sectional shapes has not yet been explored beyond rectangular and circular shapes. Ketsdever et al. 1998, Daduí C Guerrieri et al. 2016, Daduí C Guerrieri et al. 2017 and Mancas, Cervone, and Zandbergen 2014 have performed basic geometry optimization which at most involved varying a few parameters such as aspect ratio of rectangular slots, slot placement, and convergent and divergent angles of the nozzle. Note that varying angles of a section of the microchannel, which goes through-and-through the silicon wafer, using Microelectromechanical systems (MEMS) processes is not straightforward and a study regarding its manufacturability needs to be carried out.

To generate a heaterchip design that can yield a high thrust-to-power and high specific impulse heater-chip design for a given set of constraints, the Response Surface Methodology (RSM) was deemed suitable as an optimization methodology after consulting various works in the literature. The work by Muñoz-Paniagua and García 2020 and Vytla, Huang, and Penmetsa 2010 on design optimization of the nose of a high-speed train touch on all aspects of the RSM. In RSM, the goal is to build a response-surface (also known as a surrogate model, approximate model, and meta-model) which is essentially an approximation of the objective function (also known as the response variable, loss function, and cost function). The reason to build a response surface is to replace costly and time-consuming experiments. The term 'experiments' include computer simulations such as CFD and DSMC simulations. To build the response surface, samples must be selected from the design space such that the design space is sufficiently explored i.e. samples should be uniformly distributed throughout the design space. The design space is a multidimensional space, also known as a hypercube, where each dimension corresponds to a parameter controlling the design. The selection of sample points from the design space is known as Design of Experiments (DoE). After a DoE run and evaluation of the sample points using experimentation, the data is fed to a response surface builder for which many techniques exist, though in recent times Neural Networks are gaining popularity. Finally, the optimization algorithm (deterministic or stochastic) can use the response surface to evaluate designs when they are generated during the optimization iterations. The choice of methods used in this project is as follows:

- Design of Experiments (DoE): Sobol Sequences (implementation algorithm: Bratley and Fox 1988)
- Experiment: VSTRAP
- Response-surface: Convolutional Neural Network (CNN)
- Optimization: Genetic Algorithm (GA)

The thesis addresses the research questions:

1. By how much can the performance (thrust, power requirement and specific impulse) of an LPM system be increased by solely (i.e. with fixed boundary conditions) optimizing the MHX geometry?

    (a) Can the GA method supported by a neural network surrogate of a DSMC solver be used to optimize the design of the MHX geometry?
    (b) What should the architecture of the CNN method look like such that it efficiently (takes less time to train and is $> 90\%$ accurate) maps an image of the MHX geometry and input parametric values to the performance of the MHX?
    (c) How close are the results by the CNN to the DSMC results?
    (d) Is a CNN model a sufficient replacement of a DSMC solver for comparing the performance of MHX designs?
    (e) How large should the dataset (composed of MHX designs and their DSMC-derived performance values) used for training the CNN be such that the CNN is $> 90\%$ accurate

2. How close is VSTRAP's results to the experimental results of LPM systems documented in literature?

The report is divided into the following sections: Chapter 2 (MHX Designs: Review) provides a review of the current status of MHX designs both inside and out of the satellite propulsion domain, Chapter 3 (RSM: Method and Setup) delineates the methods used and their setup, Chapter 4 (Geometry generation) describes the base geometry and how a complete geometry is created from parametric values, Chapter 5 (RSM: Results) contains the results, Chapter 6 (Conclusion) contains the general inferences and recommendations that can be drawn from the results, Chapter 7 (Recommendations) informs the reader on the possible directions that one can take to further this work.

2

MHX Designs: Review

In an MHX system, the fluid-carrying channels have a micro-sized (less than 1 mm) characteristic length. The complete heat exchanger can have a large form factor, for industrial applications, containing 1000's of microchannels or a small form factor of 5 x 5 x 5 cm^3 like that of an LPM system containing around 50 microchannels. Typically, in industrial applications (ex. powerplants), the microchannels are etched on metal plates by micro-machining or deep x-ray lithography. The plates are then diffusion bonded (pressing the plates together at 60-70% of melting temperature) to yield a high convective heat transfer heat exchanger (Thulukkanam 2013). Apart from metals, highly conductive ceramics like SiC (Xie et al. 2020) and AlN (Fattahi et al. 2020) have also been investigated. Xie et al. 2020 states that ceramics can withstand harsh conditions (high temperatures, radiation, shock and corrosion) better than metals, silicon, glass and polymers. Most microchannel designs in literature are straight through-and-through the wafer with a rectangular cross-section (fig. 2.1) and its optimization revolves around finding the optimum combination of 3 parameters: width, height and number of microchannels. The popularity of the shape is possibly due to its ease of manufacturability though due to the advances in microfabrication technology, other shapes are increasingly being explored many of which have been documented in the review papers: Alihosseini et al. 2020 and Naqiuddin et al. 2018.

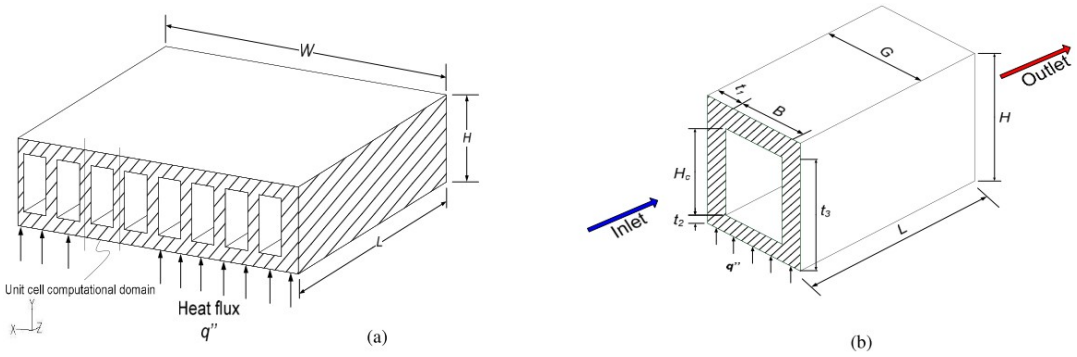
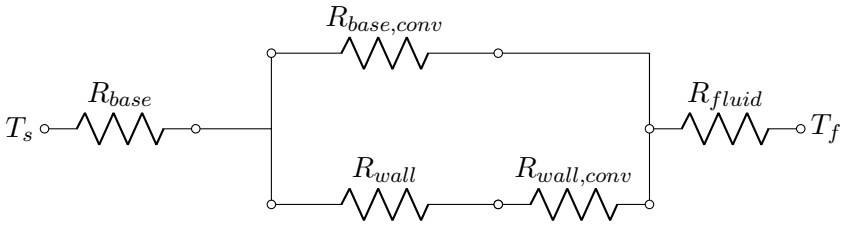


Figure 2.1: Schematic of a straight rectangular micro-heat exchanger. The height, H, and width, W, typically range from 10 to 200 μm. Image sourced from Ighalo, Bello-Ochende, and Meyer 2009.

For a rectangular cross-sectional straight microchannel, Ighalo, Bello-Ochende, and Meyer 2009 used the DYNAMIC-Q optimization algorithm coupled with ANSYS Fluent (a CFD solver) and Gambit (mesh generator) to optimize the design (minimize wall temperature) of a heat sink. The design variables for geometric optimization were t_1, t_2, t_3, H *and* G as shown in fig. 2.1. The constraints were: solid-to-void volume ratio between 0.3 to 0.8, constant total volume and, assuming the DRIE manufacturing process which is popular for etching silicon substrates, a maximum aspect ratio[1] = 20 and a minimum wall thickness of 50 μm. The objective of the optimization was to increase heat transfer and decrease the pressure drop. Other studies presented in this section have a similar structure; Design variables are identified and then optimized using some algorithm to maximize heat transfer and minimize pressure drop along the microchannels. Shao et al. 2009 reduced the geometry to a thermal resistance network and performed *multi-objective optimization* (Sequential quadratic programming) to come up with a geometric configuration that maximized heat transfer (by minimizing thermal resistance) and minimized pressure drop. For the optimized geometry, the total thermal resistance was calculated using 2 methods: thermal resistance method and CFD (used for verification). In the thermal resistance method, the thermal resistances lying in between the source, T_s, and the sink, T_f, are determined using empirical equations. In the thermal resistance network illustrated below – R_{base}, R_{wall}, $R_{wall,conv}$, $R_{base,conv}$, R_{fluid} are the thermal resistances of the base (referring to the surface in fig. 2.1 subjected to the heat flux), the side walls, the interface between side-walls and the fluid, the interface between the base and the fluid, and the fluid flowing in the microchannel respectively. However, the empirical equations simplify the relation between fluid flow and heat transfer by establishing a relation between Reynold number and the Nusselt number respectively. In a CFD simulation the fluid flow is simulated by solving the Navier–Stokes equations (not applicable for rarefied fluid) and are more accurate, albeit computationally expensive. The total thermal resistance calculated by both methods differed by 8%.

1. Here aspect ratio is the ratio of the etching depth to the characteristic length of the cross-section.



Shao points out in his later paper (Baodong et al. 2011) that gradient-based methods like DYNAMIC-Q tend to give locally optimum solutions and multi-objective optimization solutions depend heavily on the weights that are assigned, often arbitrarily, by the user. The solution, according to Baodong et al. 2011, is to use *genetic algorithms*. This technique is analogous to biological evolution; A starting group of 'N' configurations, each with a unique set of parameters, are allowed to mutate (changes in the parameter value) and 'evolve'. Low performing branches are removed, and eventually, only high performing designs remain. Statistical methods such as the *Design for Six Sigma (DFSS)* method (coupled with the ANSYS Fluent CFD solver) and the *Box-Behnken design* method (experimentally validated) by Jiang et al. 2017 and Hamidnia et al. 2018 respectively, has been used to optimize the geometry as well as the input conditions for optimal performance i.e. high heat transfer and low-pressure drop.

Study of other microchannel shapes include work by Ma et al. 2016 who etched 2 different configurations of a zigzag patterned microchannel to increase heat transfer (vortices result in better fluid mixing) but the designs suffered from greater pressure drop. Note that the straight rectangular cross-sectional shape is the reference case for comparisons. Kuppusamy et al. 2015 claims that their design of triangular grooves in a rectangular microchannel, for the correct selection of geometric parameters, yielded a design that has both enhanced heat transfer and slightly reduced pressure drop. The selection of geometric parameters was done by changing parametric values one by one, simulating the flow and observing the trends in data. Lee, Wong, and Zohar 2002 describes a *CMOS*-compatible fabrication process of a triangular cross-sectional microchannel over a silicon substrate. Luo, Fan, and Tondeur 2007 states that the usage of fractal geometrical cross-section (ex. fig. 2.2) can increase heat transfer by at least double for every increase in the fractal scale albeit with a pressure drop increase by 24%. Apart from the cross-section, the shape (pathway) of the microchannel has also been experimented with. Luo, Fan, and Tondeur 2007 in their research found that introducing turns in the microchannel (but making sure that all microchannels have the same total length) resulted in lower pressure drops for the same heat transfer performance. A fractal branching network of microchannels was found to increase total heat transfer and decrease the pressure drop when compared to parallel channels (Chen and Cheng 2002). Alihosseini et al. 2020 has reviewed a wide variety of papers concerned with the analysis of different channel and channel cross-sectional shapes.

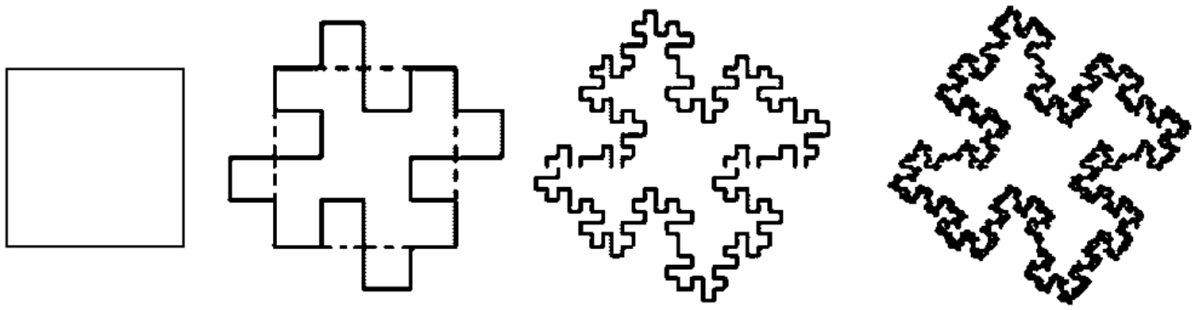


Figure 2.2: Different scales of a Koch island fractal. Image sourced from Luo, Fan, and Tondeur 2007.

2.1. Rarefied Micro Heat Exchangers

When the fluid flow is in the continuum regime then at the fluid-surface interface, the fluid velocity is 0 and the fluid temperature is same as that of the surface. This is not applicable to rarefied fluid flows in which the flow velocity at the surface is non-zero and the fluid temperature at the surface is not the same as that of the surface. This is also known as *slip flow*. To model heat transfer in slip flow conditions, authors have proposed models for specific geometries that are essentially modified versions of the Navier-Stokes or the Boltzmann equation. Scherer and Barichello 2009 and Misdanitis and Valougeorgis 2008 used the Bhatnagar-Gross-Krook (BGK) (Bhatnagar, Gross, and Krook 1954) model, which is the Boltzmann Equation with modifications introduced in the collision term, to model the heat transfer between 2 parallel plates. According to Vocale, Morini, and Spiga 2014, continuum fluid flow formulations are still valid in slip flow conditions but only beyond one mean free path from the wall surface. Thus, by modifying the boundary conditions to capture the rarefaction effects on the velocity and temperature of the fluid, the Navier-Stokes equations could be used even for rarefied fluid flow (Yu and Ameel 2001). Using this approach, Pamela and Marco 2014 modelled the heat transfer in an asymmetrically heated circular microchannel and Vocale, Morini, and Spiga 2014 did it for an elliptical microchannel for which they found that, for an aspect ratio (length/breadth) < 7.3, Kn = 0.1, and Nitrogen as the fluid, convection was higher in an elliptical microchannel than in a rectangular microchannel. For both elliptical and rectangular designs, Vocale, Morini, and Spiga 2014 observed an increase in convective heat transfer with a decrease in aspect ratio and/or a decrease in Knudsen number. Various triangular microchannels were simulated using the BGK model by Naris and Valougeorgis 2008 for the Kn range from free molecular to continuum. Using the Navier-Stokes equations with rarefaction effects incorporated in the boundary conditions, Yu and Ameel 2001; Kuddusi and Çetegen 2006; Kuddusi 2007 simulated slip flow in rectangular microchannels for different geometries and boundary values. Sofloo, Shams, and Ebrahimi 2009 and Yang et al. 2010 used the DSMC method to simulate heat transfer in a rectangular microchannel.

2.2. Low-Pressure Microresistojet Micro Heat Exchangers

The LPM system was first conceptualised by Ketsdever et al. 1998. In their work, they presented the design of the system (fig. 2.3) and evaluated its performance using the DSMC method. The characteristic length of the geometry is less than or only slightly more than the mean free path of the propellant. At such small scales, long ($\approx$ 1cm) and narrow slots (1 to 100 μm) are preferred over a small nozzle expansion (nozzle throat $\approx 20\mu m$) due to the increased risk of clogging by contaminants.

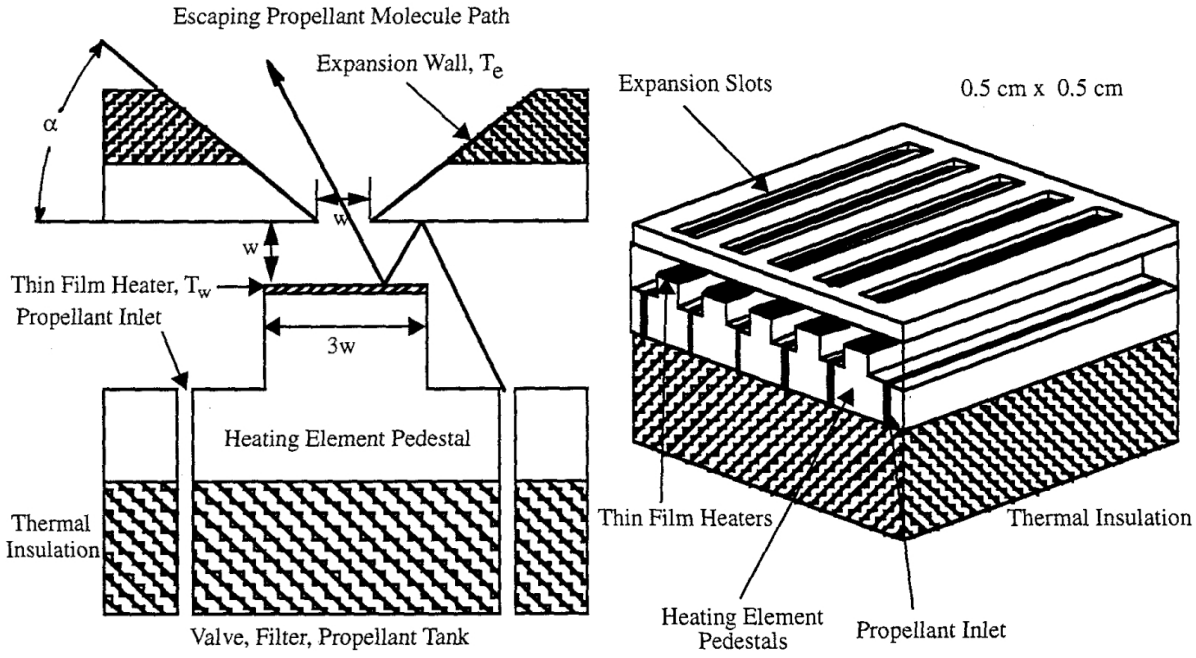


Figure 2.3: Schematic design of the LPM system by Ketsdever et al. 1998. Width, w, is 100 μm.

For the DSMC simulation, the diffusive [2] gas-surface interaction model was used. The simulation was performed for a single slot with a slot depth of 250 μm, width (w) of 100 μm and heater width (3w) of 300 μm (see fig. 2.3). The following parameters were varied to observe their effect on performance: heater wall temperature, heater surface area, propellant, Knudsen number (Kn) and slot divergence (α). Non-heater surfaces were maintained at 300 K. The following trends were observed:

- Increasing the wall temperature increased the specific impulse (I_{sp} [3]).
- Increasing molecular weight decreased the I_{sp}.
- With Argon as the propellant, increasing the divergent angle from 40°, increased the I_{sp} till the divergent angle was approximately 50° after which I_{sp} decreased.

2. Gas molecules attain the same temperature as the wall and scatter according to the Lambertian cosine scattering law.

3. Defined as the change in momentum per unit of propellant consumed

- With Argon as the propellant, increasing Kn decreased I_{sp}, though, after Kn = 0.5, the change in ISP and thrust was minimal.
- Heater width = 3w = 300 μm was found to provide a reasonable compromise between performance and input heating power. Increasing the heater width increases the heating area due to which the I_{sp} increases up to a point after which it plateaus. At the same time, input power increases as more heat is transferred to the gas molecules and also radiated out due to the increased surface area.

The resulting system (Argon as the propellant, wall temperature = 600 K, and Kn = 1), according to the DSMC simulation, had a thrust of approximately 0.025 μN per slot with an I_{sp} of 45 seconds. To increase thrust the authors suggest increasing: pressure, slot width, slot length, and the number of slots. This will allow for a higher mass flow rate without affecting the flow regime. Ketsdever et al. 1998 recommends water (ice) as a propellant stored at 245 K (open to vacuum). Ice is dense and will require much smaller propellant tanks than Noble gases (commonly used in plasma propulsion systems). At 245 K and vacuum conditions, the vapour pressure of ice is 50 Pa for a 100 μm slot.

Later, Ketsdever, Lee, and Lilly 2005 experimentally tested the LPM system with a different design of the heaterchip (fig. 2.4). In this design, the number of slots is 44 where each slot has the dimension: 100 μm wide, 5.375 mm long and 500 μm deep (etched through-and-through the silicon wafer). At 112 Pa Argon and chip temperature = 575 K, the mass flow rate was 50 sccm, I_{sp} = 51 s, and thrust = 0.0007 N.

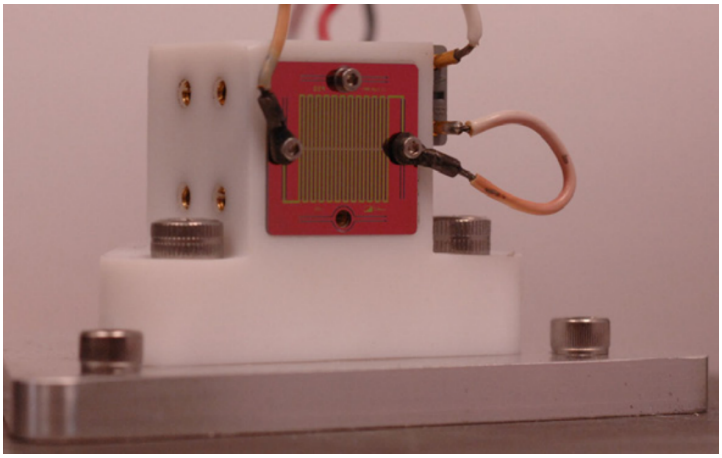


Figure 2.4: Image of the fabricated LPM system by Ketsdever, Lee, and Lilly 2005 bolted onto a Teflon casing (plenum).

The concept of a water propellant for LPM was explored by Mancas, Cervone, and Zandbergen 2014. In this work, the geometry is the same as fig. 2.3. The propellant is stored in the form of ice and its vapour pressure and temperature are maintained with the help of heaters. Optimization of the LPM MHX geometry was done by varying parametric values and evaluating the performance using the *dsmcFoam* solver (OpenFOAM, n.d.). They found that increasing the

slot length and adding an exit divergence (as opposed to a cuboid slot geometry) increased the thrust efficiency ($= \frac{Thrust \times I_{sp} \times g_0}{2(Power_{chip} + Power_{tank})}$). The optimized slot geometry is illustrated in fig. 2.5. For a thruster with ten 5 mm wide slots made of silicon (coated with gold to reduce radiative losses), the LPM system had a simulated performance as follows: I_{sp} = 70 - 110 s, thrust = 1 - 1.6 mN, power requirement = 0.8 - 5.6 W and total mass <380 g.

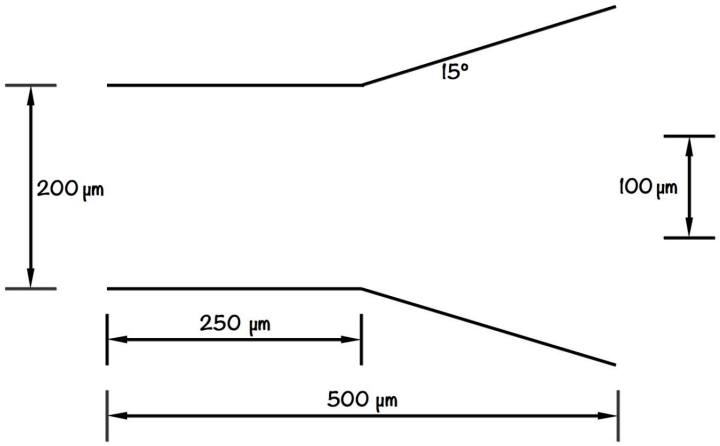


Figure 2.5: Optimized slot geometry of a water propelled LPM system. Image sourced from Mancas, Cervone, and Zandbergen 2014.

Daduí C Guerrieri et al. 2016 and Daduí C Guerrieri et al. 2017 continued the work by Mancas, Cervone, and Zandbergen 2014 by evaluating different geometrical designs by simulations and experiments. For the geometrical design of the MHX illustrated in fig. 2.6, Daduí C Guerrieri et al. 2016 varied the parameter, L, and the wall temperature to observe their effect on the simulated performance of the thruster. Daduí C Guerrieri et al. 2017 manufactured and compared 3 slot geometries as illustrated in fig. 2.7. All 3 slot geometries are cut into the same heater chip (1st image in fig. 2.7). Daduí C. Guerrieri et al. 2018 formulated analytical equations for the simple geometries: "short uniform circular cross-section" and "short uniform rectangular cross-section".

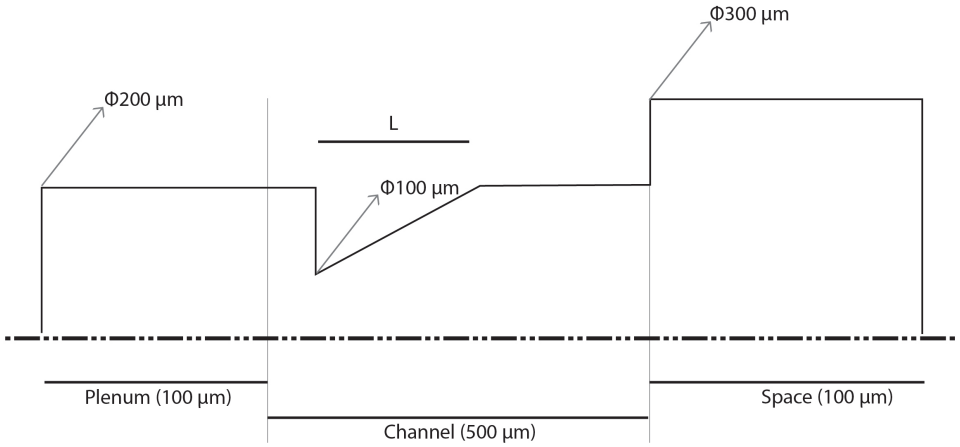


Figure 2.6: Geometry setup for the optimization problem. Image reproduced from Daduí C Guerrieri et al. 2016.

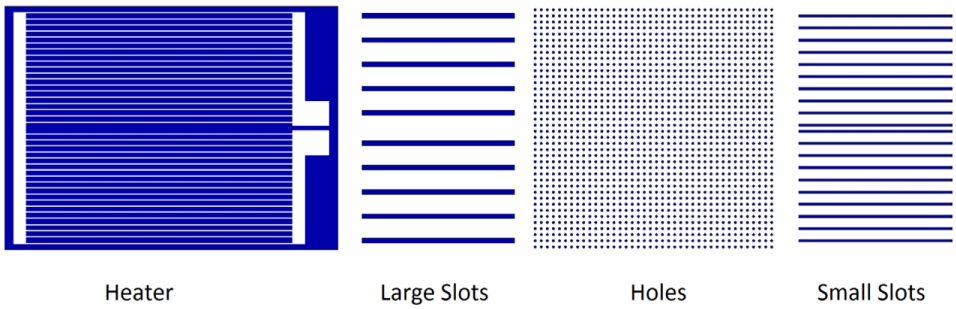


Figure 2.7: Heater chip (leftmost) followed by slot geometries. The slots are cut into the heater chip. Image sourced from Daduí C Guerrieri et al. 2017.

Three main groups have worked on the fabrication and testing of LPM systems the details of which have been documented in Ketsdever et al. 2000 and Tew et al. 2000 (USA Air force Research Laboratory and University of Southern California), Palmer, Nguyen, and Thornell 2013 (Uppsala University), and Daduí C Guerrieri et al. 2017 (TU Delft). The characterization process for the heater chip was found to be the same across all groups i.e. after fabrication, the LPM system undergoes, in order, Mechanical, Electrical and Propulsion characterization.

In mechanical characterization, the dimensions of the LPM system is checked against the initial design, generally, with the help of a microscope. Next, the electrical characterization step involves plotting a resistance vs heaterchip temperature curve and then using that data to determine the temperature coefficient (β) using eq. 2.1. Eq. 2.1 is only valid for a narrow temperature band wherein the resistance is varying with temperature linearly.

$$\beta = \frac{R - R_0}{R_0(T - T_0)} \tag{2.1}$$

Where β is the temperature coefficient of resistance, R is the resistance of the film heater measured at heaterchip temperature T, R_0 is the resistance of the thin-film heater at room temperature T_0. β is different for every material and for thin films it even depends on the thermal coefficient of expansion between the film and the substrate (Hall 1968). For most metals, the resistance increases with an increase in temperature and for semi-conductors, resistance decreases with an increase in temperature because the valence electrons start occupying the conduction band. For some polymer thermistors, resistance increases with temperature and then decreases after a certain temperature (Okutani et al. 2020).

If the resistance variance is not linear with temperature then the Steinhart-Hart equation (eq. 2.2, Steinhart and Hart 1968) should be used instead.

$$T^{-1} = A + B \log R + C(\log R)^3 \tag{2.2}$$

Where T is the heaterchip temperature in K, R is resistance, and A, B, and C are constants which are determined from the resistance of the thin-film heater vs heaterchip temperature curve.

During operation, eq. 2.1 or eq. 2.2 can be used to predict the temperature of the heater chip by measuring the resistance of the heaterchip. This removes the need for the installation of a

separate temperature sensor. To plot these curves, Ketsdever et al. 2000 used an aluminium-6061 block to uniformly heat the heater-chip from 300 K to 600 K and measured the resistance at intermediate steps. Palmer, Nguyen, and Thornell 2013 used a temperature-controlled oven to change the temperature and note the changes in resistance. The temperature was raised in steps of 5° from 25 to 75°. Daduí C Guerrieri et al. 2017 used an Infrared camera to gauge the heater-chip temperature, which was painted with graphite ink to increase the emissivity to 0.97.

Propulsion characterization requires, at the very least, a vacuum chamber replicating, to an extent, the space environment, and a thrust stand that can work in the vacuum chamber and be able to detect thrust from, roughly, 0.1 to 5 mN. In Ketsdever et al. 2000, the thruster is mounted at the end of a rotating arm as illustrated in fig. 2.8. At the rotational axis lies a 'flex pivot' with a spring constant of 0.0016 Nm/deg. The rotational distance is determined by using a Linear Voltage Differential Transducer (LVDT) which is placed at the end of the arm. To determine how much of a linear displacement recorded by the LVDT correspond to thrust for a fixed spring constant flex pivot, a calibration curve was plotted using known weights.

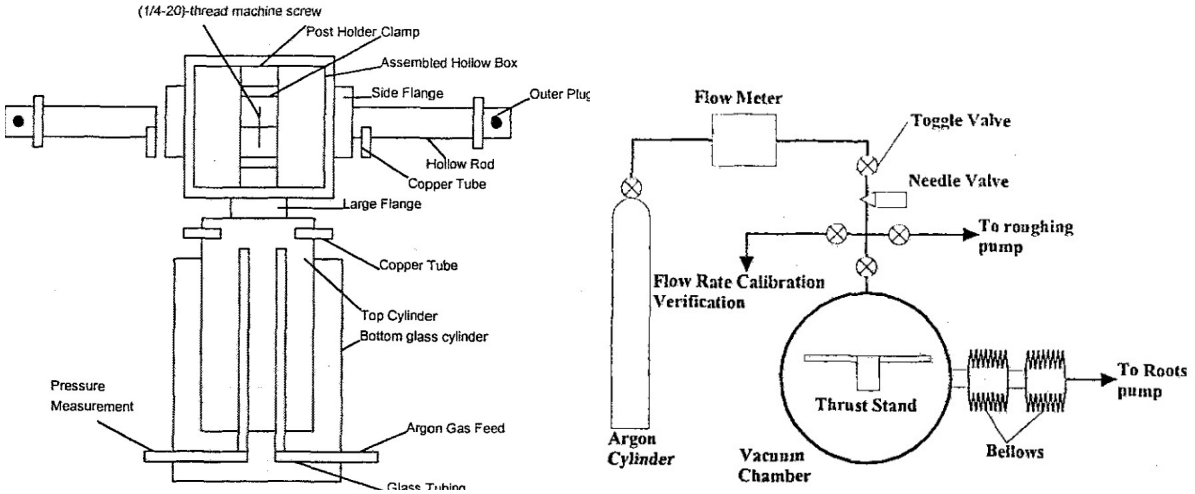


Figure 2.8: Thrust stand (left) and thrust setup (right) schematic. Sourced from Tew et al. 2000.

In Daduí C Guerrieri et al. 2017's work, the experimental setup, described in the work by Bijster 2014 and Jansen 2016, is employed. In this setup, the thruster is mounted at the end of a pendulum (see fig. 2.9). A torsion spring provides the restoring force for the pendulum. To detect the displacement of the thruster, a *capacitive displacement sensor* is used. The sensor head is kept static at some small distance from the pendulum arm (at neutral position). The target, which is a conductive material and has the same size as the sensor head, is bolted onto the pendulum arm. To correlate the thruster displacement to the thrust generated, a *varying turn-density coil* is used to simulate thrust. It repels or attracts a ferromagnetic target (bolted on the pendulum arm) when current passes through it. As the force is electromagnetically generated, the calibration can be done in vacuum as well, which is the major advantage of this system. The thrust bench range is 0.1 to 15 mN with an accuracy of 0.036 mN (Jansen 2016).

An image of the thrust stand placed in a vacuum chamber is given in fig. 2.10.

The experimental set-up employed by Daduí C Guerrieri et al. 2017 seems to be the most adequate for LPM thruster testing because the thrust bench allows for remote calibration in the vacuum chamber i.e. in the same ambient conditions in which the propulsion characterization is to take place. This is unlike Ketsdever et al. 2000 thrust bench in which the calibration is done using physical weights that needs to be attached by a string in the open.

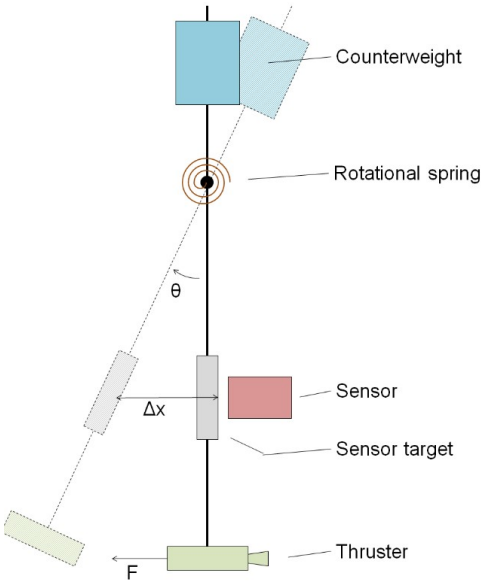


Figure 2.9: Schematic of the thrust stand. Sourced from Jansen 2016.

Figure 2.10: Picture of the thrust stand in the vacuum chamber with the varying turn-density coil highlighted in red. Sourced from Jansen 2016.

Table 2.1 lists out a reading from each of the experiments performed by the aforementioned

groups to give an idea about the typical values of an LPM propulsion system. $I_{SP,theoretical}$ was calculated using eq. 1.3.

Table 2.1: Values from Experimental testing of LPM system by three different groups.

	Ketsdever, Lee, and Lilly 2005	Daduí C Guerrieri et al. 2017	Palmer, Nguyen, and Thornell 2013
Propellant	Argon	Nitrogen	Nitrogen
Inlet Pressure (Pa)	112	250	548
Background Pressure (Pa)	0.00133	15	253.31
Heaterchip temperature (K)	575	422	300
I_{sp} (s)	51	26.44	53.5
$I_{SP,theoretical}$ (s)	63.121	62.716	52.879
Thrust (mN)	0.7	0.73	0.53

2.3. Conclusion

Micro-Heat Exchangers are compact and light-weight heat exchangers due to their higher convective heat transfer coefficient as compared to traditional shell and tube heat exchangers. These characteristics reduce structural support requirements and also save cost in fabrication. The fluid flow in most industrial applications is in the continuum regime. In PCB-integrated electronic cooling systems and LPM thrusters, the fluid flow is in the rarefied regime. For each application domain, several designs have been thought of and evaluated using simulations. The designs are often simple, such as rectangles and circles, with a few parameters, such as length and breadth, that are used to conduct parametric analyses. Since the conceptualization of the LPM thruster in the late 1990s, experimental methods have improved, more concepts for the overall design have been generated, and parametric analysis has been performed for the MHX design but only for rectangular and circular microchannel shapes. There may exit a microchannel shape, very different from standard shapes like polygons or circles, that result in the most energy-efficient performance under a particular set of input conditions. To find such a shape first requires the definition of a base geometry and a set of rules that determine how parametric values affect the overall geometry. Once the design is parameterized, an optimization algorithm can be used to find the most energy-efficient design. The solutions presented in literature to model rarefied fluid flow using modified versions of the Navier-Stokes or the Boltzmann equation are specific to particular geometries and thus, the DSMC must be used as its methodology is independent of the geometry and is well suited to evaluate varied designs. To save the computational expense of DSMC runs, a response surface can be created to give instantaneous results, though it would deviate from the DSMC result depending on its accuracy. In every application domain, the search for a shape that can lead to optimum performance continues and, with betterment in fabrication technologies, it may very well continue far into the future.

3

RSM: Method and Setup

3.1. Introduction

Optimization, as defined by Cambridge Dictionary, n.d., is "the act of making something as good as possible". The 'something', in this case, is the design of the heaterchip geometry where 'good' refers to a design that has high thrust, low power requirement and high specific impulse. An optimization problem has several input parameters influencing several output parameters. Some (or all) of the input parameters are variables and the rest are fixed parameters. In this study, the inlet pressure and temperature, heaterchip temperature and outflow conditions (vacuum) are the fixed parameters whereas the design of the slot is variable (more in chapter 4). The output parameters are Thrust, Specific Impulse (I_{sp}) and power requirement.

The aim of the optimization [1] can be written as,

$$\underset{x}{minimize} f(x), x \in X \subseteq R^n \tag{3.1}$$

where, x is the set of input variables which belongs to the design space, X, which in turn is a subset of an n-dimensional space which is also called a *hypercube*. f is the function whose minima needs to be found. An optimization method starts with an initial guess of x, typically provided by the user. The method then chooses the x^{s+1} set based on information from the previous s iterations. The history of iterations can be represented as the following -

$$y_r = f(x_r) \tag{3.2}$$

where r is the iteration number, $r = 1, 2, 3...s$.

Learning from history, the optimization method ideally makes better and better prediction every iteration until, eventually, the quality of solutions starts to stagnate. This point of

1. Optimization is *often* described as a problem of finding the minimum of a function

stagnation may very well be a locally optimum solution whereas the point one wants is the global optimum, which can be quite difficult depending on the problem.

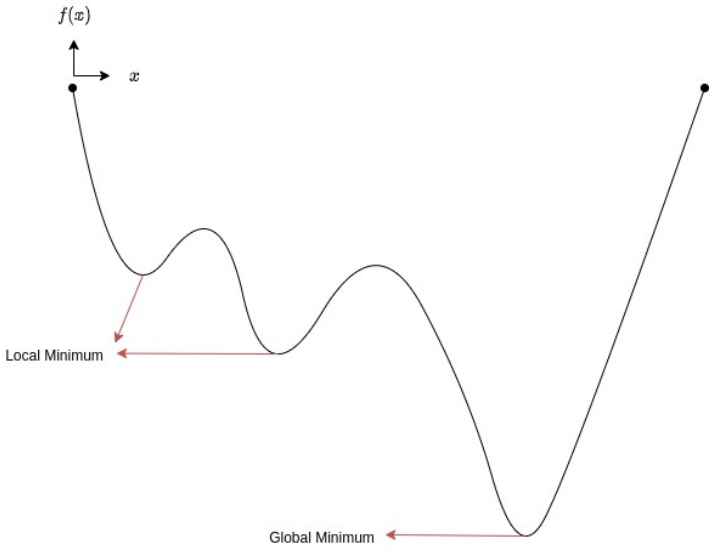


Figure 3.1: Illustration of the difference between local and global optimum. In this case the optimum was defined as the minimum of $f(x)$.

Sometimes the link between the input and the output can be expressed analytically, $y_{output\ parameters} = f(x_{input\ parameters})$. However, in the case of an LPM system, $f()$ is not known. There are no analytical functions that link the design of an heaterchip with the performance except for a design with rectangular and circular slots (Daduí C. Guerrieri et al. 2018). Even then, Daduí C. Guerrieri et al. 2018's analytic functions are valid only if certain geometrical constraints are satisfied. Experiments [2] can be performed to determine $output\ parameters$ from $input\ parameters$ but during the optimization process repeated evaluations (eq. 3.2) using experiments is not practical. The solution is to use an approximate function, also known as a surrogate function, meta-model or response surface, that is created by observing multiple input-output combinations and guessing their relation represented as an analytical function, evaluating which is much faster than experimental evaluation. This method is known as Response Surface Methodology (RSM) the steps for which are covered in the following sections.

3.2. Design of Experiment (DOE)

The purpose of a DoE run is to determine the samples (input variables) for which experiments must be performed. A dataset containing the samples and the experiment results are then used to build a high fidelity response surface. A good DoE method extracts a high amount of information from the low number of samples. To do so, the samples must be uniformly spread over the design space.

2. The term experiments also include computer simulations.

Random number generators fall into 3 categories (James 1990):

1. Truly random numbers: These numbers are not generated by an algorithm but rather produced by some random physical process such as radioactive decay (which exploits the randomness of a Geiger–Müller tube registering an electron from a ^{90}Sr source (Schmidt 1970)), chaotic lasers or atmospheric noise (see random.org, n.d.).
2. Pseudorandom numbers: A numerical algorithm is used to generate numbers that are not truly random but appear to be.
3. Quasirandom numbers: The objective of the numerical algorithm used to generate quasirandom numbers is to generate numbers that are distributed as uniformly as possible in an n-dimensional space.

Truly random numbers and pseudorandom numbers (which try to emulate truly random numbers) are not efficient in sampling the design space uniformly as there is no guarantee that the samples will not cluster together thereby leaving large areas of the design space unexplored. For the same number of samples, a quasirandom generator guarantees that the points will be uniformly distributed over the design space. Quasirandom numbers are not random. For example, consider the interval $(0, 1)$ and N number of trials. A uniform random generator may, purely coincidentally, generate N numbers that only fill the first half of the interval. It is because the "next" point does not have information about the previous points. A quasirandom generator, however, keeps track of the previous points and will ensure that all N generated numbers span the entirety of the interval. The measure of irregularity in the distribution of samples in the design space is known as *discrepancy* which is calculated using eq. 3.3 (Niederreiter 1988).

$$\begin{aligned} D(J;N) &= A(J;N) - V(J)N \\ \Delta(N) &= \sup_J |D(J;N)| \end{aligned} \tag{3.3}$$

Consider an s-dimensional unit cube ($s \geq 1$), $I^s = (0,1)^s$, of which J is a subset of. Let $x_1, x_2, \cdots, x_N$ be the N points generated by a random number generator. $V(J)$ is the volume of J, $A(J;N)$ is the number of generated points $(x_1, x_2, \cdots, x_N)$ that fall under J with the requirement that J contains at least one of the generated points. $\Delta(N)$ is the *discrepancy* of the generated numbers.

A quasi-random number generator like Van Der Corput, Halton, Faure and Sobol sequences are ideal for performing a DoE run as they generate low discrepancy sequences. Of these, the Sobol algorithm generates the lowest discrepancy sequences which essentially means the points are evenly distributed. A high discrepancy sequence set would have some points that are clustered and some points that are far away. The work by Bratley and Fox 1988 delineates a faster-than-original Sobol algorithm. Fig. 3.2 visually demonstrates the sample distribution generated by a uniform random generator (using the Python library – numpy, n.d.) and Sobol sequences (using the Python library – TensorFlow, n.d.) for 2 parameters. Notice large gaps

near the middle right, and clustering of points just below it in the case of a uniform random generator.

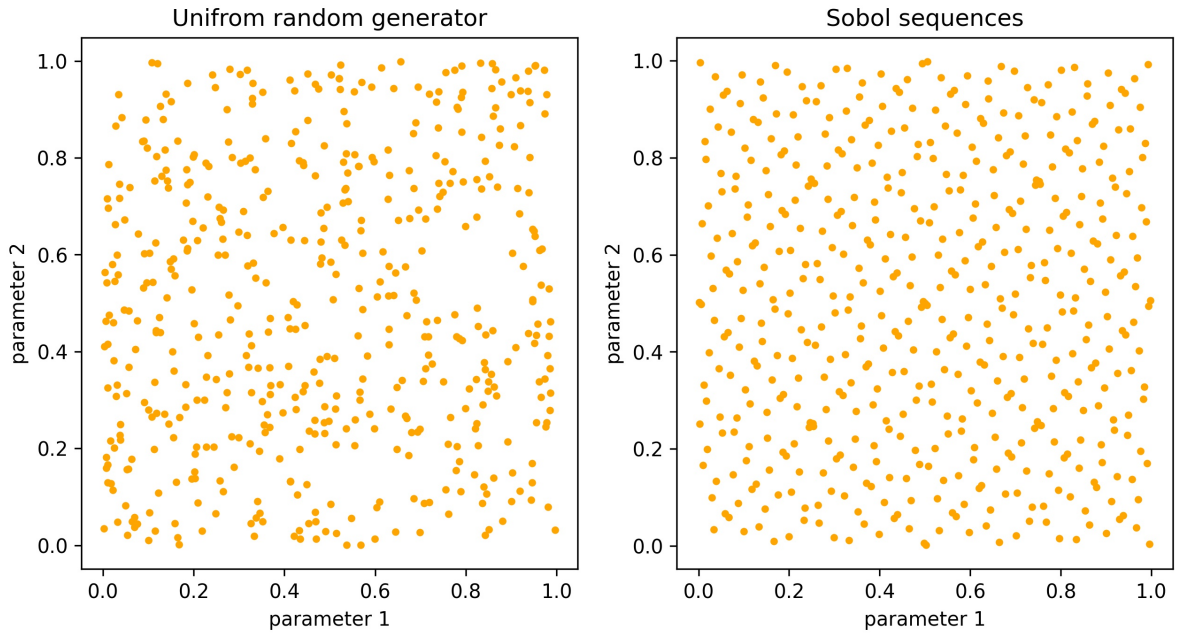


Figure 3.2: Visual demonstration of sample distribution in a 2 dimensional space for 512 samples generated by a uniform random generator (left) and Sobol sequences (right).

3.3. Experimental Evaluation

As mentioned in the Introduction, traditional CFD codes, which solve the Navier-Stokes equation, cannot be used to simulate the performance of an LPM system as the Knudsen number (Kn) of the fluid is too high for the Navier-Stokes equation. Rather the Direct Simulation Monte Carlo (DSMC) method is more suited to the task, which has been mathematically proven to converge, for a limiting case of infinite particles, on the solution of the Boltzmann equation (Wagner 1992). Fig. 3.3 shows the Kn range at which a mathematical model is valid.

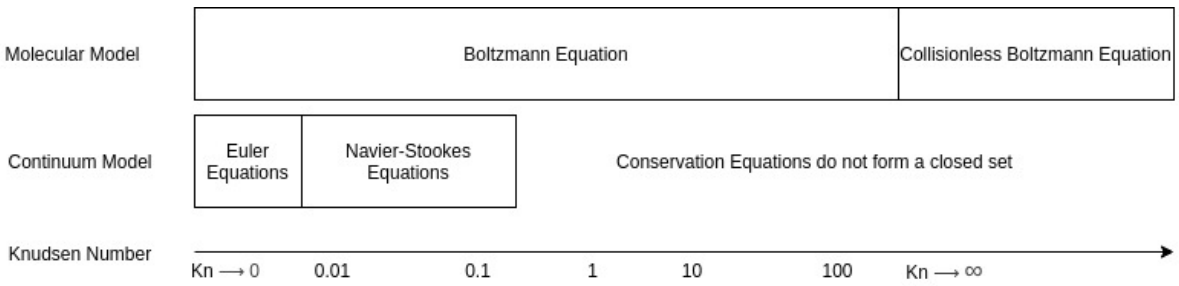


Figure 3.3: Validity of mathematical models with respect to the Knudsen number (not to scale). Image reconstructed from G. A. Bird 1994b.

The Boltzmann equation for a single-species, monoatomic, non-reacting gas is (White et al. 2018):

$$\begin{aligned} \frac{\partial(nf)}{\partial t} + \mathbf{c}\frac{\partial(nf)}{\partial \mathbf{r}} + \mathbf{F}\frac{\partial(nf)}{\partial \mathbf{c}} &= J\left(f, f^*\right) \\ J\left(f, f^*\right) &= \int_{-\infty}^{\infty}\int_{0}^{4\pi} n^2\left(f^* f_1^* - f f_1\right) c_r \sigma_T d\Omega d\mathbf{c}_1 \end{aligned} \tag{3.4}$$

where nf is he product of number density and the normalized molecular velocity distribution, r and c is the position and velocity vectors of a molecule, **F** is external force per unit mass, and the integral, $J\left(f, f^*\right)$, describes the collision process. * represents the post-collision state, f and f_1 are the velocity distribution function at c and c_1 respectively, c_r is the relative speed of two colliding particles, σ_T is the molecular cross-section, σ_T is the solid collision angle and $d\mathbf{c}_1$ is an element of the velocity space.

The problem in evaluating eq. 3.4 lies in the sheer computational effort needed to solve the integral. According to Rieffel 1995, solving the integral using the quadrature formula would require 3,200,000 evaluations at each grid point (for a meshed geometry), for each possible velocity at each timestep.

The Direct Simulation Monte Carlo (DSMC) method was developed by G.A. Bird in the 1960s over the course of a few works which are mentioned in G. A. Bird 1970. The method has been documented in his book, G. A. Bird 1994b. In this method, the kinematics of individual molecules are directly modelled using Newtonian mechanics. To ease computational demands, multiple molecules are represented by a single simulated molecule. Computational demands are further eased by simulating the collisions between a pair of particles for only the pairs that have a high probability of collision. The collisions in a low-pressure high-Kn environment can be assumed to be binary in nature i.e. when a molecule suffers a collision, it does so with only one other molecule at a time, un-influenced by other molecules in the system. This assumption is valid when the intermolecular distances are much larger than the diameter of the molecule (definition of a *dilute gas* (pg. 7 G. A. Bird 1994b)). Several models exist in the literature for simulating binary elastic collisions that occur between mono-atomic molecules (such as the Inverse Power Law (IPL), Hard Sphere (HS), Variable Hard Sphere (VHS), Variable Soft Sphere (VSS), and Generalized Hard Sphere model (chapter 2 G. A. Bird 1994b)) and binary inelastic collisions that, typically, occur between di-atomic and poly-atomic molecules, which have rotational and vibrational degrees of freedom (chapter 5 G. A. Bird 1994b)).

The following solvers were studied to understand the key similarities and differences between them: HAWK (Rieffel 1995, MONACO (Dietrich and Boyd 1996), SMILE (Ivanov, Markelov, and Gimelshein 1998), DAC (LeBeau and Lumpkin III 2001), SUPREM-DSMC (Wadsworth et al. 2001), DS2V/3V (G. A. Bird 2005, SMILE++ (Ivanov et al. 2011), MGDS (Gao, Zhang, and Schwartzentruber 2011), SUGAR (Korkut and Levin 2014), Voldipar (Turansky and Argrow 2015), NFS-DSMC (Kumar and Chinnappan 2017), DgSMC-A (Kargaran, Minuchehr, and Zolfaghari 2017) and dsmcFoam+ (White et al. 2018).

The DSMC solvers were found to differ on the following fronts: Collision pair selection, Elastic

collision model, Energy-Exchange/Inelastic collision model, Fluid-surface interaction model, mesh generation, particle tracking and Load-balancing (fig. 3.4). All the solvers mentioned in this section are parallelized i.e. a group of spatially close molecules (such as molecules in a grid or a group of grids (known as a 'partition')) are moved between computer processors to ensure equal load among the processors. This technique is called *Dynamic load balancing*. In the HAWK solver, a partition is defined as a group of spatially close grid cells, as illustrated in fig. 3.5, which is determined by a partitioning algorithm. An issue with parallelization is that, with the increase in the number of processors, the time in inter-processor communication also increases. This is known as 'communication overhead'. The ratio of computation to communication time is called *Granularity*. A user can change the granularity by varying the number of cells per partition. The solvers mostly feature the same aforementioned elastic and inelastic collision models. The popular choice of fluid-surface interaction model is the *Maxwell* model.

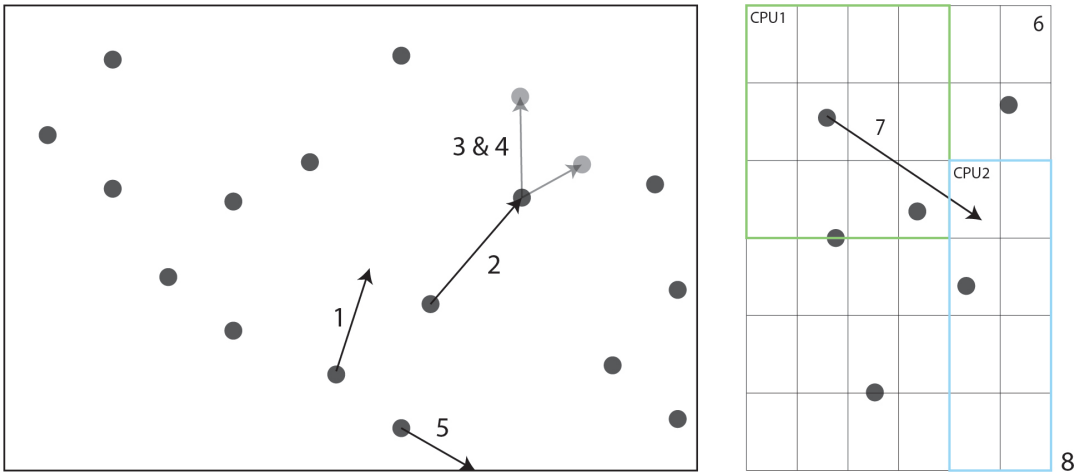


Figure 3.4: Main aspects of DSMC solvers: 1) Newtonian mechanics 2) Collision pair selection model 3) Elastic collision model 4) Inelastic/Energy-Exchange model 5) Fluid-Surface interaction model 6) Grid generation model 7) Particle tracking algorithm 8) Load balancing algorithm.

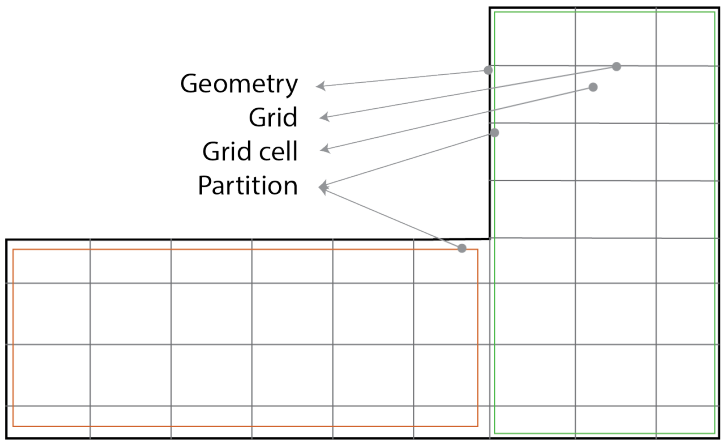


Figure 3.5: Distinction between Geometry, Grid and Partition. Image reproduced from Rieffel 1995.

A pre-processing step of the simulation involves the discretisation of the geometry into grid cells as shown in fig. 3.5. The grid-cell size or cell size affects the collision pair selection and the collision model and hence the mesh must be appropriately generated. According to Zhang 2010, to obtain accurate results, the cell size should be in the order of the mean free path (λ) and the time-step (Δt) should be in the order of the mean collision time (τ). Zhang 2010 states that, in practice, the cell size is generally set as, $h = \lambda/2$ and the time-step is set as, $\Delta t = \tau/5$. An issue with the grid or mesh is that the local mean free path and the local mean collision time requires the flowfield information but the mesh itself affects the flowfield. Due to this cyclic relation, it requires some iterations to get the mesh right. The NFS, SMILE, DgSMC-A, SUGAR and MGDS solvers employ the 'adaptive mesh refinement' method in which the mesh automatically adjusts itself according to the flowfield.

Within a simulation time-step, Δt, it is possible that a molecule may exit the cell it is residing in and enter another cell. In this case, the remaining time that the molecule will travel along the same velocity vector without collision needs to be written to the particle list of the adjacent cell the molecule is travelling to. The transfer of particle data (position, velocity, remaining time of flight and molecular properties) is typically carried out at the end of the simulation time step to reduce communications overhead. The Hawk code employs the *ray tracing* method to determine the exact point on the cell boundary where the particle will pass through. Determining the cell face through which the particle will pass through and the exact point of passing through that cell face is computationally expensive though it was preferred by Rieffel 1995 because the underlying mathematics is simple which makes the implementation easier. In the MONACO solver, the particles are moved by the entire time-step and then a simple check determines if the particle is in the same cell or not. Generally, only a small subset of particles leave the cell in a time-step and only for those particles the exact point of passage and remaining time of flight in the other cell is calculated.

To reduce steady-state time, the SMILE solver first simulates a small set of molecules until the system achieves steady state. Then the particles are cloned (hence doubling the particle count) and allowed to reach steady state. The process continues until the correct number of particles are in the simulation. The authors observed that this method decreased the time it takes for the system to achieve steady state.

In this project, the DSMC solver, VSTRAP, by SPARC Industries ("SPARC Industries SARL," n.d.) has been used. Some of its features are now mentioned. To avoid the problems associated with mesh generation, a gridless approach has been implemented in VSTRAP. In this method, particles are grouped together with their nearest neighbours instead of being grouped according to which grid cell they occupy (Olson and Christlieb 2008). This alleviates the need to generate a 3D mesh. A surface mesh does need to be generated to mark the physical boundaries. Pairs for collision are selected according to the 'Natural sample size' method which has been described in Baganoff and McDonald 1990. The elastic collisions are modelled using the VHS method in which the collision cross-section is treated as variable; it is inversely proportional to the relative

velocity between 2 colliding molecules. The reasoning can be related to the decrease in the coefficient of viscosity of a fluid with an increase in temperature (chapter 2.6 G. A. Bird 1994b). The fluid-surface interaction model is the Maxwell model, described in the subsequent section.

3.3.1. Fluid - Surface Interaction models

Selection of a Fluid - Surface Interaction model in the DSMC simulation of an LPM system is important as it controls the amount of energy extracted by the fluid from the heated surface as well as the direction of reflection of a fluid particle after it strikes a surface. Ketsdever, Wadsworth, and Muntz 2001 found that in the range of the energy accommodation coefficient, from 0 to 1, specific impulse changed by $\approx 20\%$. In between Maxwell and Cercignani-Lampis model, the maximum difference in specific impulse, for the same energy accommodation coefficient, was found to be 5%.

Gas molecules incident on a surface generally reflects off surfaces in three ways as illustrated by fig. 3.6. As with the scattering of light, the direction of scatter depends on the surface. Highly polished surfaces will lead to specular scattering and rough surfaces will tend to reflect particles in a diffusive manner. In addition, gas molecules exchange energy with the surface atoms which affect the magnitude of the reflected velocity. This effect is expressed with the help of 'accommodation coefficients' which are generally user inputs in gas-surface scattering models.

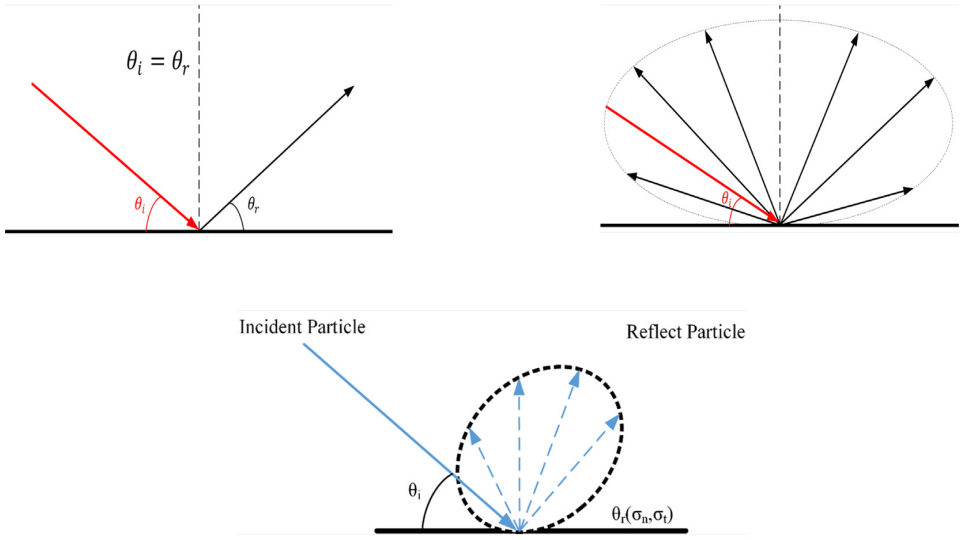


Figure 3.6: Gas scattering (clockwise): Specular scattering, Diffusive scattering and partially-diffusive scattering. Image sourced from Yousefi-Nasab et al. 2019.

Accommodation coefficient

Accommodation coefficient is a quantity that is used to express the efficiency of energy or momentum transfer between gas and surface boundary. The general expression is given in eq.

3.5.

$$\alpha = \frac{\varphi_i - \varphi_r}{\varphi_i - \varphi_w} \tag{3.5}$$

where, φ_i is the magnitude (energy or momentum) of incident flux, and φ_r is the magnitude of reflected flux. φ_w is the magnitude of the reflected flux had the particle attained thermal equilibrium with the surface. Tab. 3.1 lists out the energy and momentum accommodation coefficient formulas in the tangential and normal directions.

Table 3.1: Accommodation coefficients (Tengfei 2015).

	Tangential	Normal
Energy	$\alpha_t = \frac{\langle {v_t}^2 \rangle_{in} - \langle {v_t}^2 \rangle_{re}}{\langle {v_t}^2 \rangle_{in} - kT_{surface}/m_g}$	$\alpha_n = \frac{\langle {v_n}^2 \rangle_{in} - \langle {v_n}^2 \rangle_{re}}{\langle {v_n}^2 \rangle_{in} - 2kT_{surface}/m_g}$
Momentum	$\sigma_t = \frac{\langle v_t \rangle_{in} - \langle v_t \rangle_{re}}{\langle v_t \rangle_{in} - V_{t,surface}}$	$\sigma_n = \frac{\langle v_n \rangle_{in} - \langle v_n \rangle_{re}}{\langle v_n \rangle_{in} - \sqrt{\pi kT_{surface}/2m_g}}$

In Tab. 3.1, $\langle\rangle$ is the average operation, *t* and *n* represents tangential and normal component of velocity respectively, *in* and *re* indicates incident and reflected velocity respectively, v is the velocity vector, $V_{t,surface}$ is the tangential velocity of the wall (0 if wall is stationary), m_g is the mass of the gas molecule, $T_{surface}$ is the temperature of the surface and k is Boltzmann constant.

In the energy accommodation coefficient calculation for the tangential and normal component, φ_w in eq. 3.5 was replaced by $\frac{1}{2}kT_{surface}$ and $kT_{surface}$ respectively [3]. This is because when the gas molecule is fully accommodated with the surface, the velocity distribution is Maxwellian in nature. According to the equipartition theorem, particles have an average energy of $\frac{1}{2}k_BT$ per degree of freedom. A tangential component comprises of 2 degrees of freedom whereas the normal component comprises of 1 degree of freedom. The accommodation coefficient has the following effects on a fluid flow that is constrained with walls hotter than the fluid. If the energy accommodation coefficient is high then there is increased heat transfer from the wall surface to the fluid. If the tangential momentum accommodation coefficient is high then the particle would reflect in a diffusive manner without any change in velocity. If the normal accommodation coefficient is high then the velocity of the reflected particles will be more than their incident velocity.

Model dependent relations between accommodation coefficients: The popular scattering models assumes some relation between the accommodation coefficients. This is tabulated in tab. 3.2.

3. Note that for the equations in row 1 tab. 3.1, multiplication of both numerator and denominator by $\frac{1}{2}m_g$ would yield the initial form of the equation corresponding to the definition set in eq. 3.5

Table 3.2: Relations between accommodation coefficients corresponding to the scattering model.

Model	**Relation**
Specular	$\alpha_t = \alpha_n = \sigma_t = \sigma_n = 0$
Maxwell	$\alpha = \sigma$
CL	$\alpha_t = \sigma_t(2 - \sigma_t)$ and $\alpha_n = \sigma_n$

In Tab. 3.2, for the Maxwell model, α and σ are mentioned without any subscript. In this case the full velocity magnitude is considered for calculating the accommodation coefficient. α or the energy accommodation coefficient is (Tengfei 2015),

$$\alpha = \frac{\langle v^2 \rangle_{in} - \langle v^2 \rangle_{re}}{\langle v^2 \rangle_{in} - 4kT_{surface}/m_g} \tag{3.6}$$

where v is the velocity magnitude, *in* and *re* represent incident and reflected velocity vectors, $T_{surface}$ is the surface temperature, m_g is the mass of the gas molecule and *k* is the Boltzmann constant.

Model overview

As of yet, the Maxwell and the CL (Cercignani and Lampis 1971) models seem to be the best contenders for being used in the simulations of LPM systems. The Maxwell model is popularly employed in DSMC solvers including VSTRAP. Andric, Meyer, and Jenny 2019 state that the model has successfully modelled the scattering of monoatomic gases but, in general, it is considered a simplistic model because of its base assumption (linear combination of specular and diffusive reflections) and its inability to reproduce the scatter pattern made by reflected particles in a molecular beam experiment. The CL model can reproduce such a pattern and might become available in the majority of DSMC solvers in the future.

The Maxwell model is a simplistic model that assumes that some particles (proportional to accommodation coefficient) reflect diffusively while the rest reflect specularly. The model has been successfully used for modelling the scattering of monoatomic particles but it is unable to reproduce the experimentally (molecular beam experiments) observed lobular patterns (Andric, Meyer, and Jenny 2019).

The probability kernel is given by Peddakotla, Kammara, and Kumar 2019,

$$P_{\mathrm{M}}\left(V' \to V\right) = (1-\alpha)\delta\left(V' - V_{\text{specular}}\right) + \alpha f_{\mathrm{M}}(V)|V \cdot \mathbf{n}| \tag{3.7}$$

where V' and V are the incident and reflected velocity vectors, α is the momentum accommodation coefficient, n is the surface normal vector, and

$$f_M(V) = \sqrt[3]{\frac{M_g}{2\pi k_B T_s}} \exp\left(\frac{M_g V^2}{2k_B T_s}\right)$$

where M_g is the molecular mass of the gas, T_s is surface temperature, and k_B is the Boltzmann constant.

Some energy accommodation coefficients, α, for the Maxwell model relevant to LPM systems are: 0.91 (Argon on Silicon), 0.82 (Nitrogen on Silicon) and, 0.86 (Water on Silicon). References: Trott et al. 2011 and Kinefuchi et al. 2017. These values were determined experimentally.

Some other fluid-surface interaction models are:

- Specular model: This model is not applicable for simulating an LPM system as it would mean that no energy exchange takes place between the surface and the fluid. This model should be used when one or more of the following conditions are satisfied (G A Bird 1994a):
  - A smooth metal surface that has been out-gassed through exposure to high vacuum and high temperatures.
  - The ratio of the molecular weight of the gas to that of the surface molecules is small in comparison with unity.
  - The translational energy of the molecules relative to the surface is larger than several electron volts.

  Additionally, the specular model should be used for: heated wires with thoroughly outgassed surfaces with static gases, supersonic flow in highly rarefied medium, and for carbon surface in space with atomic oxygen (G A Bird 1994a).
- The molecular dynamics method (Finger 2005): The model is accurate when surface roughness is taken into account, but it is computationally expensive.
- The LLY model (Liang, Li, and Ye 2018): A relatively new model that would require more testing in order to be considered for implementation. Liao et al. 2018 states that the corrugation parameter and the potential well depth coefficients used in the LLY model are oversimplifying the physics and the accuracy of the collision simulation may not be valid for the whole velocity range.

3.3.2. Simulation setup

In this study, an inlet pressure of 50 Pa, an inlet temperature of 300 K and heaterchip temperature of 600 K with Argon as the propellant are the fixed input parameters. The drift velocity of particles is 0 and their velocities are initialized with a Maxwell–Boltzmann distribution at the inlet temperature. The response surface that would be built and the subsequent optimization process will yield a design that is optimum for this particular set of fixed input parameters only. The input parameters were chosen for the following reasons – In Daduí C Guerrieri et al. 2016, the plenum is at 50 Pa and 300 K. They considered a heaterchip temperature of 300 K and 900 K. As the focus of the thesis is on the design of the microchannel rather than the

influence of boundary condition on the performance of the thruster, a similar set of boundary conditions was considered for the simulations carried out in this thesis. Additionally, increasing inlet pressure increased the number of simulated molecules in the system that increased the computational load per simulation and thus the inlet pressure of 50 Pa was deemed appropriate. Argon was chosen as the propellant because, at the time of writing, VSTRAP can only model monoatomic molecules and, additionally, the collision cross-section data file for Argon was readily available for VSTRAP. It is possible to treat the fixed input parameters as variables (in contrast to fixed parameters) but has not been done so in this study and is left for future studies. Table 3.3 was used to set up the simulation. The underlined quantities were explicitly written in the input file for the simulation.

Physical surface name	**Property**	**Value**	
outlet-slot-interface inlet-slot-interface slot-walls	Maxwell -scattering Acc. Coeff. 0.91	600 K	
inlet	Inlet boundary (number density and temperature specified by user)	Propellant Pressure$_{inlet}$ (Pa) Temperature$_{inlet}$ (K) Number density (molecules/m^3) Total cross section (m^2) Mean Free Path (m) Mean velocity (m/s) Argon molecular mass (kg) Particle weight < Δt (s) <	Argon 50 300 $1.21E$+22 $3.31E$-19 $2.51E$-4 432.8 $6.63E$-26 $1.90E$+10 $1.9E$-7
outlet outlet-walls	Vacuum		
inlet-walls	Specular surface		

Table 3.3: Simulation setup.

Note: Particle weight was taken $7E$+9 and Δt was taken as $7E$-8 seconds. These values were chosen based on practical concerns. Decreasing particle weight increases the number of simulated particles in the system and thus increases computational load. Decreasing Δt delays the time it takes for the system to achieve steady-state. However, high particle weight results in the data having a low signal-to-noise ratio and high Δt results in a particle getting stuck in reflections between 2 points in the geometry. Thus a reasonable value was chosen between their lower bounds – Particle weight = 1 and $\Delta t \to 0$ – and their upper bounds (see tab. 3.3) based on the computational resources available. Steady-state was obtained after 2000 iterations, by which point of time there were ≈100000 simulated particles in the simulation domain. Generally, steady-state criterion is set by the DSMC user and for this thesis, it was

said that the simulation has achieved steady state if the fluctuation of the number of particles in the simulation domain reduced to less than 0.02% (illustrated in fig. 3.7).

The following equations were used to calculate quantities in Tab. 3.3:

1. Number density (molecules/m^3):

$$\frac{\text{Pressure}_{inlet} \times 6.0221 \times 10^{23}}{\text{Temperature}_{inlet} \times 8.3144}$$

2. Mean Free Path (m):

$$\lambda = \frac{1}{\textit{Total cross section} \times \textit{Number density}}$$

3. Mean velocity (m/s):

$$\sqrt{\frac{3k_B \text{Temperature}_{inlet}}{mass_{Argon}}}$$

4. Particle weight (Number of real/Number of simulated particles):

$$\frac{\lambda^3 \times \text{Number density}}{10}$$

5. Time step:

$$\Delta t = \frac{\lambda}{3 \times \text{Mean velocity}}$$

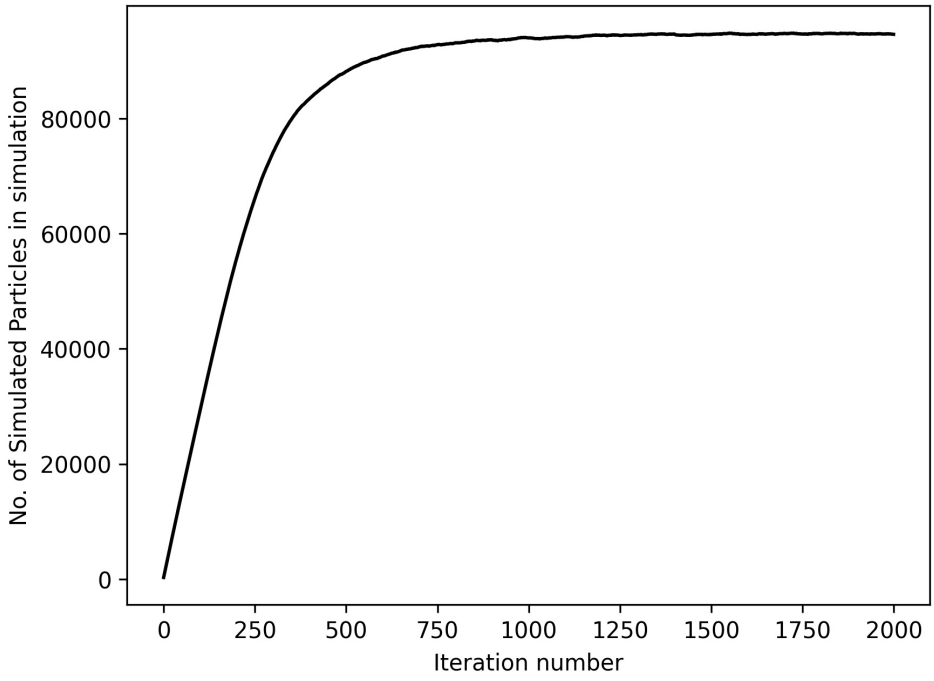


Figure 3.7: Number of simulated particles in the simulation domain per time-step.

3.4. Response Surface Modelling

The response surface is an analytical function (which guarantees almost instantaneous evaluation) that links the input variable and the output parameter which, for this project, is thrust, I_{sp} and power required. The coefficients of the analytical function are determined by fitting the data. The data is a set of inputs from the DoE run and its corresponding outputs from experiments.

The thesis aims to explore as many designs as possible (like the generated images on the cover page). Suffice to say, polygons and conic sections make up a tiny proportion of all possible geometries that the slot on a heaterchip can take on. The design parameters that are varied to generate a closed Non-Uniform Rational Basis Spline (NURBS) are discussed in the next chapter. The parameters ($p_i \in (0, 1)$) control the position of control points in the XY-plane. From the position of control points, and from specifying the model parameters in the NURBS formulation, a closed curve is generated. The curve is the perimeter of the slot, which is a through-and-through cut into the heaterchip for fluid to pass through. As a simple array of input variables ($p_1, p_2, \ldots, p_n$) undergo various processing steps, it seemed like a good idea to link, using a response surface, the final image of the curve to the output rather than the input variables to the output.

The idea spawned from the work of Guo, Li, and Iorio 2016 who used a Convolutional Neural Network (CNN) with a deconvolution layer to map a 2D/3D geometry to a detailed steady-state flow field. For representing the geometry and the space near it, the authors used the Signed Distance Function (SDF) which is illustrated in fig. 3.8. The formulation given in eq. 3.8 is that of a *distance function*.

$$d(\vec{x}) = \min(|\vec{x} - \vec{x}_I|) \, \forall \vec{x}_I \in \partial\Omega \tag{3.8}$$

where $d(\vec{x})$ is the SDF and $\partial\Omega$ is the boundary. As $\vec{x}_I$ represent all points on the boundary, if $\vec{x}$ lies on the boundary then $d(\vec{x}) = 0$.

The difference between a distance function and SDF is minor. In an SDF the points outside a boundary are taken as negative and the points inside the boundary are taken as positive. In this project, the points outside the boundary are taken as 0 (see fig. 3.9). This is because, unlike an airfoil, the fluid is flowing through the slot in a heaterchip and the points outside the boundary are irrelevant to the flow. SDF is more effective as an input to a neural network than a simple boundary representation as it packs more information about the object and its surroundings. The effectiveness of SDF is intuitive as it makes sense that fluid flow will be more affected in regions close to the object than regions further away from the object; using SDF helps the neural network to understand that.

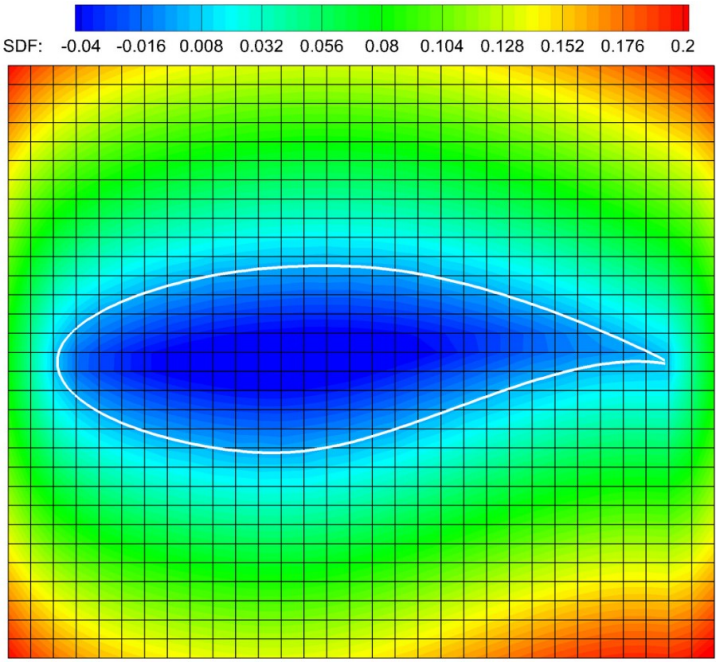


Figure 3.8: A SDF representation of airfoil RAE2822. Image sourced from Hui et al. 2020.

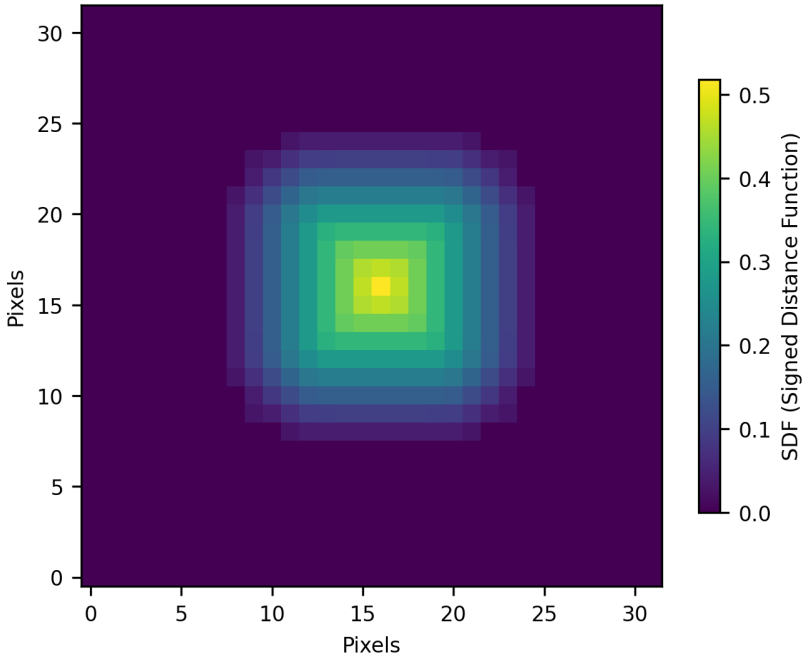
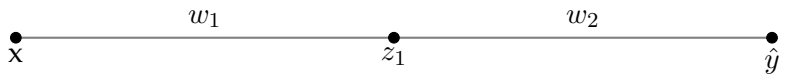


Figure 3.9: A modified SDF representation a 32 x 32 px square with rounded corners. The value of an interior pixel is the closest it is from a boundary pixel divided by 16. Points at and outside the boundary are assigned 0.

The working of a CNN is now explained in the context of this project [4]. The input is expected to be a 2D matrix with pixel values determined using the SDF. Each SDF image is linked to a

4. For a more in-depth explanation of Deep Neural Networks, refer Aggarwal 2018

thrust, power and I_{sp} value that had been determined using the DSMC method. First, consider a 1 hidden-layer neuron neural network to get an idea of the basic working principle of neural networks.



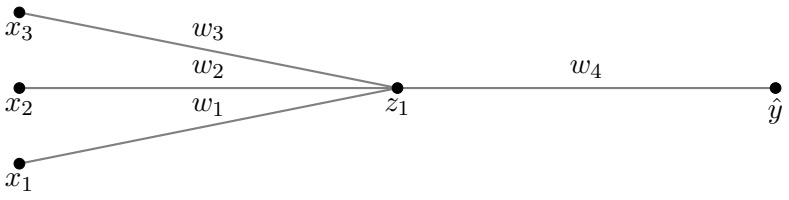
x is the input, $z_1 = \sigma(xw_1 + b_1)$ and $\hat{y} = \sigma(z_1 w_2 + b_2)$. w_i are called weights and b_i are called biases. The values of w_i and b_i are determined in the course of training the neural network, which is essentially solving the equation — $\hat{y} = \sigma(z_1 w_2 + b_2)$ — by solving for w_i and b_i. σ is a non-linear function that must be differentiable. Let y be the actual value (also known as *ground truth*) for the input, x. For a user specified σ, the value of $\hat{y}$ depends on w_1, w_2, b_1, b_2. Their values can be, initially, randomly assigned which would give a certain value for $\hat{y}$. The loss function $L(x, w_i, b_i)$, whether it be absolute error or mean squared error, determines how far $\hat{y}$ is from the ground truth, y. Depending on $L(x, w_i, b_i)$, the weights and bias values must be adjusted such that $\hat{y}$ is closer to y. This is similar to an optimization process where the goal is the minimization of error, $L(x, w_i, b_i)$. The gradient descent algorithm (eq. 3.9) is a popular tool for optimization.

$$W = W - \eta \frac{\partial L}{\partial W} \tag{3.9}$$

where W are the variables, η is the learning rate (too high could mean the solution never converges and too low would mean a long time for convergence) which is specified by the user but in some variations of the algorithm, the learning rate adapts depending on how large the gradient is or how fast the learning is happening. The term learning simply refers to the optimization of the loss function. The gradient, $\frac{\partial L}{\partial W}$, is calculated using a process known as *backpropagation* (eq. 3.10) which falls under the mathematical field of *Automatic Differentiation*. It deals with the evaluation of the derivative of a function using a computational algorithm. This is why the σ non-linear function must be differentiable, so as to not hinder the backpropagation process.

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_1} \frac{\partial z_1}{\partial w_1} \tag{3.10}$$

When multiple neurons are involved, the value of a neuron in the next layer can be written as:



$$z_1 = \sigma(b_1 + X^T W) \tag{3.11}$$

where, $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, $W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$ and b_1 is the bias.

Now, the idea of a CNN is to take an image, process it (using kernels/filters) and flatten the image into a 1D array of neurons after which the above-mentioned steps apply, as shown in fig. 3.10. Note that the kernel is a matrix of weights, the values of which are changed during the training process. Eq. 3.12 illustrates the convolution operation between an 8 x 8 (n_a x n_a) pixel image input and a 3 x 3 kernel (with all weight values = 1) with a 2 x 2 stride and bias value = 2. In a convolution operation (denoted as $\times$) the kernel moves over the image and the result is a dot product between the kernel and the part of the image the kernel is coinciding with. A 'stride' refers to the number of pixels the kernel skips as it moves over the image. Stride is used to shrink/downsample the input image to reduce the number of neurons (after flattening) and hence the number of trainable parameters. A 2 x 2 stride reduces the volume of the input by a factor of 4. Extra rows and columns with pixel value 0 are added to the input image depending on the padding requirements. In this example padding = 'same' which means that the number of padding rows = number of padding columns = $\lceil \frac{n_k}{s} \rceil - 1$, where n_k is the kernel length, s is the stride length, and $\lceil\ \rceil$ is the symbol for rounding up to an integer. In eq. 3.12, the number of padding rows and columns $= \lceil 3/2 \rceil - 1 = 1$. The row and column are added to the right and bottom of the matrix respectively. Had the padding number been more, the padding rows would have been added on both left and right of the matrix (split equally) and the padding columns would have been added on both top and bottom of the matrix (split equally). The other padding option is 'valid', in which no padding rows and columns are added. Using this option can lead to loss of information on the image borders after subsequent convolution operations. Additionally, the reduction in size of the image is greater in this padding option. This can be demonstrated as such: consider an input image of size 16 x 16 px and a kernel of size 3 x 3 with stride 2 x 2. Padding = 'same' results in an output image of 8 x 8 px calculated using the formula, $\lceil \frac{n_a}{s} \rceil = \lceil \frac{16}{2} \rceil = 8$. Padding = 'valid' results in an output image of, 7 x 7 px calculated using the formula, $\lfloor \frac{n_a - n_k}{s} \rfloor + 1 = \lfloor \frac{16-3}{2} \rfloor + 1 = 7$, where $\lfloor\ \rfloor$ is the symbol for rounding down to an integer. The first element of the result matrix is $1 \times 1 + 2 \times 1 + 3 \times 1 + 9 \times 1 + 10 \times 1 + 11 \times 1 + 17 \times 1 + 18 \times 1 + 19 \times 1 + 2(bias) = 92$. The 4th element of the result matrix is $7 \times 1 + 8 \times 1 + 0(padding) \times 1 + 15 \times 1 + 16 \times 1 + 0(padding) \times 1 + 23 \times 1 + 24 \times 1 + 0(padding) \times 1 + 2(bias) = 95$.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\ 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 \\ 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 \\ 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + 2 = \begin{bmatrix} 92 & 110 & 128 & 95 \\ 236 & 254 & 272 & 191 \\ 380 & 398 & 416 & 287 \\ 326 & 338 & 350 & 240 \end{bmatrix} \quad (3.12)$$

The convolution operation can be done using the TensorFlow python library (TensorFlow, n.d.) with the listing 3.1.

```
import tensorflow as tf
import numpy as np

x = np.array([   [1 ,2 ,3 ,4 ,5 ,6 , 7, 8],
                 [9 ,10,11,12,13,14,15,16],
                 [17,18,19,20,21,22,23,24],
                 [25,26,27,28,29,30,31,32],
                 [33,34,35,36,37,38,39,40],
                 [41,42,43,44,45,46,47,48],
                 [49,50,51,52,53,54,55,56],
                 [57,58,59,60,61,62,63,64]], dtype = 'float')

x = x.reshape((1,8,8,1))  # 1 mage x image dimension (8x8) x 1 channel
# for an RGB image the number of channels will be 3

initializer = tf.keras.initializers.Constant(1)  # sets value to 1
values = initializer(shape=(3, 3))  #  to display kernel matrix
print('Kernel matrix -->')
print(values)

print('Input X Kernel -->')  #  Convlutional operation
y = tf.keras.layers.Conv2D(1, (3,3), strides = (2,2), activation='relu', padding="
    same", bias_initializer=tf.keras.initializers.Constant(2), kernel_initializer=
    initializer)(x)
y = tf.reshape(y,(4,4))  # for display purpose
print(y)
```

Listing 3.1: Convolution demonstration using tensorflow.

Fig. 3.10 illustrates a small CNN architecture. The first kernel for the 8 x 8 input has the dimensions 32 x 5 x 5. Here 32 indicates the number of kernels. Convolution between the input and each of the 32 kernels result in 32 images of dimensions 4 x 4 (With padding = 'same' the output image size = input image size/stride). In the figure * represents 2 other operations that follow convolution: Max pooling (returns the maximum value of all elements of the input

over which the kernel is) and a non-linear activation function, Rectified Linear Unit (ReLU), which converts negative values to 0 and leaves the positive values as is. The 32 images are then stacked to form a 3D matrix of dimension 4 x 4 x 32. As the input is now 3D, the kernel also needs to be 3D i.e. 3 x 3 x 32. In the second layer, each of the 64 kernels has the dimension 3 x 3 x 32. After convolution, max pooling and ReLU, the input for the third layer has the dimension 2 x 2 x 64 which after the same set of process results in a 2 stack image of dimensions 2 x 2. The total number of elements are then 2 x 2 x 2 = 8. These 8 values are the input to the fully connected neural network which has 2 hidden layers with 4 neurons each. The number of parameters (weights and bias) to be trained is 19519 = 832 (kernel1:32x5x5 + bias:32) + 18496 (kernel2:64x3x3x32 + bias:64) + 130 (kernel3:2x1x1x64 + bias:2) + 36 (connections1:8x4 + bias:4) + 20 (connections2:4x4 + bias:4) + 5 (connections3:4x1 + bias:1).

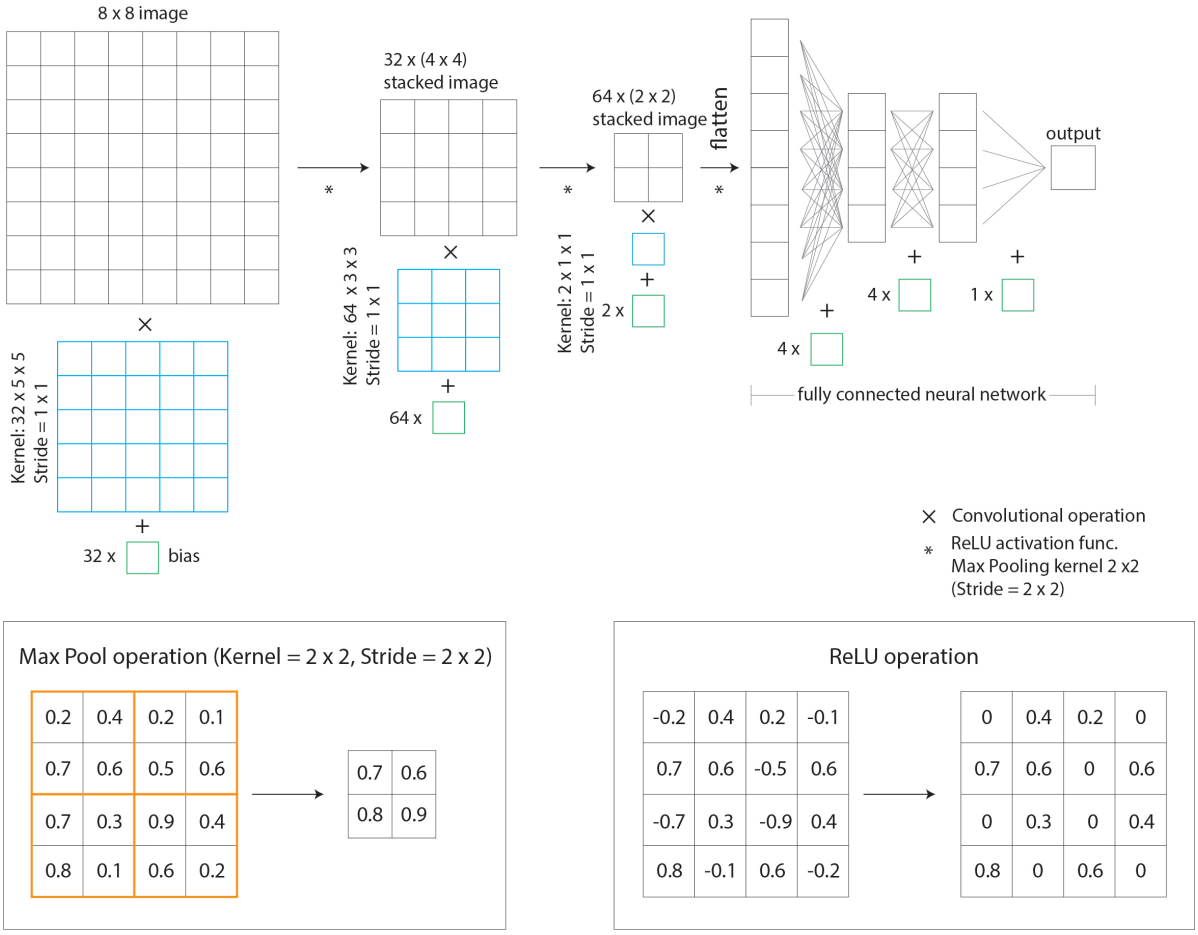


Figure 3.10: Example architecture of a CNN.

In TensorFlow code, fig. 3.10 translates to listing 3.2.

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.utils import plot_model

input_img = layers.Input(shape = (8, 8, 1))

# setting up convolutional layer
x = layers.Conv2D(32, (5,5), strides=(1,1), activation='relu', padding = 'same')(
    input_img)
x = layers.MaxPooling2D((2,2), strides=(1,1), padding = 'same')(x)
x = layers.Conv2D(64, (3,3), strides=(1,1), activation='relu', padding = 'same')(x
    )
x = layers.MaxPooling2D((2,2), strides=(2,2), padding = 'same')(x)
x = layers.Conv2D(2, (1,1), strides=(1,1), activation='relu', padding = 'same')(x)
x = layers.MaxPooling2D((2,2), strides=(2,2), padding = 'same')(x)
output = x

# setting up fully connected layers
output = layers.Flatten()(output)
out    = layers.Dense(4, activation='relu')(output)
out    = layers.Dense(4, activation='relu')(out)
out    = layers.Dense(1, activation='relu')(out)
model = models.Model(inputs = input_img, outputs = out)
model.summary()  # prints summary of setup
```

Listing 3.2: CNN architecture expressed in TensorFlow code.

Though CNN or neural networks, in general, may look like an empirical algorithm, it has a theoretical backing that stems from Hilbert's 13$^{\text{th}}$ problem that was presented in a lecture in 1900 (Charpentier, Lesne, and Nikolski 2007). The problem was titled: "Impossibility of the solution of the general equation of the 7$^{\text{th}}$ degree by means of functions of only two arguments". A general polynomial equation has the form:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0 \tag{3.13}$$

Using Tshirnhaus Transformations, the general equation of degree 5, 6 and 7 can be expressed as:

$$\begin{aligned} x^5 + ax + 1 &= 0 \\ x^6 + ax^2 + bx + 1 &= 0 \\ x^7 + ax^3 + bx^2 + cx + 1 &= 0 \end{aligned} \tag{3.14}$$

Notice that the 7-degree polynomial is being expressed as a function of 3 arguments instead of the desired 2. Arnol'd and Kolmogorov disproved Hilbert's statement, and eventually showed that not only a 7-degree polynomial but, rather, **every** continuous function can be written as

(Charpentier, Lesne, and Nikolski 2007):

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} g_q \left(\sum_{p=1}^{n} \varphi_{pq}(x_p) \right) \tag{3.15}$$

where $f : [0,1]^n \longrightarrow R$, $\varphi_{pq} : [0,1]^n \longrightarrow R$ and $g_q : R \longrightarrow R$. Essentially, a continuous function can be written as a combination of a non-linear function + linear algebra, which is the basic working principle of a neural network. Neural networks are often called 'universal approximators'. If the neural network is not accurate then the problem is not with the neural network concept but rather with the neural network architecture, training time or the absence of a deterministic relationship between the input and the output (Hornik, Stinchcombe, and White 1989).

Examples of work employing CNN as a response surface for engineering applications are now mentioned. Hui et al. 2020 used CNN and SDF geometry representation to make a surrogate model that can predict the pressure distribution for an arbitrary airfoil shape. 1500 airfoils were generated by introducing modifications to a base airfoil, RAE2822. Out of these 1500 airfoils, 2/3 were used for training the CNN and 1/3 were used for validation. Hui et al. 2020 also compared CNN to a *forward neural networks* and found that the CNN architecture took half the time to train and can predict the pressure distribution 3 orders of magnitude faster than a forward neural network. In Zhang, Sung, and Mavris 2018, the AeroCNN-I architecture (fig. 3.11) is interesting as it incorporates inflow and boundary conditions in addition to the image of the airfoil to predict the airfoil lift coefficient. The CNN part of the architecture is illustrated in fig. 3.12.

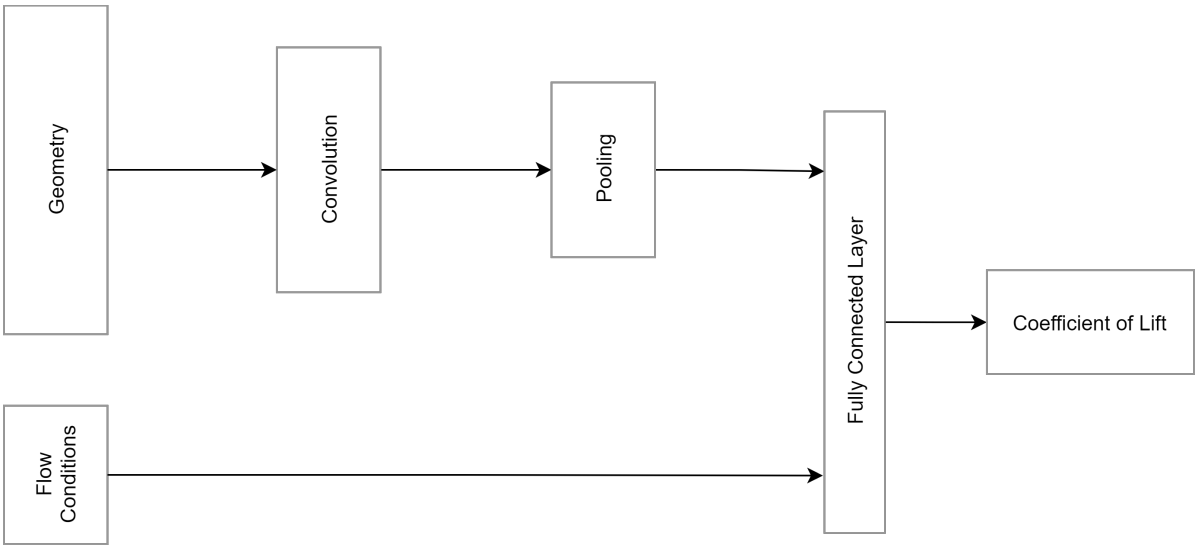


Figure 3.11: Simplified illustration of the AeroCNN-I neural network architecture used to predict airfoil lift coefficient. Image reproduced from Zhang, Sung, and Mavris 2018.

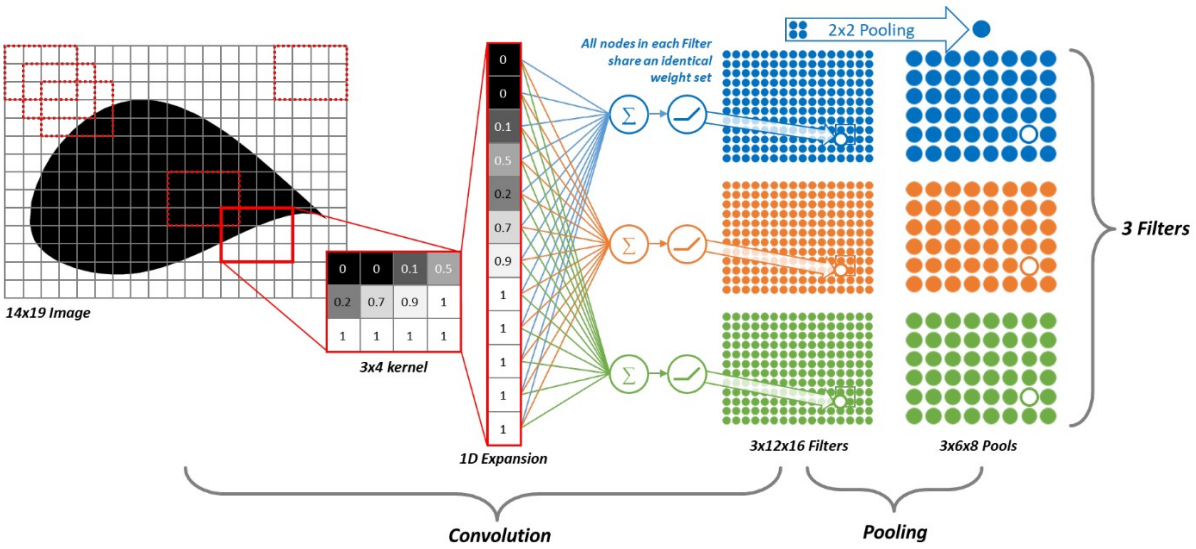


Figure 3.12: Illustration of the convolutional operation part of the AeroCNN-I neural network architecture. Image sourced from Zhang, Sung, and Mavris 2018.

Zhang, Sung, and Mavris 2018 used the LeNet-5 CNN architecture, which is one of the first CNN architectures ever developed. Since then (1998), several *improved* architectures have been proposed, as documented by Khan et al. 2020. Improvement can be in terms of accuracy or training time.

3.4.1. Convolutional Neural Network architecture

Several architectures were tested on the dataset taking inspiration from popular architectures such as AlexNet (Krizhevsky, Sutskever, and Hinton 2017) and ResNet (Kaiming et al. 2016). The architecture, represented by the tab. 3.4, gave good results ($< 10\%$ error) for predicting Thrust, Power requirement and Specific Impulse. 'Mean squared error' was the loss function which was minimized using the Adagrad (Duchi, Bartlett, and Wainwright 2012) optimization algorithm (implemented in TensorFlow). The initial learning rate was set as 0.0001. The number of epochs was set based on how well the training was progressing. If the validation error was decreasing then the training was continued for more epochs. An epoch is an iteration over the entire dataset. In this architecture the *batch size* was set as 32, which is the number of samples from the training dataset used to calculate the values of weights and biases. Each iteration, the mean-squared-error was calculated for 32 samples from the training dataset and based on this error value the weights and biases were updated using the Adagrad algorithm. The number of iterations per epoch = size of training dataset/batch size.

Layer (type) No. of kernels x (kernel size), stride	Output Shape	Param #
Convolution and ReLU: 96 x (7x7), 1x1	(None, 32, 32, 96)	4800
MaxPooling: 1 x (3x3), 1x1	(None, 32, 32, 96)	0
Convolution and ReLU: 256 x (5x5), 1x1	(None, 32, 32, 256)	614656
MaxPooling: 1 x (3x3), 2x2	(None, 16, 16, 256)	0
Convolution and ReLU: 384 x (3x3), 1x1	(None, 16, 16, 384)	885120
Convolution and ReLU: 384 x (3x3), 1x1	(None, 16, 16, 384)	1327488
Convolution and ReLU: 256 x (3x3), 1x1	(None, 16, 16, 256)	884992
MaxPooling: 1 x (3x3), 2x2	(None, 8, 8, 256)	0
Flatten	(None, 16384)	0
Dense and ReLU	(None, 1024)	16778240
Dense and ReLU	(None, 256)	262400
Dense and ReLU	(None, 1)	257

Total parameters to be trained: 20,757,953

Table 3.4: CNN architecture used for creating the response surface.

3.5. Optimization method

Optimization algorithms fall into 2 categories: Deterministic and Stochastic. Deterministic algorithms are typically faster i.e. require fewer iterations, to converge on the solution but they tend to converge to a locally optimum solution. Introducing some randomness in the algorithm (Stochastic algorithms) helps in exploring more of the solution space which can lead to the algorithm finding the globally optimum solution. For an overview of different optimization algorithms refer to Cavazzuti 2013. In this project, the Stochastic algorithm, Genetic Algorithm (GA), was used to find an optimum design of the heaterchip or more specifically, the shape of the slot of the heaterchip.

The Genetic Algorithm (GA) method is an optimization method that is ideally suited to solve problems that contain a large number of input parameters and multiple objectives. Though using GA does not guarantee a globally optimum solution, it could come close due to the stochastic operators in the algorithm that allow it to jump and evaluate $f(x)$ at several random x (from eq. 3.1: $x \in X \subseteq R^n$) at once and then use that information to determine the next set of inputs, $x^{(s+1)}$, where s is the number of iterations completed. Baodong et al. 2011 used GA and found a better solution than what they found by using the deterministic algorithm, *Sequential Quadratic Programming*. In GA the intermediate solutions exchange information which can speed up the process when compared to the *simulated annealing* method (pg. 10 Fang 2007), which is another stochastic optimization algorithm. A drawback of GA is that they can be computationally expensive to run though they are certainly faster than using a brute force method i.e. evaluating every possible parametric combination. It is important to note that several optimization algorithms exist and choosing the right one is not straightforward. Experience and experimentation with models play a role in the selection of an appropriate model and also the selection of the model parameters. Cavazzuti 2013 says that the selection of a suitable optimization method and model parameters is in itself an optimization problem.

GA work as follows (refer fig. 3.13): Design parameters are randomly varied to generate a set of 'initial' designs. Each of these designs is evaluated for its performance. High performing designs are selected which are then grouped into pairs. 'Offspring' or new designs are generated by letting each pair exchange parametric values with each other as shown in fig. 3.13. The new designs are evaluated for performance and thus the loop continues until a stopping criterion set by the user is satisfied. A new design may be mutated i.e. one of its parameters randomly changed to ensure diversity in the population and prevent premature convergence. Mutations should be programmed to decrease as iterations increase otherwise convergence would not be achieved.

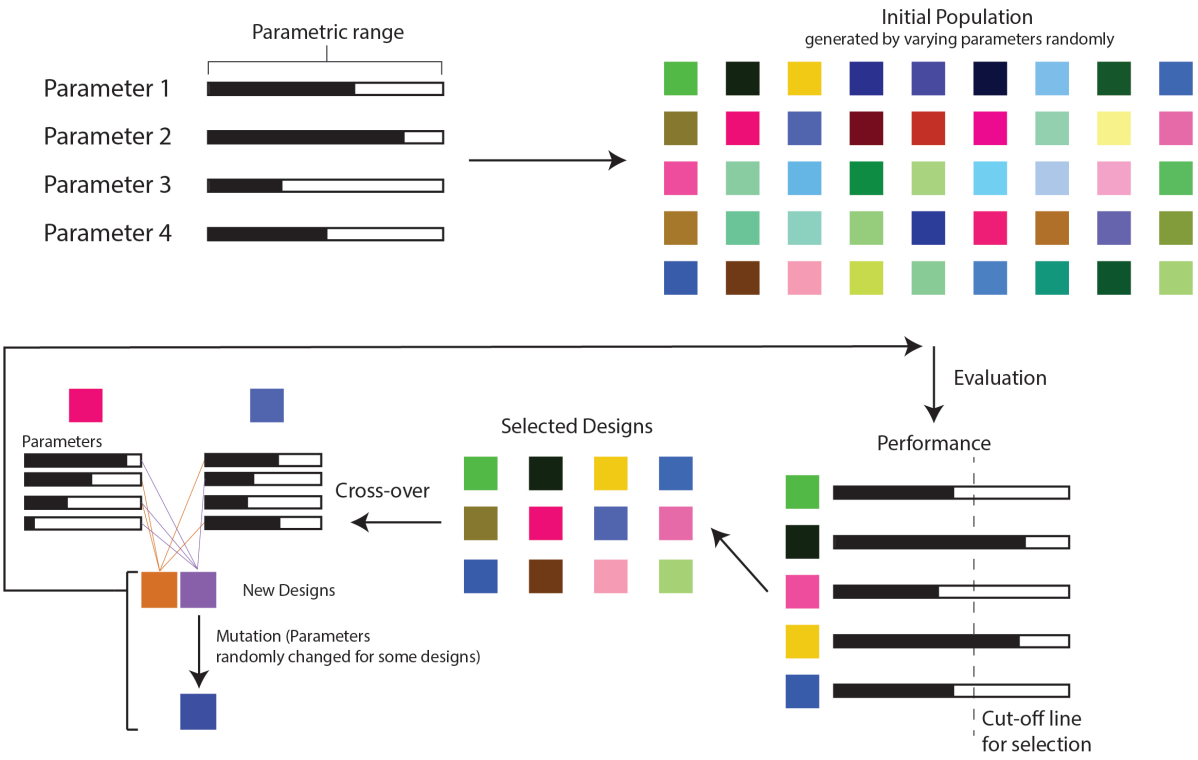


Figure 3.13: Illustration of the steps involved in a Genetic Algorithm.

A few examples of work using GA: Michalak 2019 used GA (evaluation using ANSYS Icepak) to generate heat sink geometry for high power output power electronics such as those installed in electric vehicles; Didominic et al. 2020 parameterized the geometry of an aircraft nozzle and used GA (evaluation using a neural network-based surrogate CFD model) to optimize the design for a low noise signature; Muñoz-Paniagua and García 2020 optimized the geometry of a standard high-speed train design for less drag using GA coupled with a surrogate performance evaluation model instead of a CFD solver. Parameterization of the geometry was done using bézier curves.

3.5.1. Genetic Algorithm setup

The GA architecture used in this thesis is illustrated by fig. 3.14. This is the $\mu+\lambda$ algorithm where μ and λ are the number of parents and children respectively, each generation/iteration. The initial population should be large and as diverse as possible to avoid premature convergence. The solution is said to be converged when subsequent iterations do not produce a better result. 10000 was deemed appropriate according to the computational resource available and the diversity was taken care of by relying on Sobol sequences to generate the geometries. μ and λ were also set as high as possible for the same reason. λ was set high to increase the chances of finding an child better than the parent. If the population is small then there is an increased chance of premature convergence. It is not clear whether cross-over operations should be preferred over mutation operations (Cavazzuti 2013). The values of the hyperparameters were selected by arbitrarily tweaking the values and observing which set of parameters gave the best solution.

In this architecture $\mu = 200$ and $\lambda = 400$. At first, 10000 designs were generated using Sobol Sequences to ensure population diversity. From these 10000 designs, the best 200 were selected which were then used to generate 400 children by applying cross-over (60% probability) or mutation (40% probability) operator. From the $\mu + \lambda$ designs, again, the best 200 are selected. This marks the end of one iteration. On the onset, there is no indication of when the solution will converge. By 150 iterations it was clear that the iterations had converged.

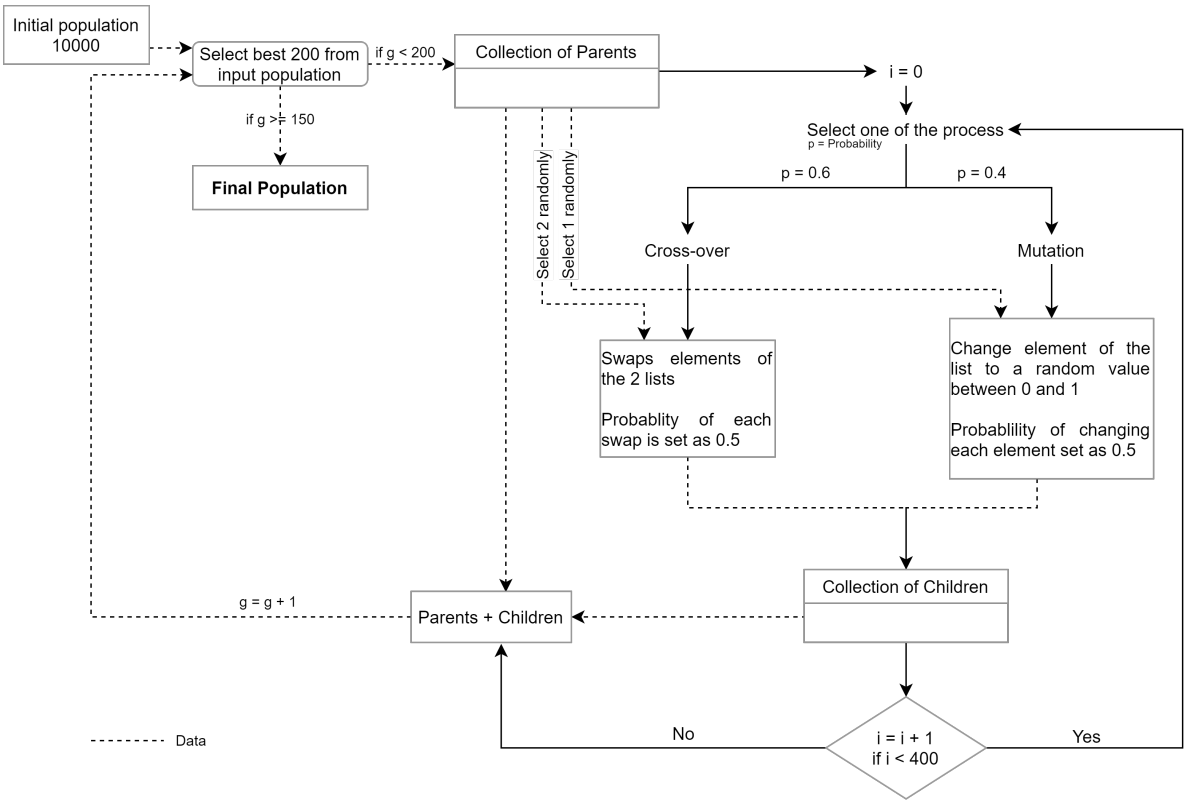


Figure 3.14: GA architecture used for optimization.

4

Geometry Generation

An optimization algorithm controls the value of input variables to see how the output behaves. In this project, the input variables are geometrical parameters that affect the shape of the slot. Each parameter controls the location of a *control point* which are later joined together using a NURBS curve to form a closed curve, defining the perimeter of a slot. Parameters are restricted to, $p_i \in (0, 1)$. The base geometry (fig. 4.1) consists of radially emanating "spokes" that are separated by the angle, $d\theta = \frac{2\pi}{N_{ctrl}}$, where N_{ctrl} is the number of control points specified by the user. More control points enable the construction of finer geometrical features but also increase optimization time due to the increased number of tunable parameters. Finer features also force the reduction of the time step in simulations (particles get stuck in reflections) which in turn increases the number of iterations needed for the simulation to attain steady state. Additionally, it is possible that thin features might not withstand the force exerted by the fluid flow. The number of control points was set to 8. A higher number of control points enable the creation of complex shapes but due to the issues described above and due to the fact that it is possible to create geometries generated with a lower number of control points using a higher number of control points, 8 control points was deemed appropriate. It is important to know that a better performing design may be achieved with a higher number of control points but that shape, most likely, cannot be generated using 8 control points.

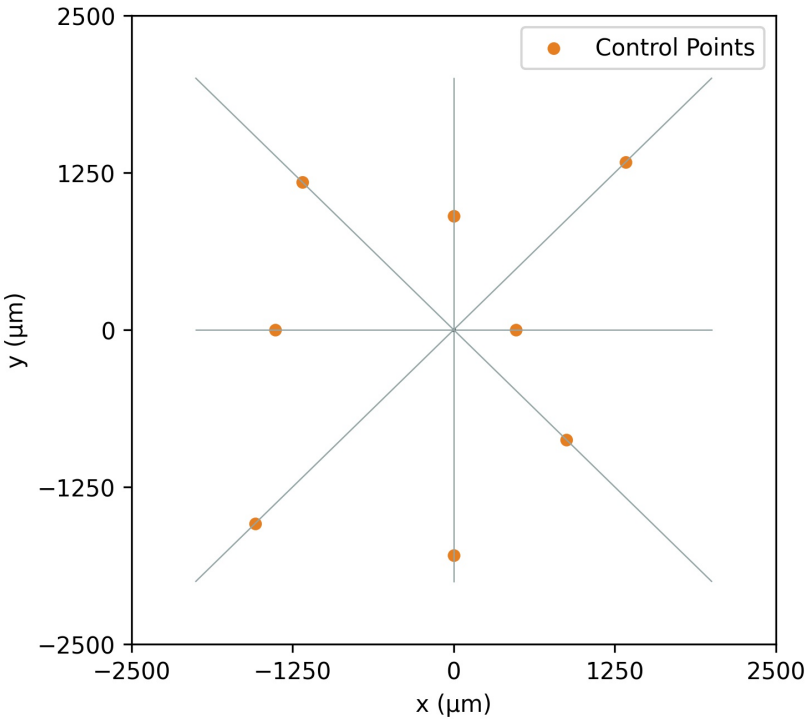


Figure 4.1: Illustrations of base geometry set-up for 8 control points, randomly positioned on a spoke. Each point is allowed to move along one spoke.

For the Deep Reactive Ion Etching (DRIE) process the aspect ratio i.e. width to depth of channel should be less than 20 for a 0.5 mm deep etch (De Boer et al. 2002). It was thus decided that the starting location of the control points will be from a radius which ensures that the distance between the points are greater than 25 μm; 50 μm has been considered. The following process convert p_i to a Cartesian coordinate:

1. Angle between spokes,
$$\Delta\theta = \frac{2\pi}{N_{ctrl}} \tag{4.1}$$
2. Minimum radius to ensure < 20 aspect ratio,
$$R_{min} = \frac{50}{2\sin\Delta\theta/2} \tag{4.2}$$
3. Unit vectors of each control point,
$$\hat{v_i} = [\cos i\Delta\theta, \sin i\Delta\theta] \; for \; i = 0, 1, 2, \ldots, N_{ctrl} - 1 \tag{4.3}$$
Parameter, $p_i \in (0, 1)$
4. Radial distance of control point from center of heaterchip,
$$r_i = p_i(L/2 - R_{min}) + R_{min} \; for \; i = 0, 1, 2, \ldots, N_{ctrl} - 1 \tag{4.4}$$
where L is the length of the square to which the slot is constrained to

5. Cartesian coordinate of control point,

$$x_i, y_i = \frac{\hat{v_i} r_i}{\max[abs(\cos i\Delta\theta), abs(\sin i\Delta\theta)]} \text{ for } i = 0, 1, 2, \ldots, N_{ctrl} - 1 \tag{4.5}$$

Using eq. 4.1—4.5, fig. 4.2 was plotted for 3 different values for all the parameters.

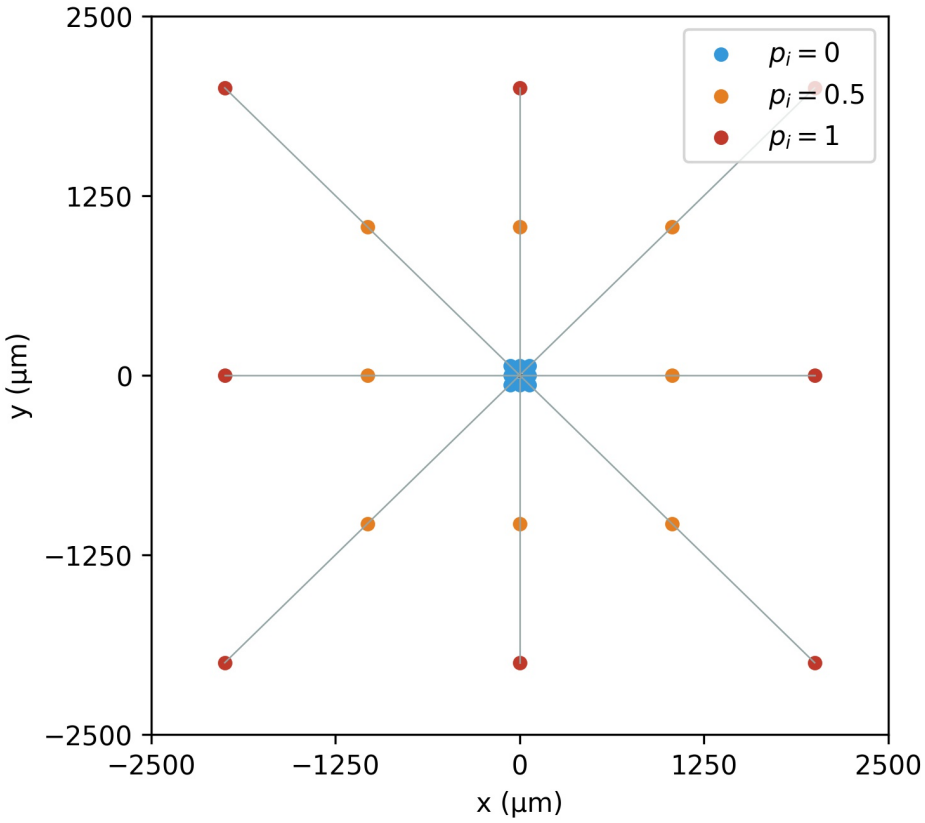


Figure 4.2: Visualization of the effect of parametric value on the real control point location using eq. 4.1—4.5 for 3 different parameter values.

Non-uniform rational basis spline (NURBS) was then used to join the control points to form a closed curve. Another solution is to use bézier curves to create closed curves (ex. fig. 4.3). However, a disadvantage of using bézier curves is that it cannot generate complex curves because each control point affects the entire global curve which makes making local geometry changes without affecting the global shape difficult. Straathof 2012 hence recommends the usage of b-splines or the more general NURBS with which one gets more control of the geometry (see. fig. 4.4). In B-splines, the curve is controlled by the inputs: position of control points and the degree of the polynomial that joins 2 knot points. A disadvantage of B-splines is that *rational* curves (parametric curve in homogeneous form) like circles, ellipses and hyperbolas cannot be created. NURBS solves this by introducing *weights* in the equation. When the weights of the control points are unity then the curve is equivalent to a B-spline curve. Fig. 4.4 illustrates how weights and polynomial degree affect the NURBS curve for the same control points. The

NURBS python library by Bingol, n.d. has been used to generate the curves. listing 4.1 shows how to make a NURBS curve using the library. For more information on NURBS refer Piegl and Tiller 1995.

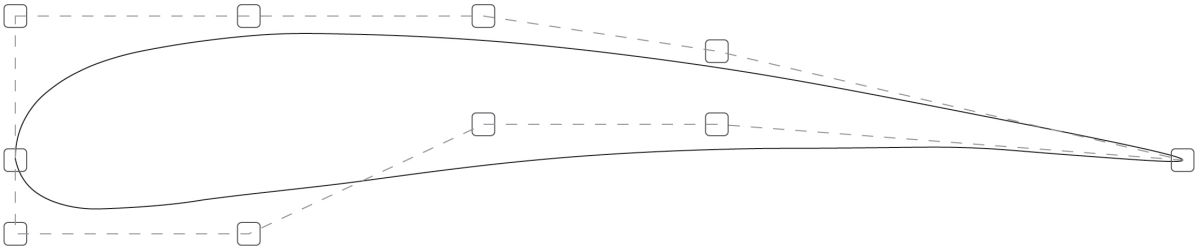


Figure 4.3: An airfoil created using control points and bézier curve. Image reproduced from pg. 24 Straathof 2012.

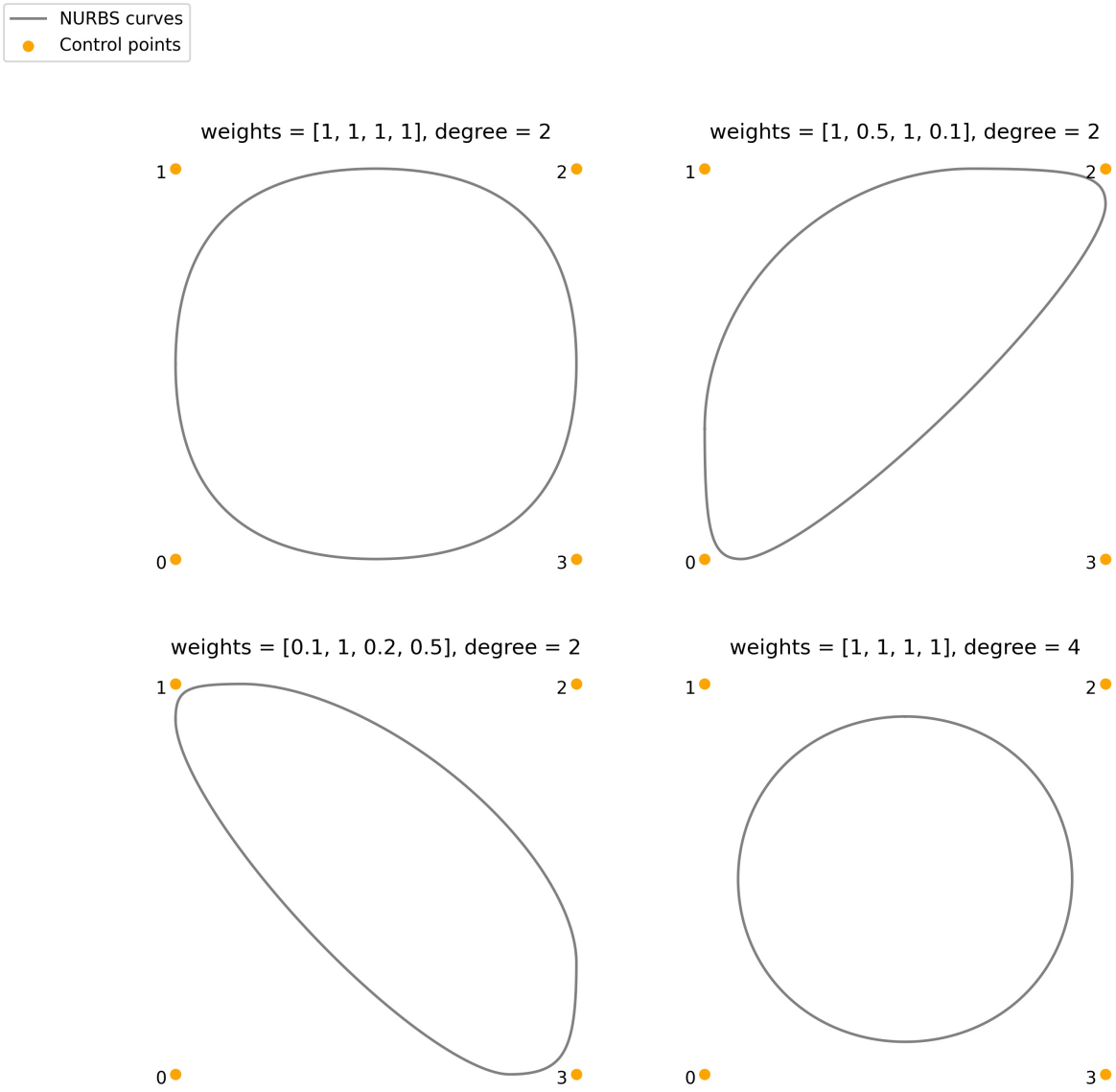


Figure 4.4: Closed curves generated by the NURBS formulation for the same control points but different weights and degree. The curve was generated using the NURBS python library by Bingol, n.d.

```
import matplotlib.pyplot as plt
import numpy as np
from geomdl import NURBS
from geomdl import knotvector

ctrl_pts = np.array([[0,0,0],
                     [2,2,0],
                     [4,9,0],
                     [5,4,0],
                     [9,0,0]])

plt.figure(figsize = (5,5)); plt.xlabel('x'); plt.ylabel('y')
plt.plot(ctrl_pts[:,0], ctrl_pts[:,1], linewidth = 0.6, c = 'grey')
plt.scatter(ctrl_pts[:,0], ctrl_pts[:,1], s = 30, c = 'orange', label = 'Control
    Points')

# nurbs section
curve = NURBS.Curve()
curve.degree = 2
# additional ctrl pts to be added = degree of curve (to make close curve)
curve.ctrlpts = np.append(ctrl_pts, [i for i in ctrl_pts[0:curve.degree]], axis=0)
weights = [0.1,0.1,0.1,0.1,0.1]  # number of original control points
curve.weights =  weights + [i for i in weights[0:curve.degree]] # len(curve.
    ctrlpts)
m = curve.degree + len(curve.ctrlpts)  # = min number of knot points - 1
curve.knotvector = [i/m for i in range(0, m+1)]  # unifromly distributed knot
    points
curve.delta = 0.001
pts_example = np.array(curve.evalpts)
plt.plot(pts_example[:,0], pts_example[:,1], c = 'black', label = 'NURBS curve')
plt.legend()
```

Listing 4.1: Generating a NURBS curve using the NURBS python library by Bingol, n.d.

As mentioned in sec. 3.4, an SDF is an appropriate input for a CNN. To convert the NURBS geometry to an SDF, it was necessary to first rasterize the image i.e. convert the slot geometry into pixels, a 2D matrix. This was achieved by using the python library, scikit-image, n.d. (see listing 4.2). For 8 design parameters, the smaller features were not lost when rasterizing to a 32 x 32 pixel image i.e. the characteristic length of geometrical features were more than $4000\mu m/32 = 125\mu m$. Increasing the number of control points would mean finer features, and thus the number of pixels in the raster image would have to be increased accordingly. A visualization can be seen in fig. 4.5. From the raster image, the SDF matrix is generated using the scikit-fmm python library (Furtney, n.d.) (see listing 4.3).

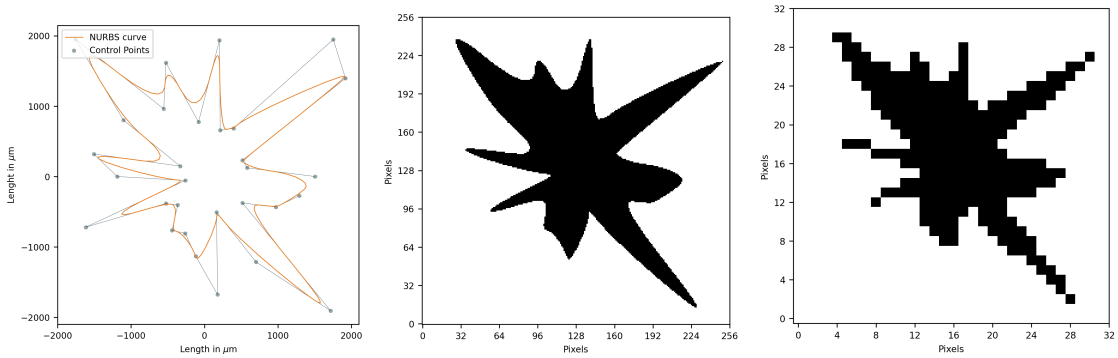


Figure 4.5: Rasterization of a closed 30-control-point NURBS curve (leftmost) to a 256 x 256 px (middle) and a 32 x 32 px image (rightmost).

```
import numpy as np
from skimage.draw import polygon  # for line rasterization
pts = Set of points constituting the NURBS curve
pixel_len = 32  # image size = pixel_len x pixel_len px
square_length = 4000  # m, length of square to which the slot is restricted to
scale_factor = square_length/pixel_len
# points also need to be translated to the first quadrant
pts_scaled = (pts + square_length/2)/scale_factor
img = np.zeros((pixel_len, pixel_len), dtype=np.double)
rr, cc = polygon(pts_scaled[:, 1], pts_scaled[:, 0], img.shape)
area = len(rr)*(scale_factor*1e-6)**2  # approx. area of slot
img[rr, cc] = 1  # raster matrix. Plot using plt.imshow(img)
```

Listing 4.2: Rasterization using the scikit-image python library.

```
import skfmm
import numpy as np
phi = img  # img is the raster matrix
pixel_len = 32  # image size = pixel_len x pixel_len px
sdf = skfmm.distance(phi)
sdf_norm = sdf/(pixel_len/2)  # To keep the values near 0 to 1. Helpful while
    training CNN.
```

Listing 4.3: SDF using the scikit-fmm python library.

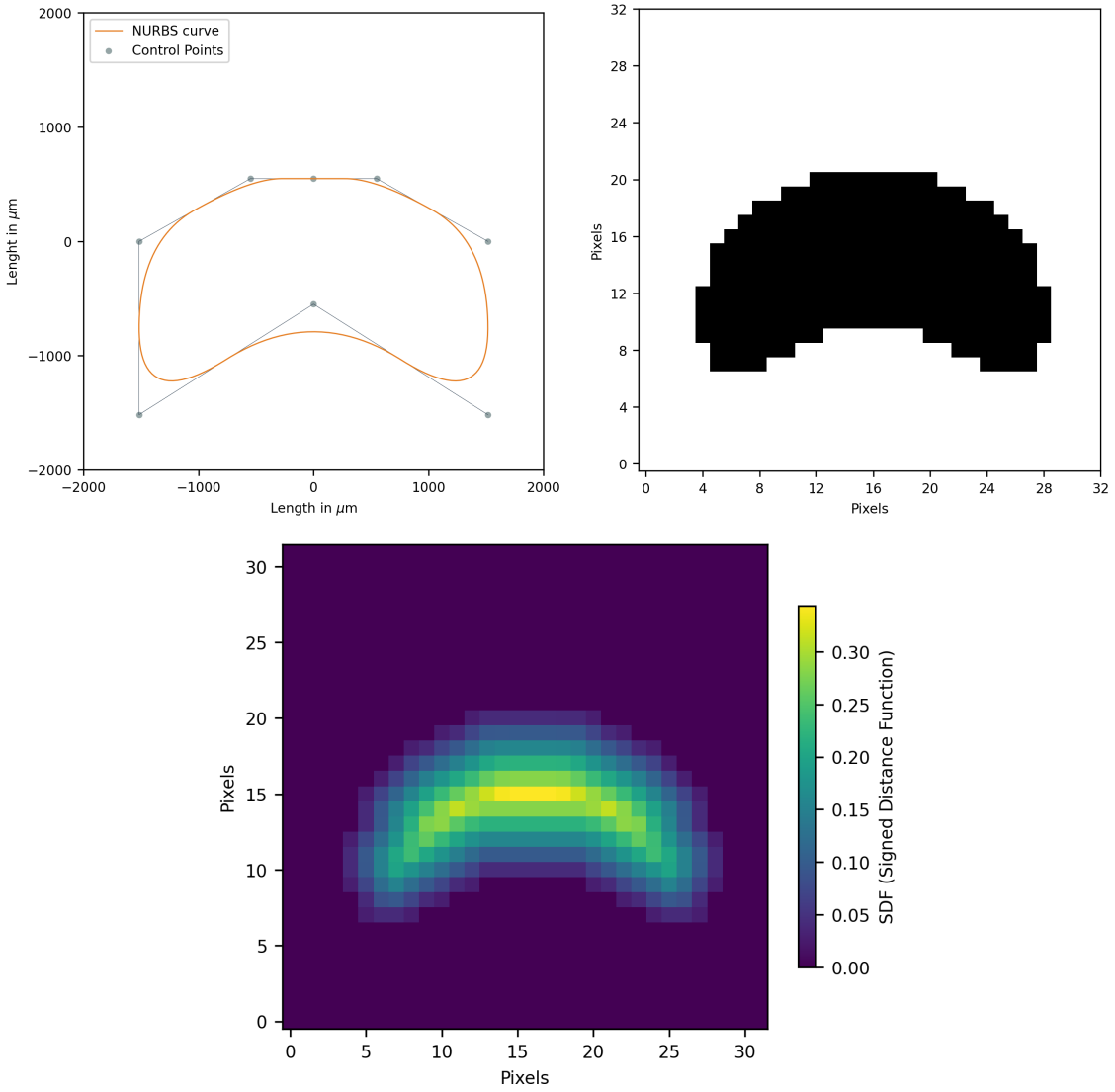


Figure 4.6: NURBS curve to Raster image to SDF, which were generated using the python libraries Bingol, n.d., scikit-image, n.d. and Furtney, n.d. respectively.

A few points regarding the overall geometry — The heaterchip is a multi-microchannel heat exchanger similar to the Ketsdver (Ketsdever, Lee, and Lilly 2005) (fig. 2.4) and Guerrieri (Daduí C Guerrieri et al. 2017) (fig. 2.7) designs with the difference that the shape of the microchannel/slot will be optimized using RSM. The idea is to optimize the design of 1 microchannel. If higher thrust is needed, the number of microchannels (with the optimum design) in a heaterchip can be increased accordingly. It is assumed that the heaterchip has a dimension of 50 mm x 50 mm x 0.5 mm, where 0.5 mm is the thickness of a standard silicon wafer. The depth of silicon nitride (≈500 nm (Daduí C Guerrieri et al. 2017)), thin-film heater (≈200 nm (Daduí C Guerrieri et al. 2017)) and gold layer (≈800 nm (Wong, Reed, and Ketsdever 2001)) was assumed to be negligible. These layers are important, but including them in the geometry would complicate the DSMC simulation unnecessarily. Heat transfer from the slot walls to the propellant predominantly takes place on the 0.5 mm thick silicon surface. A

unit is now defined as a 5 mm x 5 mm square, 0.5 mm thick, which contains 1 microchannel constrained to a 4 mm x 4 mm square as seen in fig. 4.1. Multiple units in a heaterchip increases thrust (due to increased mass flow rate) with the downside of increased power requirement.

The complete geometry (fig. 4.7) for the simulation was made using the open-source software, Gmsh ("Gmsh," n.d.). The length of outlet-walls and inlet-walls is 3 mm, outlet and inlet is a square of 5 x 5 mm, and the slot-walls are 0.5 mm in length. The inlet volume is assumed to be filled with Argon at 50 Pa and 300 K. The inlet pressure and temperature could also be variables but due to computational constraints, this was not explored.

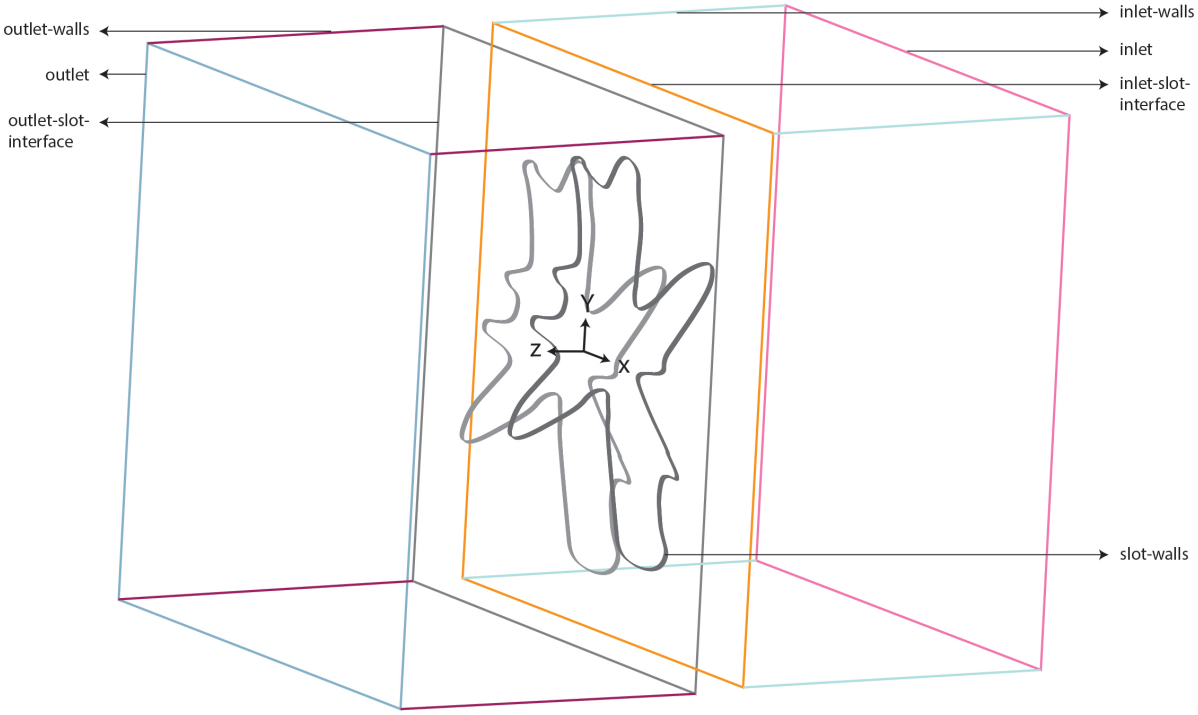


Figure 4.7: 3D representation of the simulation CAD model.

VSTRAP requires a 2D mesh with the mesh normals pointing 180° away from particle reflection normals. The mesh was generated using Gmsh, illustrated in fig. 4.8.

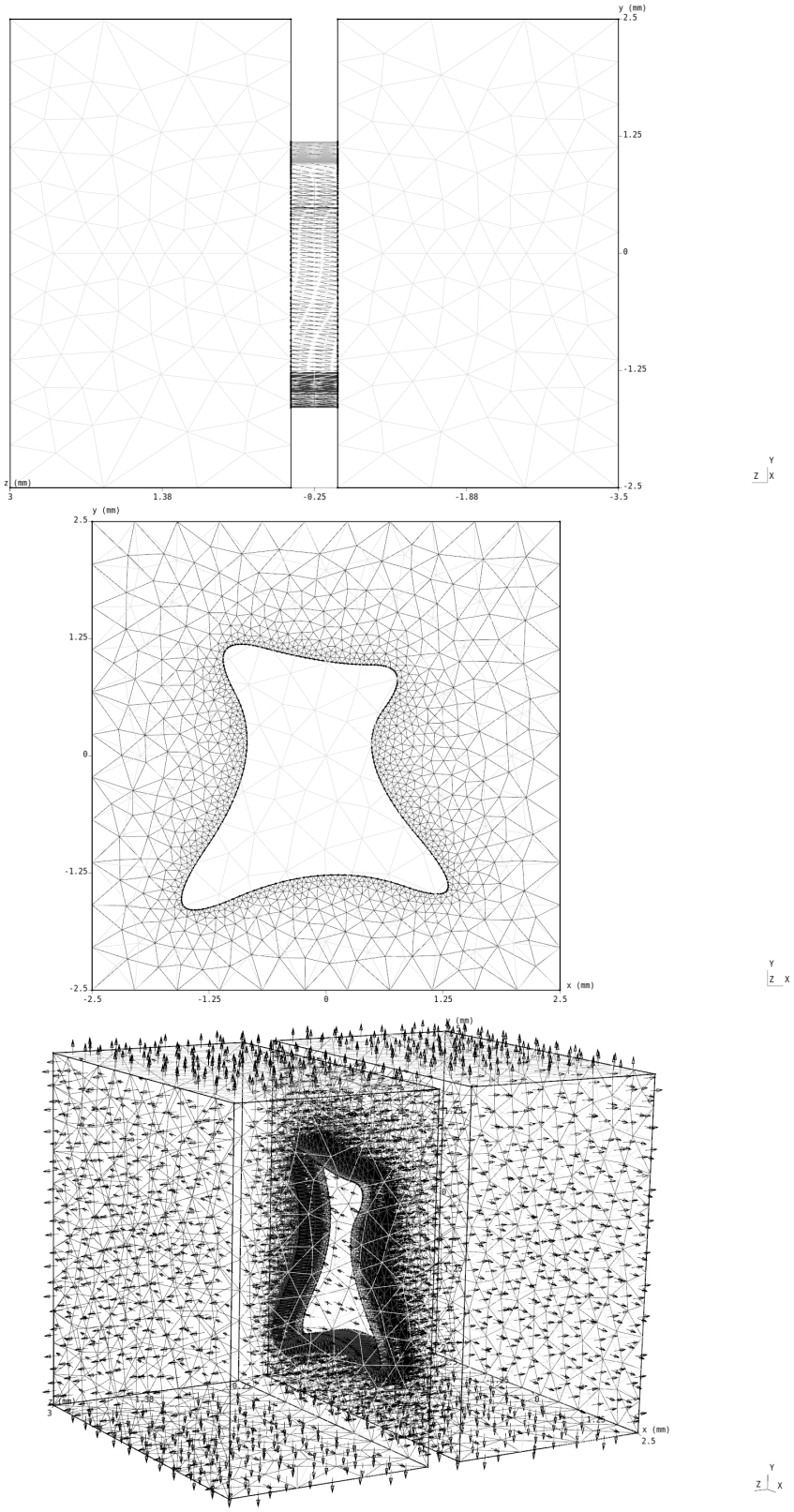


Figure 4.8: 3D modelling and Mesh generation using Gmsh. The last image shows the mesh normals pointing away from the particle reflection normal.

5

RSM: Results

5.1. Design of Experiment (DoE) results

The total number of geometry parameters was taken to be 8. All of the 8 parameters correspond to the position of the control points only. The weights were not taken as parameters to reduce the number of parameters and for a more practical concern of removing near-sharp curvatures from the slot geometry, which can be prone to structural failure. Increasing the value of the weight (associated with a control point) will pull the curve towards that control point and can, as illustrated in fig. 5.1, result in near-sharp curvatures in the geometry.

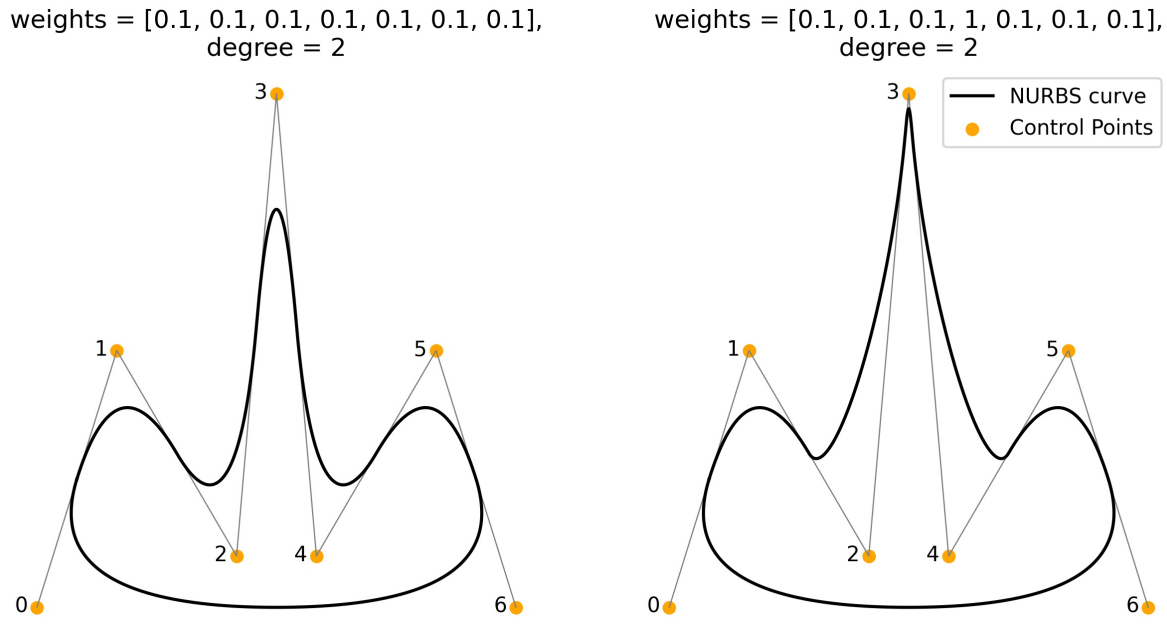


Figure 5.1: Creating a near-sharp curvature region by changing the weight of point 3 from 0.1 (left) to 1 (right).

A total of 512 designs were created, with the parametric values determined using Sobol sequences. For illustration, 8 results have been plotted in fig. 5.2 using listing 5.1 to generate the parametric values, p_i (see eq. 4.3). These geometries are now referred to as "Polar" based on the movement of the control point constrained to a radial line.

```
import numpy as np
import tensorflow as tf

control_pts_count = 8
design_iterations = 8
#  tf.math.sobol_sample(parameters=column, samples=row) -->  designs_x_params
designs_x_params = tf.math.sobol_sample(control_pts_count, design_iterations)
```

Listing 5.1: Generation of Sobol sequences using the TensorFlow Python library.

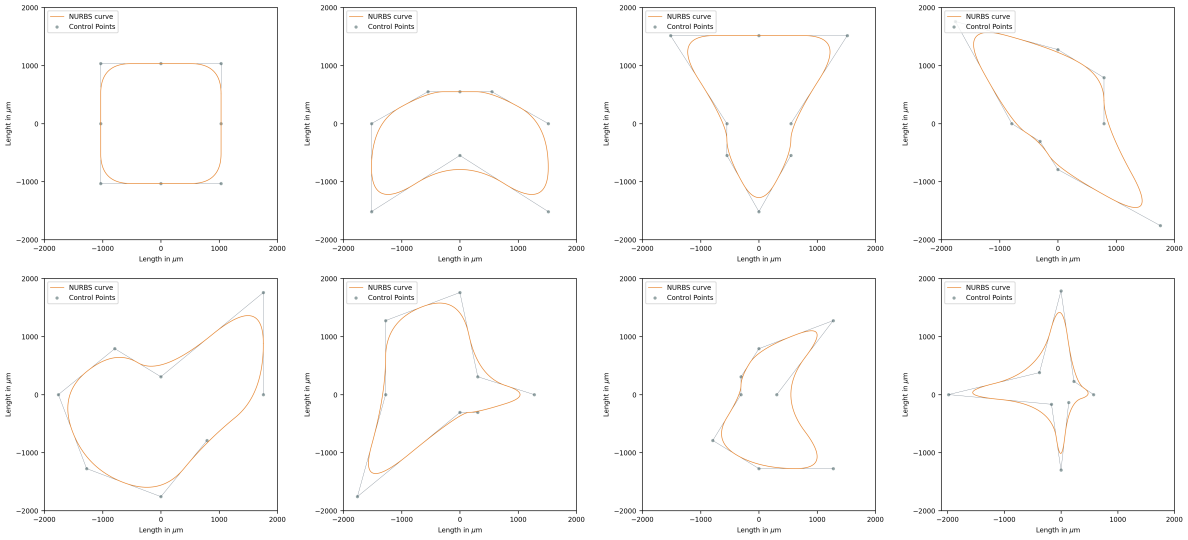


Figure 5.2: Shape of slots generated by joining control points using NURBS, the positions of which were determined using Sobol sequences.

In addition to polar geometries, rectangular geometries were also generated for comparison. As the number of parameters for a rectangle is just 2 (length and breadth), the design space is sufficiently explored using 512 samples generated using Sobol sequences as illustrated in 5.3. For illustration, 8 results have been plotted in fig. 5.2. The rectangular corners were rounded to prevent simulation particles from getting stuck at the corners though it might also prevent stress concentration at the corners when the heaterchip is under structural load. The rectangles were also made using NURBS in which the curve degree was set as 2 and the control points (starting from left bottom and going clockwise) were assigned the weights: 0.99, 1, 0.99, 1, 0.99, 1, 0.99, 1.

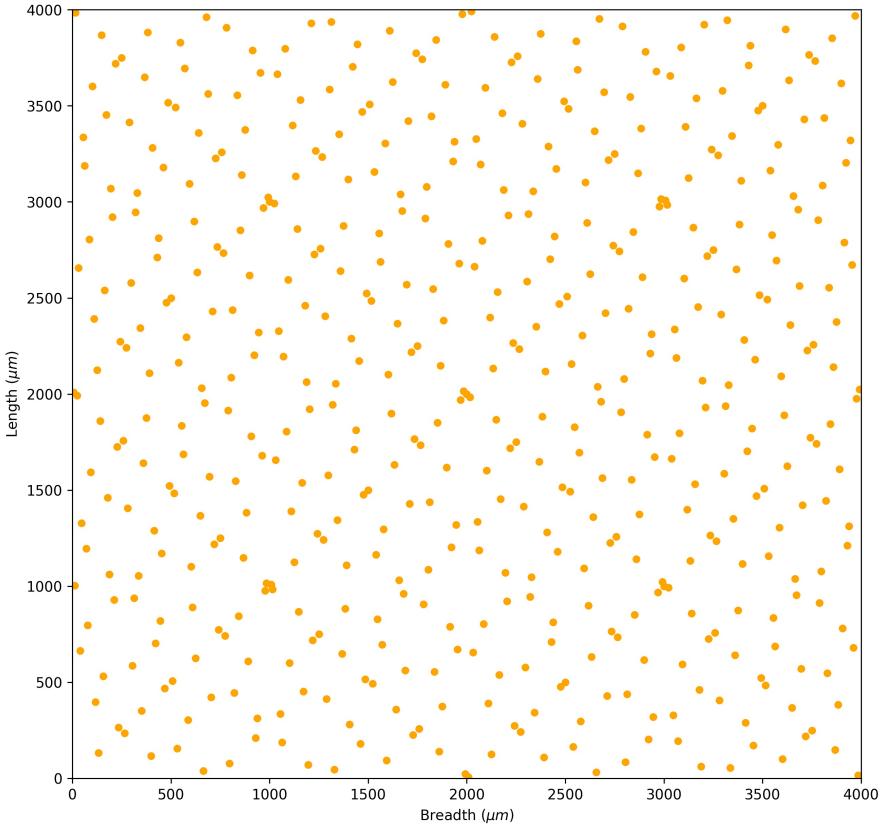


Figure 5.3: Design samples for the rectangular geometry generated using Sobol sequences.

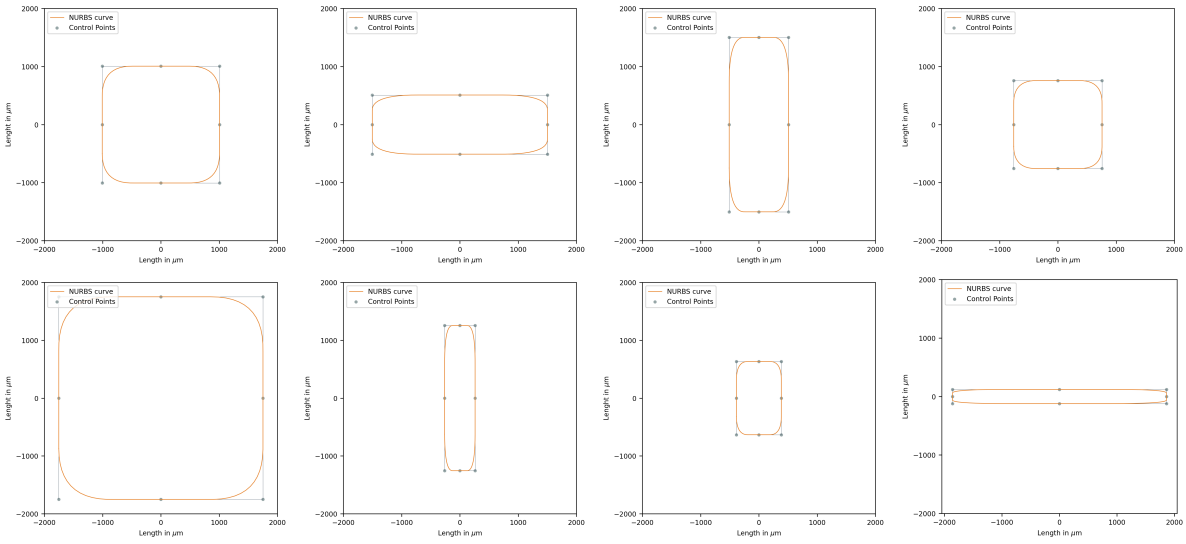


Figure 5.4: Rectangular slots generated by joining control points using NURBS, the positions of which were determined using Sobol sequences.

5.2. DSMC simulations

5.2.1. Validation

The experimental work by Ketsdever, Lee, and Lilly 2005 was chosen for validating the simulation software and the setup. So far it is the only experimental work with argon as the propellant. The software setup for validation is tabulated in tab. 5.1 and has been explained in section 3.3.2. The heaterchip contains 44 microchannels each of which has a dimension of 5.375 mm length x 0.1 mm width and 0.5 mm depth. The geometry is similar to fig. 4.7 except that the slot is a rectangle with the aforementioned dimensions and the inlet and outlet volumes have the dimension 7 mm length x 0.7 mm width x 3 mm depth (fig. 5.5). The length and width of the inlet and outlet geometry was calculated according to the spacing between the slots in the heaterchip. The outlet depth was taken to be 3 mm but a study on the effect of variation of this dimension on the result was not performed. In Daduí C Guerrieri et al. 2016, the outlet depth was taken as 100 μm (for a slot diameter of 200 μm and slot depth of 500 μm) which gave some assurance on the 3 mm value.

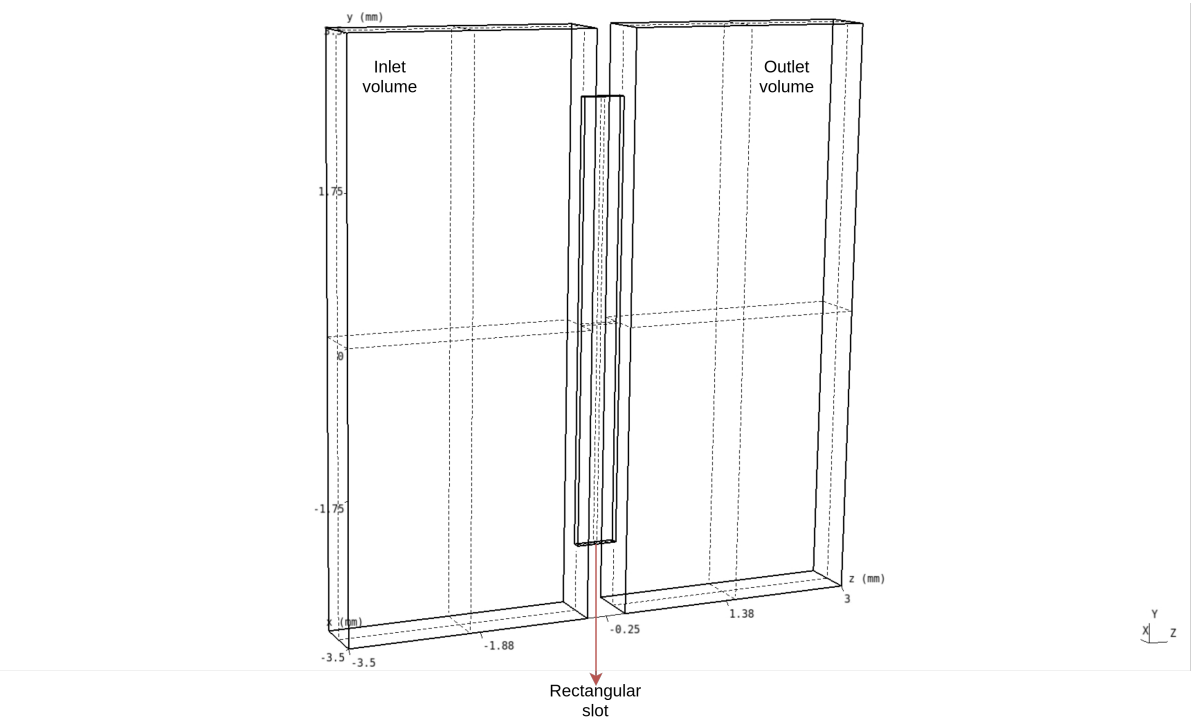


Figure 5.5: 3D model of the simulation domain for validation.

Physical surface name	**Property**	**Value**	
outlet-slot-interface inlet-slot-interface slot-walls	Maxwell -scattering Acc. Coeff. 0.91	Cases: 575 K, 478 K, 375 K, 323 K	
inlet	Inlet boundary (number density and temperature specified by user)	Propellant Pressure$_{inlet}$ (Pa) Temperature$_{inlet}$ (K) Number density (molecules/m^3) Total cross section (m^2) Mean Free Path (m) Mean velocity (m/s) Argon molecular mass (kg) Particle weight < Δt (s) <	Argon 112 300 $2.704E$+22 $3.31E$-19 $1.12E$-4 432.8035 $6.63E$-26 $3.78E$+9 $8.6E$-8
outlet outlet-walls	Vacuum		
inlet-walls	Specular surface		

Table 5.1: Simulation setup for running the validation case. Underlined quantities are the input to the VSTRAP software. "Physical surface name" refers the parts labelled in fig. 4.7.

Note: Particle weight was taken $5E$+8 and Δt was taken as $5E$-9 seconds. Steady-state was obtained after 20000 iterations (for lower heaterchip temperatures it was necessary to run more iterations in order for the simulation to attain steady-state), by which point of time there were ≈600000 simulated particles in the simulation domain.

The deviation of the simulation results from the experimental values (including error bars) is low, as seen in tab. 5.2 and in fig. 5.6, but it is also necessary to point out the imperfection of experiments. Background pressure and temperature in the vacuum chamber, leakage, and thrust stand inaccuracies are some of the problems that can affect the results of experiments.

Heaterchip Temperature		Experiment by Ketsdever, Lee, and Lilly 2005	VSTRAP simulation	Deviation
575 K	Thrust (mN)	0.6703 to 0.7384	0.64831	3.28%
	ISP (s)	48.66 to 54.05	49.78	0.00%
478 K	Thrust (mN)	0.6142 to 0.6769	0.5948	3.17%
	ISP (s)	45.11 to 50.00	45.77	0.00%
375 K	Thrust (mN)	0.5450 to 0.6010	0.5774	0.00%
	ISP (s)	39.77 to 44.44	40.10	0.00%
323 K	Thrust (mN)	0.5065 to 0.5593	0.56	0.12%
	ISP (s)	37.55 to 41.33	37.08	1.00%

Table 5.2: Simulation validation using Ketsdever, Lee, and Lilly 2005 experimental results.

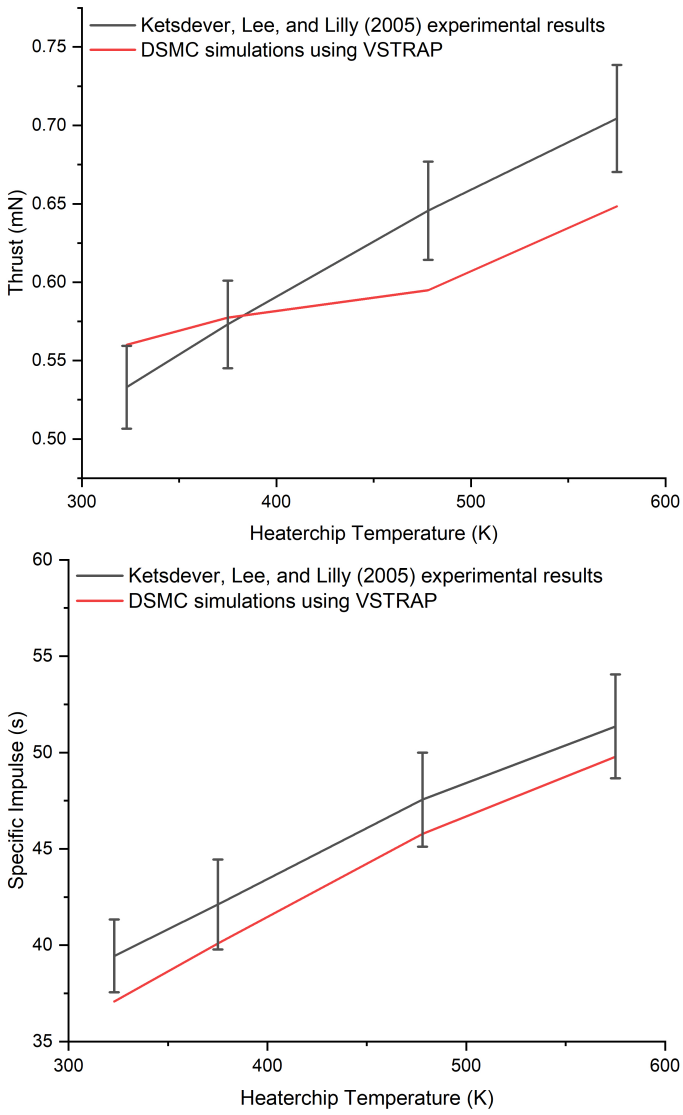


Figure 5.6: Comparison of experimental results by Ketsdever, Lee, and Lilly 2005 (with error bars) with simulations carried out using VSTRAP.

5.2.2. DSMC Simulation results

The 512 polar and 512 rectangular designs generated from the parameters from the DoE run were simulated in VSTRAP using the simulation setup tabulated in tab. 3.3. The simulation data (including CAD files, input files, SDF matrices, post-processing code, etc.) has been published in Chandra 2021a. VSTRAP outputs the particle flux and the normal momentum flux at the slot exit. From these outputs, the performance parameters were calculated using eq. 5.1 and eq. 5.2.

$$\begin{aligned} T &= N\frac{\sum^n P}{n\Delta t} \\ \dot{m} &= N m_{argon}\frac{\sum^n J}{n\Delta t} \\ I_{sp} &= \frac{T}{\dot{m} g_0} \end{aligned} \tag{5.1}$$

Where T is thrust, I_{sp} is specific impulse, N is the number of slots/microchannel in the heaterchip which has been taken as 100, P is the total normal momentum flow in the outflow direction at the slot exit, Δt is the simulation time-step which is $7E$-8 s, n is the number of time-steps considered for calculating the average value, m_{argon} is the molecular mass of Argon which is $6.6335214633E$-26 kg, J is the total particle flow in the outflow direction at the slot exit, and $g_0 = 9.8066\ m/s^2$.

$$\begin{aligned} P_{RadiationLoss} &= \sigma\epsilon(A_{heaterchip} - N A_{slot})(T_c - T_{space})^4 \\ P_{FlowLoss} &= \dot{m} C_p (T_c - T_{inlet}) \\ P &= P_{RadiationLoss} + P_{FlowLoss} \end{aligned} \tag{5.2}$$

where σ is the Stefan-Boltzman constant = $5.670373E$-8 $\frac{W}{m^2K^4}$ and ϵ is the emissivity of the outermost heaterchip surface. The surface is assumed to be coated with a thin-film gold layer which has an emissivity of 0.02. $A_{heaterchip}$ is the surface area of the heaterchip which is 25 cm^2 and A_{slot} is the cross-sectional area of one microchannel (also written as 'slot area' in this text). The heaterchip is at a temperature, T_c of 600 K and is radiating to space at $T_{space} = 0$ K. Whenever a fluid particle collides with the heaterchip surface, energy is transferred from the surface wall to the fluid particle. However the total energy transferred from the surface to the fluid is, as of yet, not a VSTRAP output. Hence, an alternative formulation was devised. In this formulation of $P_{FlowLoss}$, it is assumed that all particles flowing through the slot exit attains the heaterchip temperature. C_p is the specific heat of the fluid which was calculated using the equation,

$$C_p = \frac{\zeta + 2}{2}\frac{k_B}{m_{argon}} \tag{5.3}$$

Where ζ is the degree of freedom (= 3 for Argon) . Due to the assumption that all particles attain T_c temperature, the power estimated will be higher than reality.

Thrust efficiency (η_{eff}), was calculated using eq. 5.4. This is equivalent to the ratio of *input power to the heaterchip* to *useful output power generated by the thruster*. In Daduí C. Guerrieri

et al. 2018, η_{eff} also includes the power required to maintain the conditions in the tank. This power contribution has not been considered in eq. 5.4 as it would require assuming a propellant management system for a satellite.

$$\eta_{eff} = \frac{T \cdot I_{sp} \cdot g_0}{2P} \tag{5.4}$$

The performance parameters were plotted against the slot area and are plotted in fig. 5.7. It is clear that the performance of the heaterchip strongly correlates with the cross-sectional area of the microchannel. Polar designs perform better than rectangular designs with the same slot area mostly due to the increased heating surface area (= Curve perimeter $\times$ microchannel depth) that many of the polar designs feature.

Thrust efficiency for polar designs peak at 0.7019 when the slot area of $5.1718E$-6 m^2. The design is illustrated in fig. 5.8. For this design (design number 347 in polar dataset of Chandra 2021a), the thrust = 0.016 N, I_{sp} = 49 s, Power requirement = 5.55 W.

For rectangular designs, thrust efficiency peaks at 0.6566 when the slot area is $6.578125E$-6 m^2. The design (design number 456 in rectangular dataset of Chandra 2021a) illustrated in fig. 5.8. For this design, the thrust = 0.0233 N, I_{sp} = 46.94 s, Power requirement = 8.16 W. Plotting thrust efficiency with respect to the dimensions of the rectangle paints an interesting picture (fig. 5.9). The aspect ratio of the rectangle barely affects the thrust efficiency whereas the slot area is the major factor affecting the performance of the thruster. By way of reference, Ketsdever, Lee, and Lilly 2005 narrow slot design has the dimensions 5.375 mm x 0.1 mm = $0.05375E$-5 m^2. The authors claim that the thrust efficiency is near 15% (25% at higher mass flow rates) and with some improvement in minimizing heat loss from heaterchip to plenum, it can be pushed up to 30%.

Note: Decreasing the number of slots will linearly decrease both thrust and power requirement. I_{sp} is unaffected.

As thrust and power are increasing with slot area, the reason for thrust efficiency peak and decline is due to the decrease in I_{sp} as slot area increases. It can be reasoned that as the area increases, there is a greater chance for a particle near the centre of the heaterchip to not be influenced by the heated slot surfaces. Low values of thrust efficiencies at small areas can be attributed to the low mass flow rate and low heating (due to decreased surface area) when the area is small. The fitting equations for fig. 5.7 are given in Appendix A.

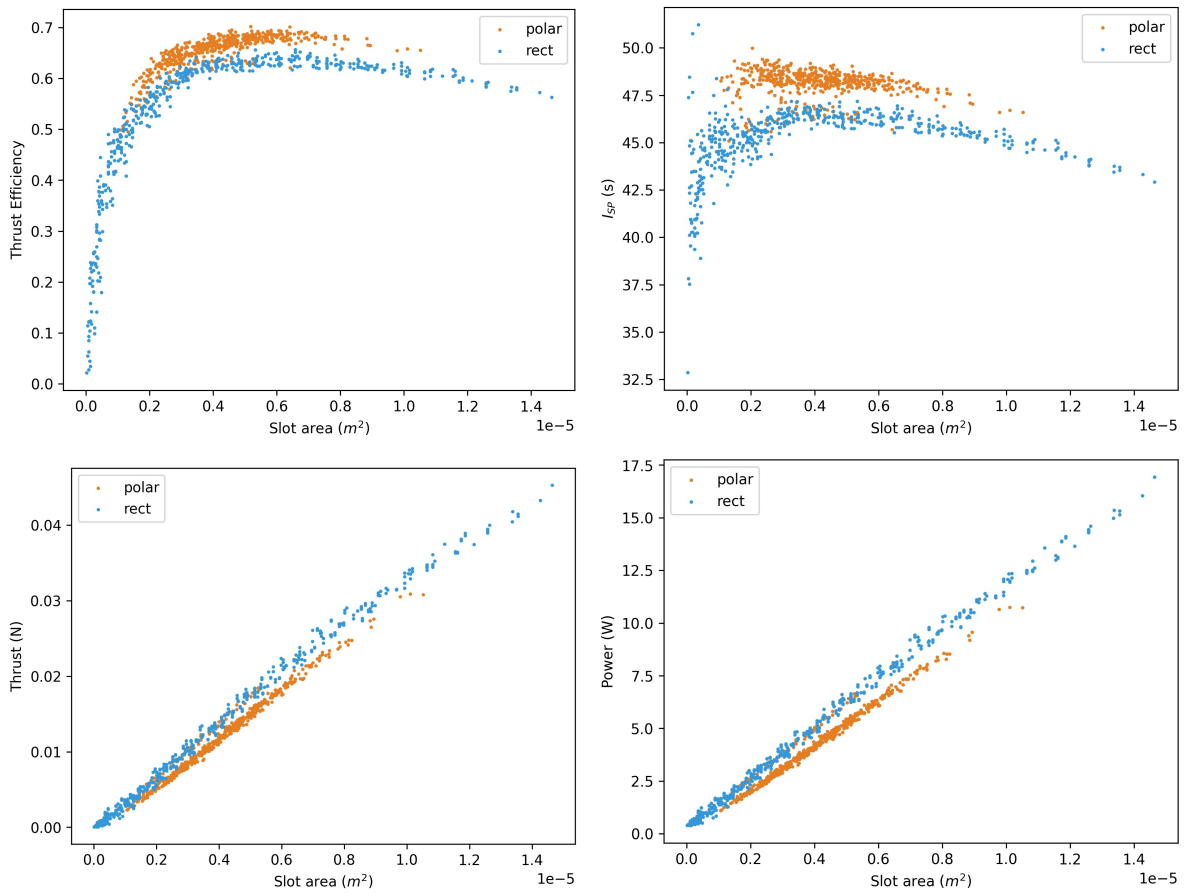


Figure 5.7: Comparison of Polar and Rectangular simulation performance parameters variation with slot area.

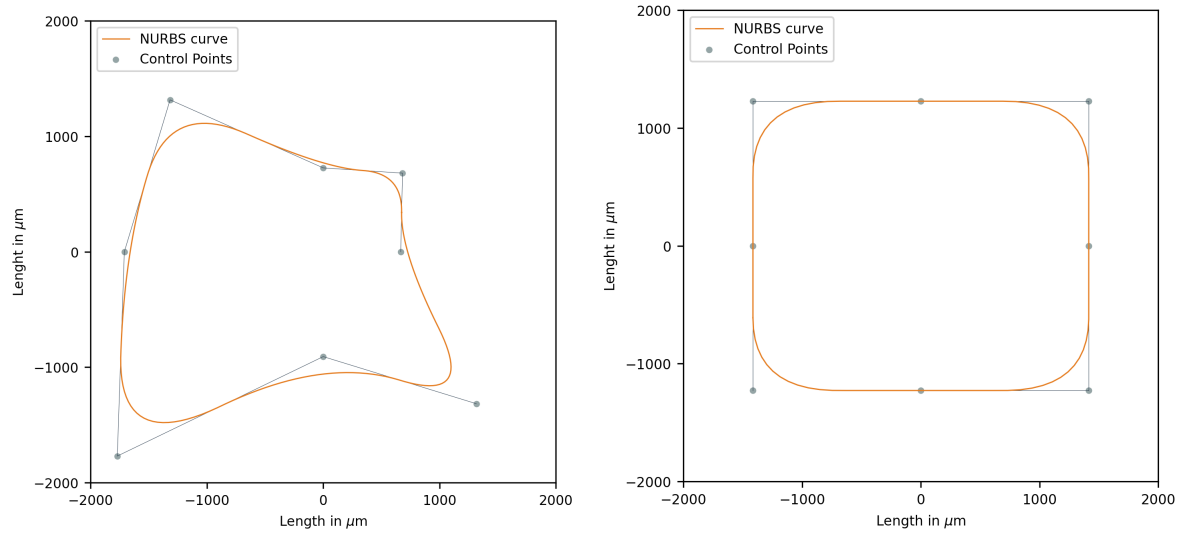


Figure 5.8: Best performing polar design (left) and rectangular design (right) from their respective datasets.

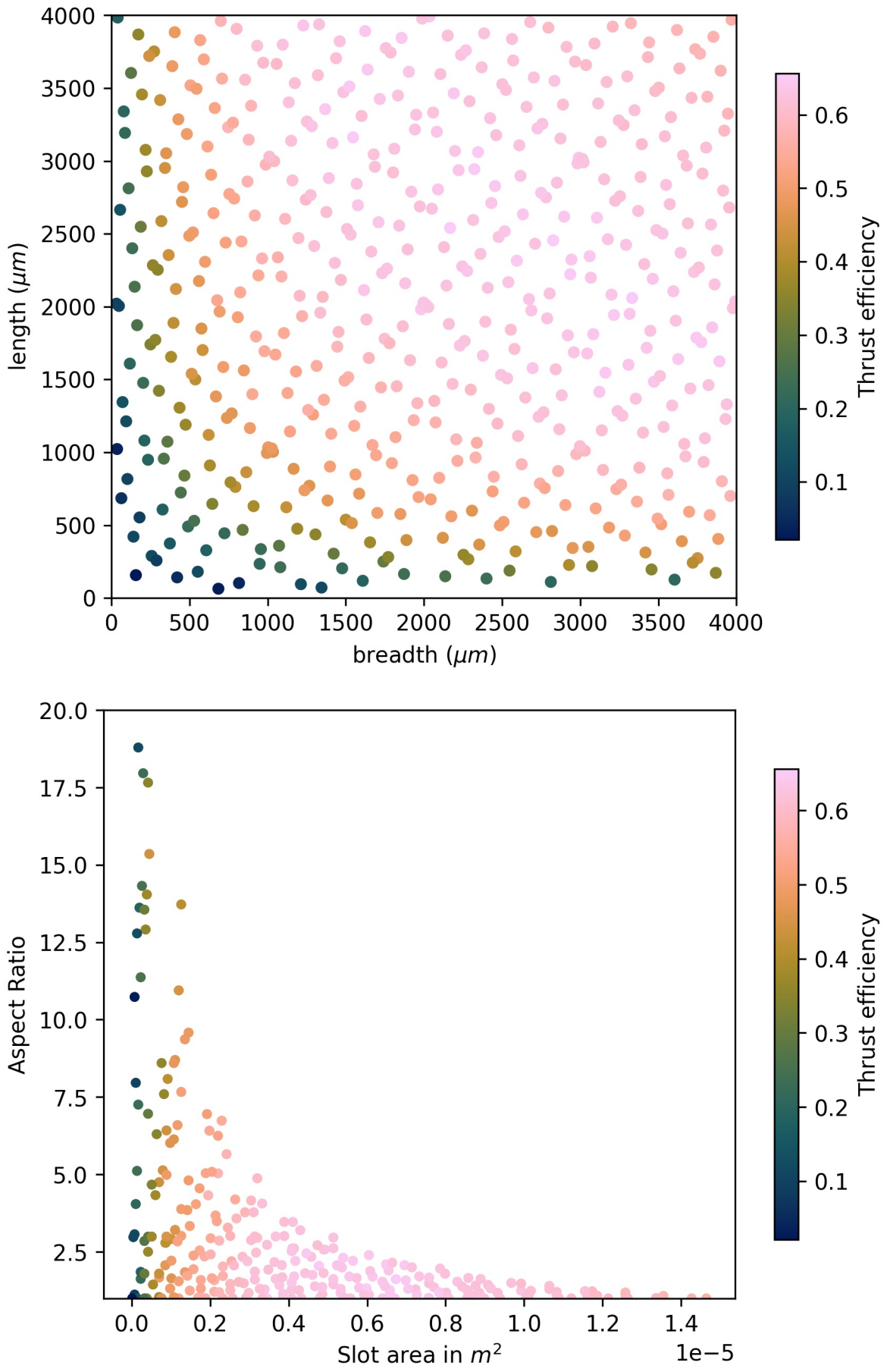


Figure 5.9: Affect of rectangular dimensions on thruster efficiency.

5.3. CNN results

5.3.1. CNN1: Polar dataset

CNN1 was trained on the 512 simulations of polar designs using the architecture described in sec. 3.4.1 for thrust ($CNN1_{thrust}$), I_{sp} ($CNN1_{isp}$), power requirement ($CNN1_{power}$) and thrust efficiency ($CNN1_{eff}$). The dataset was increased 8-fold (see fig. 5.10) by exploiting the fact that the rotation of the slot geometry does not affect the performance parameters. This works because the kernel in a CNN always moves over an image from top to bottom and from left to right. Hence, when a rotated image is fed to the CNN, the neurons in the dense layer receive a completely different set of inputs. This is to say that, to the neural network, the images in fig. 5.10 are distinct and this is why the dataset can be enlarged by rotating the designs. The trained CNN model has been published in Chandra 2021b.

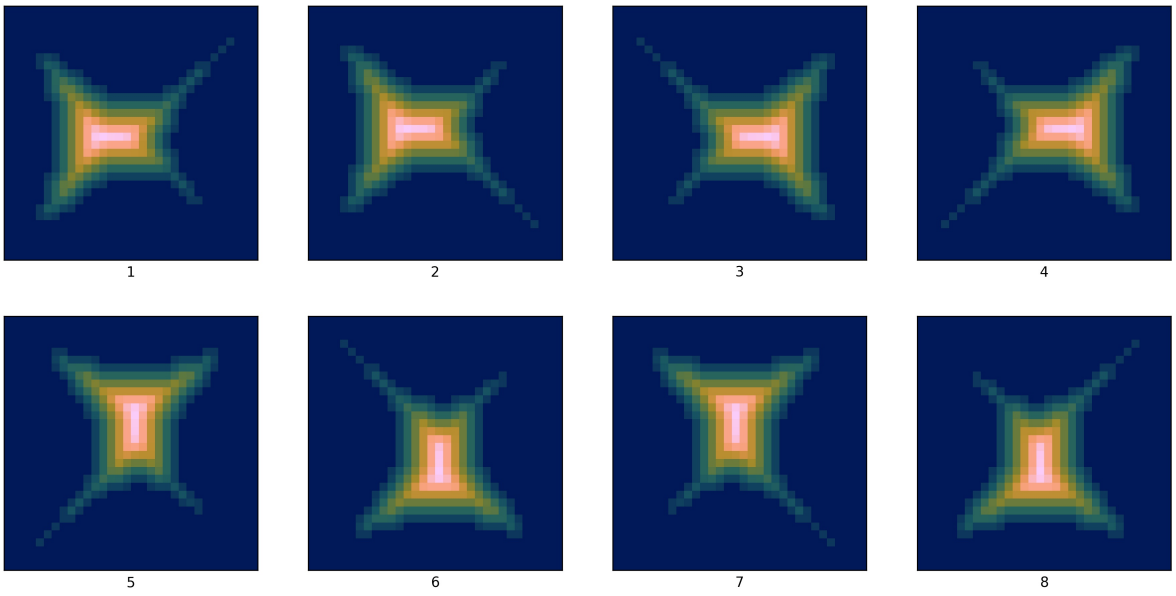


Figure 5.10: Rotating SDF images to increase dataset size.

80% of the dataset was used for the training and the rest of the dataset (called the validation dataset) was used to assess the accuracy of the trained neural network. The training progress is illustrated by fig. 5.11. Reiterating, an epoch is an iteration over the entire training dataset. The weights and biases are recalculated every iteration using the Adagrad algorithm. During an epoch, the losses after each iteration is stored, averaged at the end of the epoch and then plotted as the training loss (blue line). At the end of every epoch, the CNN predicts a value for every input in the validation dataset. The absolute error (in percentage) is calculated between the predictions and the ground-truth data (determined using DSMC simulations), averaged, then plotted as validation loss (orange line). Note that the validation dataset is only used for calculating the error and is not used for training the CNN. The mean absolute error on the validation dataset for thrust, specific impulse, power and thrust efficiency was 4.23%, 4.4%,

1.15%, and 2.5% respectively. Training for Thrust, I_{sp} and power separately gives greater freedom in the optimization process in comparison to training directly for thrust efficiency. Say, a designer wants an optimum thrust efficiency but with the constraints that I_{sp} is above 45 s or maybe power requirement below 3 W. In the genetic algorithm, a design not satisfying these constraints will be then assigned the fitness value of 0. When optimizing for maximizing thrust efficiency, the fitness value = thrust efficiency value of a design.

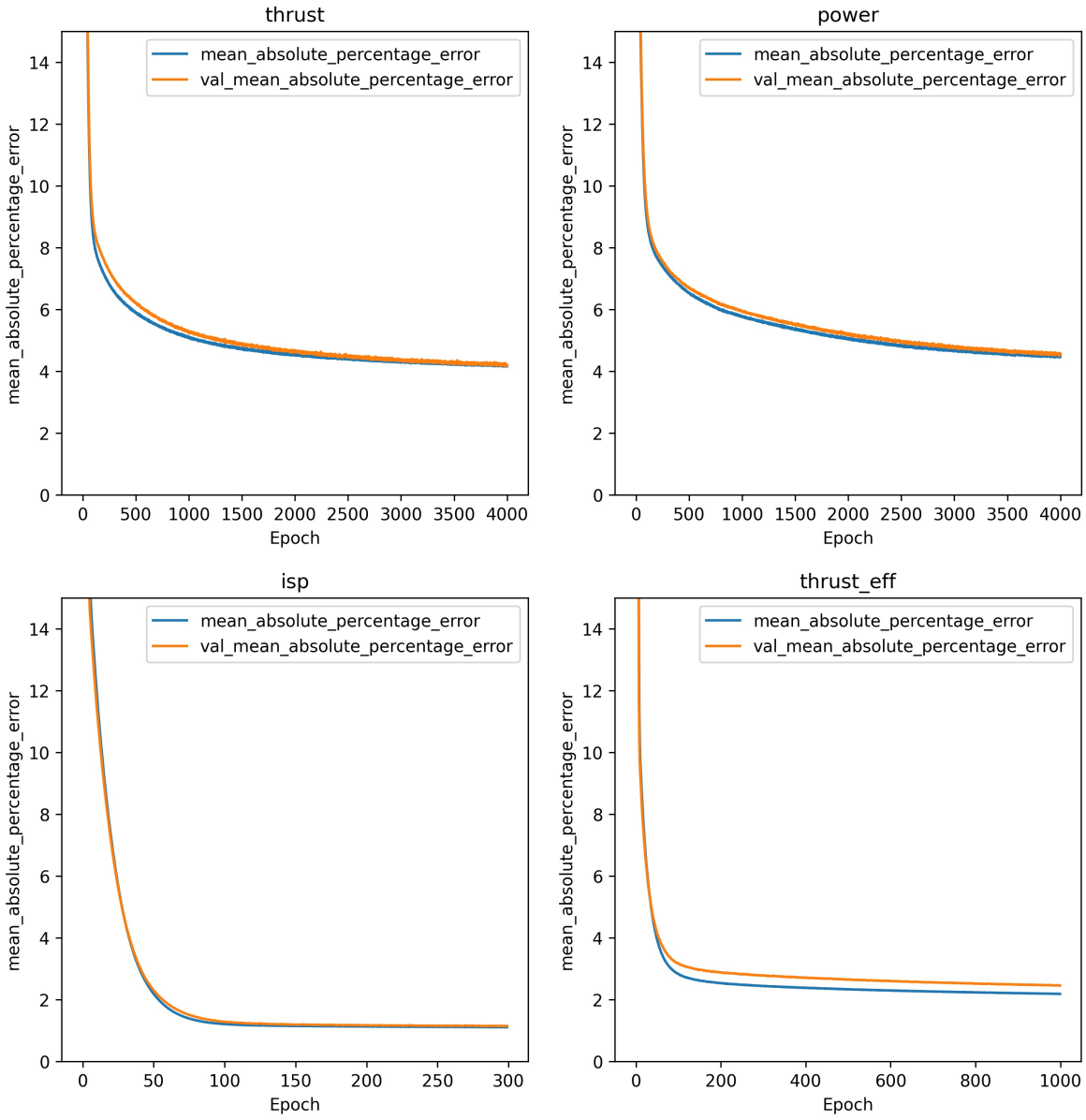


Figure 5.11: Training progress of the CNN for predicting thrust, specific impulse, power requirement and thrust efficiency.

The question was now whether the trained CNN1 could recreate the trend in fig. 5.7. For every design in the polar dataset, thrust efficiency was predicted using CNN1 by using eq. 5.4 (using $CNN1_{thrust}$, $CNN1_{isp}$, and $CNN1_{power}$) and also by using $CNN1_{eff}$. Fig. 5.12 shows that

the former method shows a better agreement with the trend than the latter method. $CNN1_{eff}$ predicts that on increasing area, the thrust efficiency is increasing which is not the correct trend. In fact, running the GA algorithm using $CNN1_{eff}$ as the evaluation model resulted in the largest possible area as the optimum design.

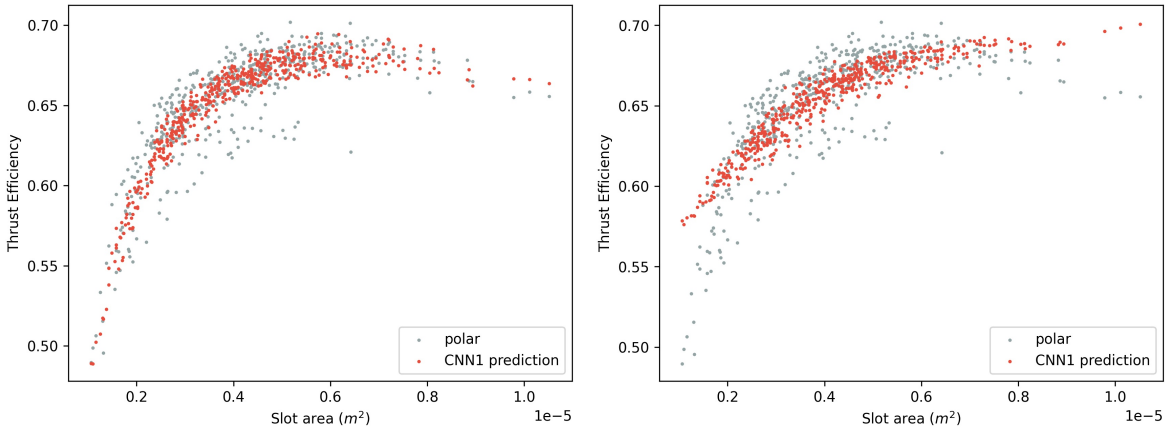


Figure 5.12: Thrust efficiency predictions of the polar dataset using the separately trained thrust, specific impulse and power CNN1 models (left) and using the thrust efficiency CNN1 model (right).

CNN1 was then tested on the rectangular dataset to see if the model has learnt enough to predict the performance of designs that look very different from the polar dataset. Using $CNN1_{thrust}$, $CNN1_{isp}$, and $CNN1_{power}$ and eq. 5.4, fig. 5.13 was plotted. The deviation from the trend is apparent. It seems that CNN1 can only be used for designs similar to the original dataset. Note that if weights are considered as design parameters, then the rectangular dataset is, in-fact, a subset of the polar dataset as both are generated using 8 control points. Increasing the polar dataset might result in a more accurate CNN model.

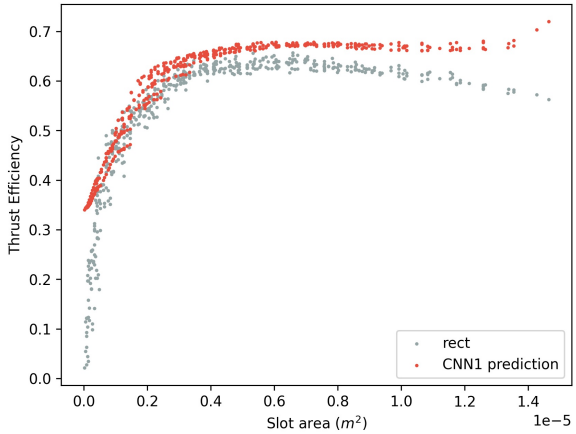


Figure 5.13: Thrust efficiency predictions of the rectangular dataset using the separately trained thrust, specific impulse and power CNN1 models.

5.3.2. CNN2: Polar + Rectangular dataset

As a remedy to the problems described in the previous subsection, an experiment was thought of to try and see if the CNN architecture (described in sec. 3.4.1) could be trained on a combination of the polar and rectangular dataset, which are, visually, very distinct from each other. The trained CNN has been termed CNN2. In this experiment, the CNN was trained for just the thrust efficiency for 10000 epochs. The mean absolute error on the validation dataset was 9.782%. The large gap between the training and the validation dataset indicates that the samples in the training dataset are unrepresentative of the design space which essentially means that the training dataset is not large enough to learn the problem and the data must be increased to get better accuracy.

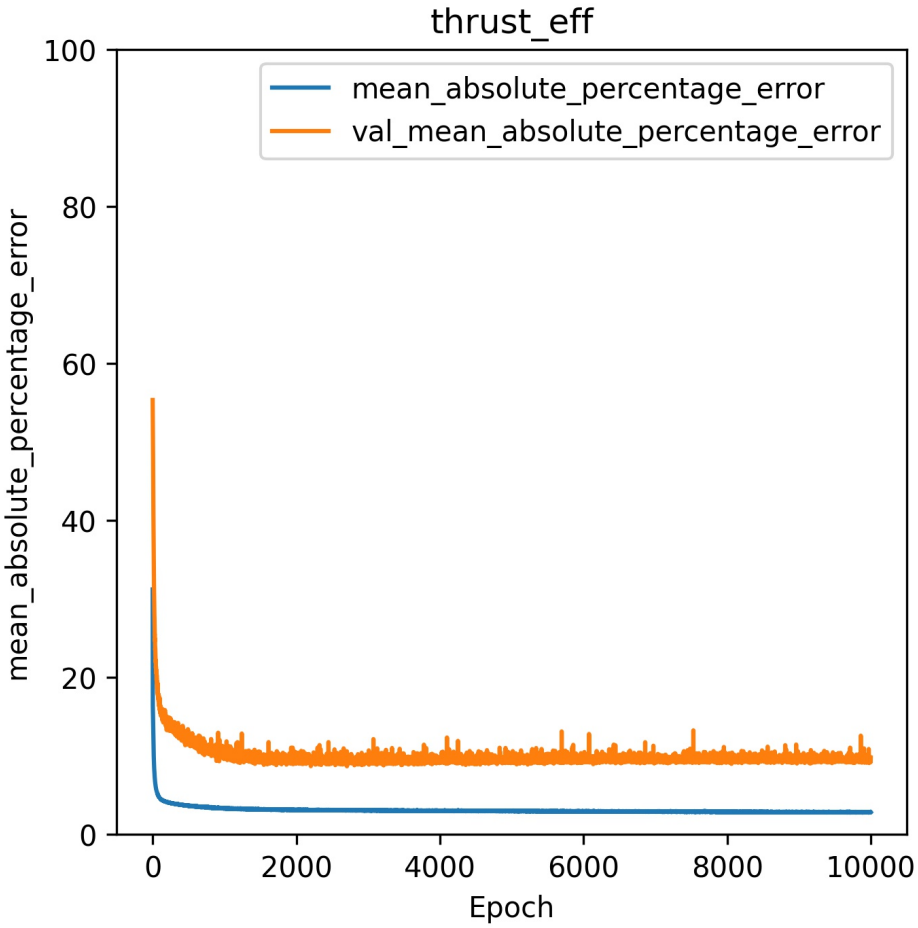


Figure 5.14: Training progress of the CNN for predicting the thrust efficiency.

Similar to the previous subsection, CNN2 was used to predict values of the polar and rectangular dataset. The trends in both cases are well reproduced and hence CNN2 is possibly better than CNN1.

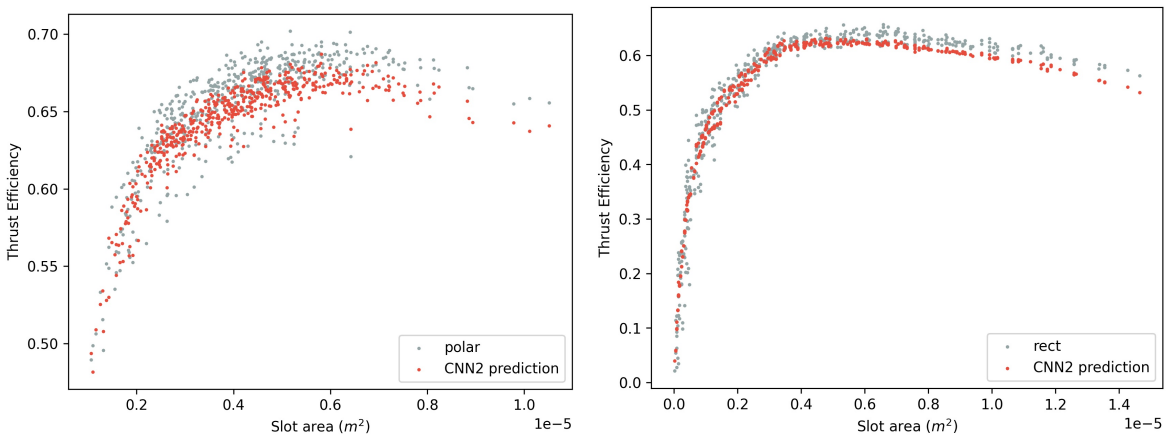


Figure 5.15: Thrust efficiency predictions of the polar dataset (left) and the rectangular dataset (right) using CNN2.

5.4. GA results

Using CNN1 and CNN2 trained on the polar dataset (sec. 5.3.1) and the polar + rectangular dataset (sec. 5.3.2) respectively, as the response surface, the GA architecture (described in sec. 3.5.1) was executed. With the average error of CNN1 and CNN2 around 10% on their respective validation datasets, the GA optimum designs were found to be, in fact, sub-optimal when simulated using DSMC. This means that the CNN accuracy is not high enough for the GA to come up with an optimum design. With a bigger dataset, the CNN model could be improved to a low enough error % for the GA to be functional. The results of the GA runs using CNN1 and CNN2 as the response surface is covered in the following subsections. Note that the best design is still fig. 5.8 as the "optimum" designs in this section are optimum only when using the CNN response surface. For the inlet pressure of 50 Pa, on observing fig. 5.7, it is observed that the particular shape of the slot matters less than the area of the slot. If similar pattern is observed for a different set of boundary condition in a DoE run, then the creation of the response surface and a GA run could be skipped which are time consuming, computationally expensive, and may only give a slight boost in performance. Using a response surface and an optimization method only becomes necessary if this trend is not observed as it indicates that the shape of the slot has more of an influence than the area of the slot.

5.4.1. GA optimization using CNN1

Using CNN1 as the response surface and 8 design variables the GA run resulted in fig. 5.16 as the optimized figure with a predicted thrust efficiency of 0.7320. However, the prediction is off by 11.2% from the DSMC simulation of the design. Simulation of the design results in a thrust of 0.017373 N, I_{sp} of 47.301 s, power requirement of 6.1212 W and thrust efficiency of 0.6582. The calculations assume a 100 microchannel heaterchip. Note that thrust and power linearly scale with the number of microchannels.

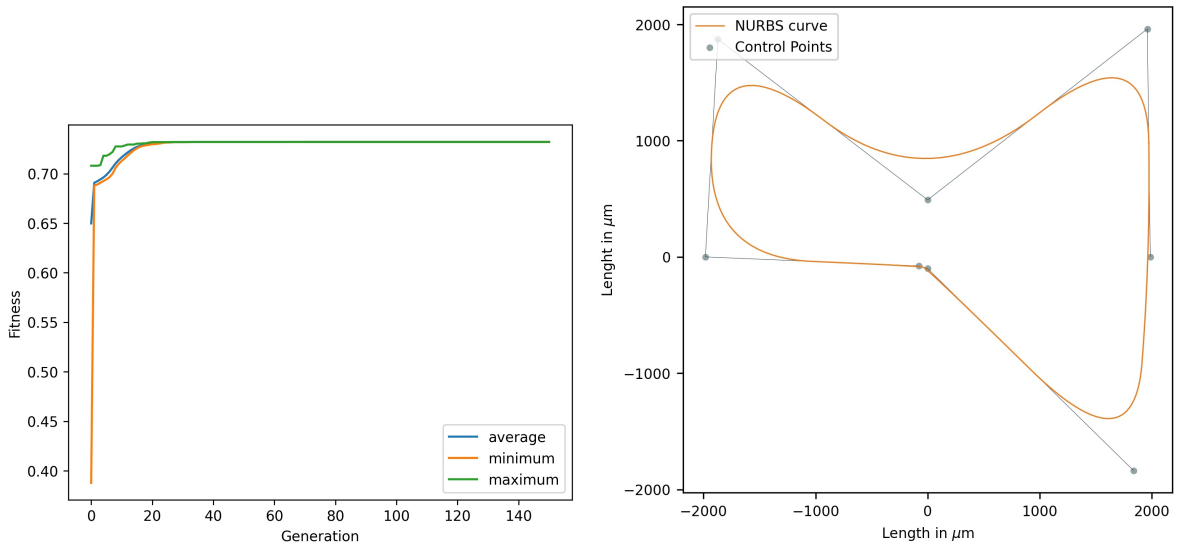


Figure 5.16: Thrust efficiency values through the iterations of the GA run (left) and the most efficient design at the end of the iterations (right).

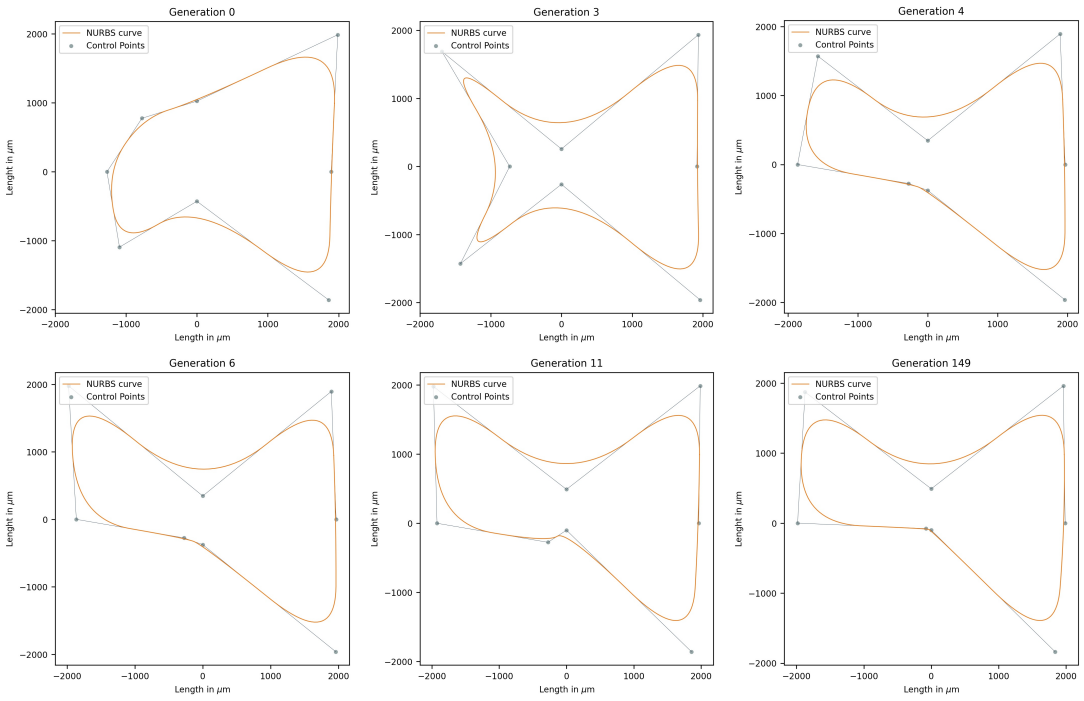


Figure 5.17: Best designs through the generations.

5.4.2. GA optimization using CNN2

Using CNN2 as the response surface and 8 design variables the GA run resulted in fig. 5.18 as the optimized figure with a predicted thrust efficiency of 0.7281. DSMC simulation of fig. 5.18 results in a thrust of 0.01714 N, I_{sp} of 47.166 s, Power requirement of 6.065 W and thrust efficiency of 0.6538. The predicted and simulated thrust efficiency differs by 11.36%.

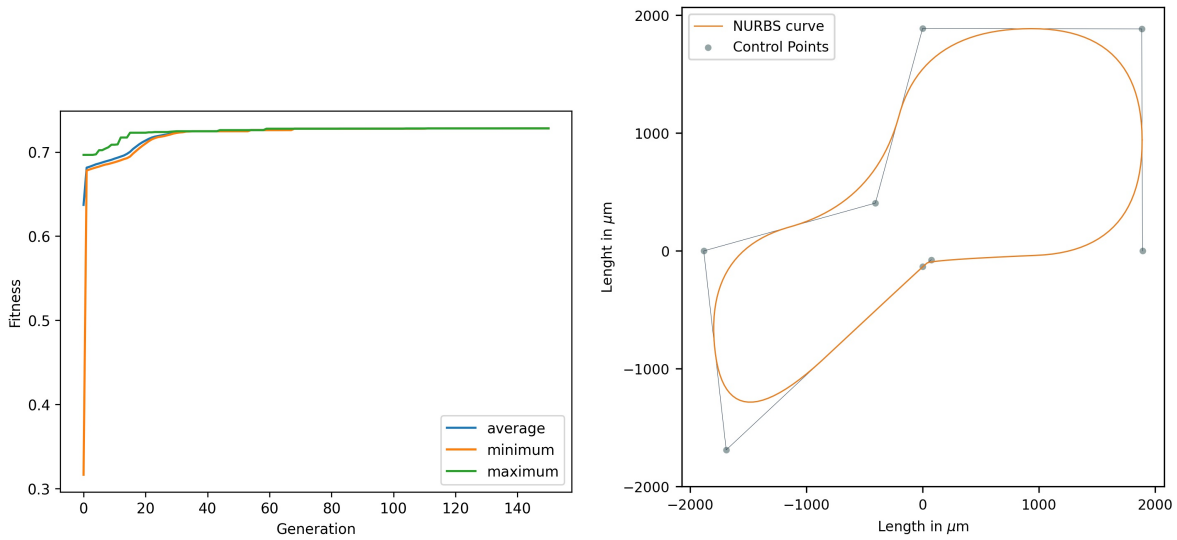


Figure 5.18: Thrust efficiency values through the iterations of the GA run (left) and the most efficient design at the end of the iterations (right).

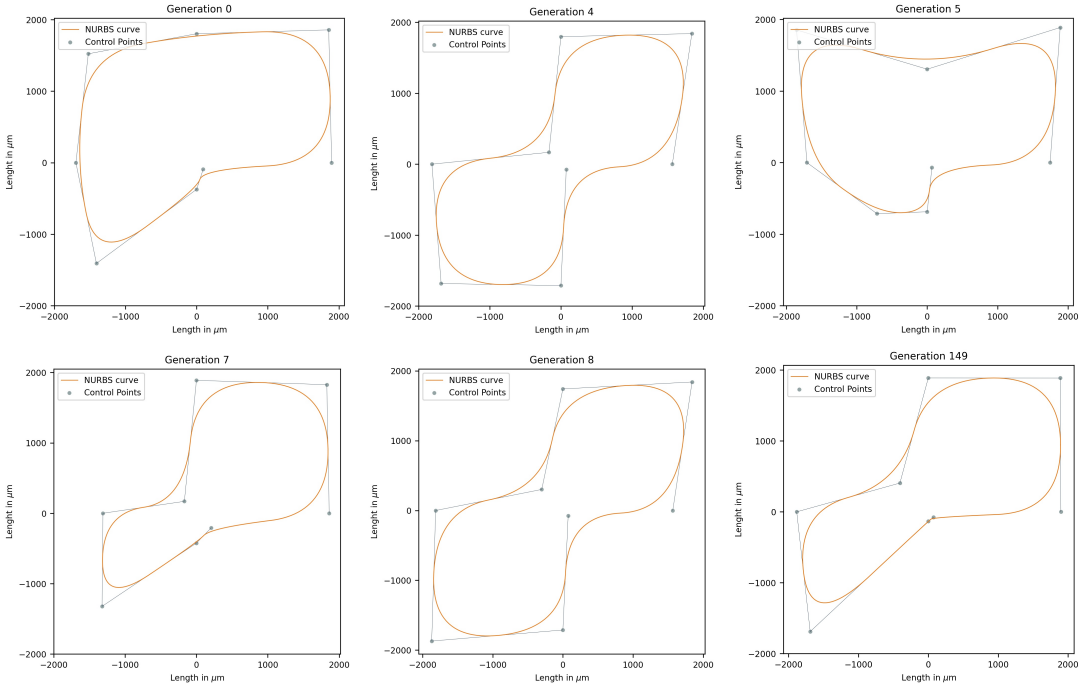


Figure 5.19: Best designs through the generations.

5.4.3. GA optimization using CNN2: Higher control points

In this optimization problem, the number of design variables was increased to 15 i.e. the geometry is controlled by 15 control points. Note that increasing control points enable the creation of finer features that may not be captured in the raster image of 32 x 32 px. The number of pixels can be increased but that would require changes to the CNN architecture so that it

could accept, for example, 256 x 256 px images, which can capture features of, at the very least, a 30 control point figure. Using CNN2, the GA resulted in fig. 5.20 as the optimized design with a thrust efficiency of 0.7455. DSMC simulation results in a thrust of 0.0218 N, I_{sp} of 46.5412 s, power requirement of 7.7067 W and thrust efficiency of 0.6453. The predicted and simulated thrust efficiency differs by 15.53%.

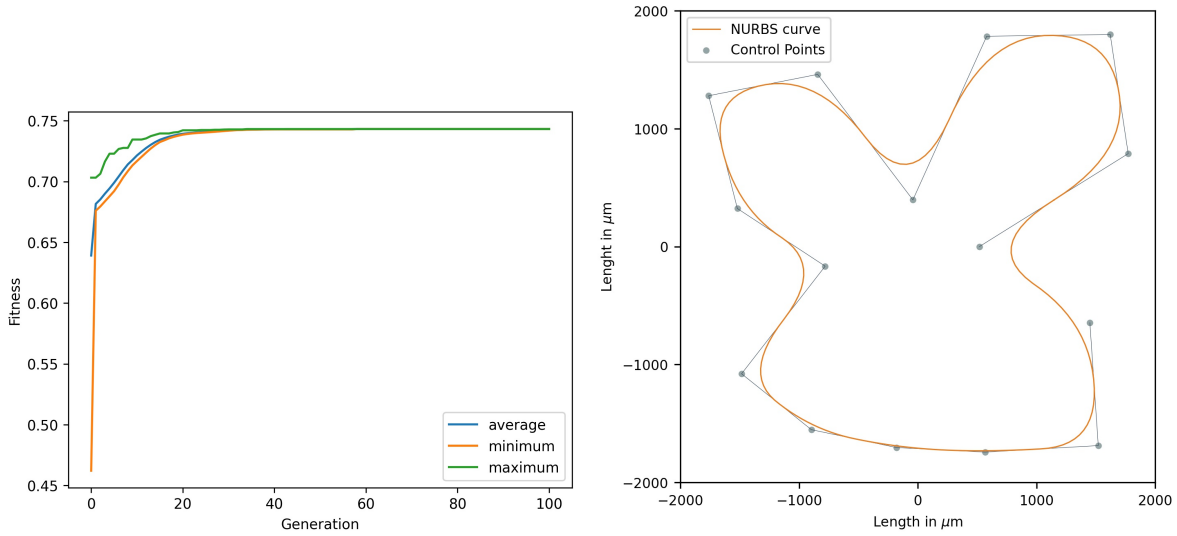


Figure 5.20: Thrust efficiency values through the iterations of the GA run (left) and the most efficient design at the end of the iterations (right).

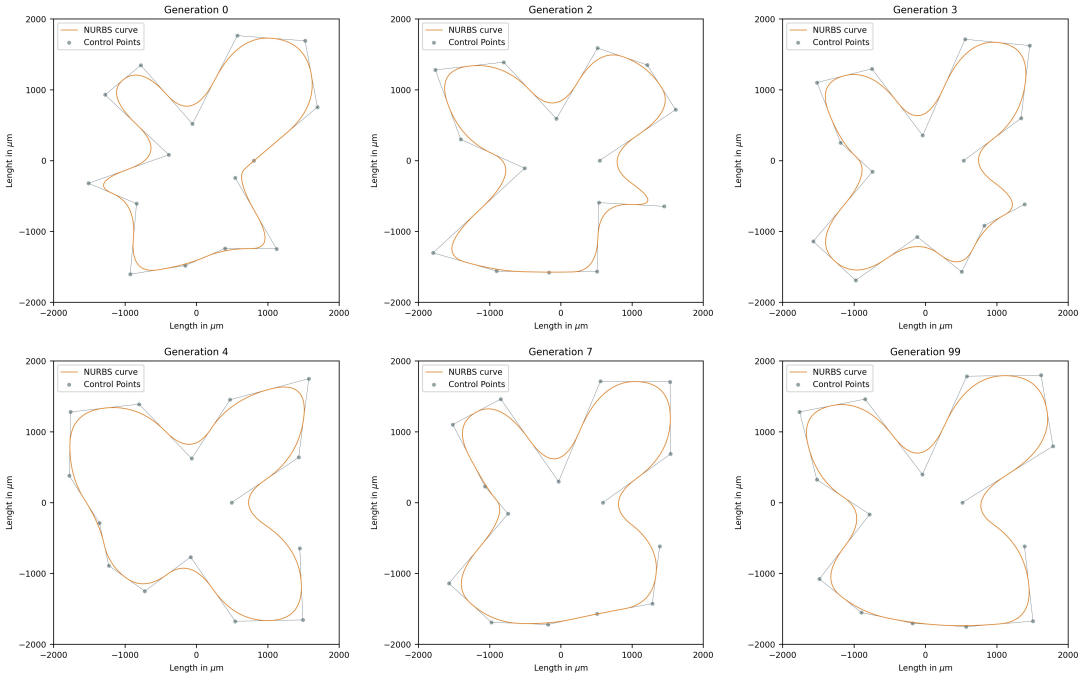


Figure 5.21: Best designs through the generations.

6

Conclusion

In this thesis, a response surface based method was introduced to optimize the design of the cross-sectional shape of a microchannel of a Low-Pressure Microresistojet (LPM). Usage of Response Surface Methodology (RSM) to optimize the design of a microchannel for greater thrust efficiency requires the following steps: Design of Experiment (DoE) run to select samples from design space, performing experiments to evaluate the performance of selected samples, building a response surface (or surrogate model) from the generated data and finally using an optimization algorithm to find a high-efficiency design. In this thesis, these steps were performed using (respectively): Sobol Sequences, Direct Simulation Monte Carlo (DSMC) method using SPARC Industries' VSTRAP software, Convolutional Neural Network (CNN) and Genetic Algorithm (GA).

As a case study, the optimization was performed for Argon as the propellant, inlet pressure of 50 Pa, inlet temperature of 300 K and microchannel wall temperature of 600 K. The number of design parameters were set as 8 though the base geometry supports higher numbers as well. Using Sobol sequences to determine parametric values, 512 designs (termed polar designs) were generated. After DSMC simulations (validated with an experimental study in literature), the best performing design was found to have a thrust efficiency of 0.7019. 512 Rectangular designs were generated as well among which the best performing design had a thrust efficiency of 0.6566. From this dataset, the main takeaway is that the thrust efficiency peaks at a certain slot area and that small areal rectangular slots, common in literature, are not as efficient as broader slots with higher cross-sectional area (up to a limit). The CNN was trained on the polar dataset (CNN1) as well as the combination of both datasets (CNN2) to predict thrust efficiency from slot images. The predictions were, on average, within 10% from the simulation results. A GA run with 8-parameter geometry and with CNN1 as the response surface resulted in an optimum design with a thrust efficiency of 0.732. With CNN2, the optimized efficiency for 8-parameter and 15-parameter geometry was 0.7281 and 0.7455 respectively. However, on simulating the GA results it was found that CNN1 and CNN2 had overestimated the thrust

efficiency by $\approx 11\%$. Thus in this case study, the most thrust efficient design is the best one from the polar dataset. The conclusion drawn is that a high number of designs generated using Sobol Sequences are enough to find the best performing design for a given set of input conditions i.e. choice of propellant, inlet pressure and temperature, and heaterchip temperature. As it seems that there is a strong correlation between slot area and performance, the number of designs to be generated can be based on whether it spans the allowable minimum to the allowable maximum area set by the designer. However, if the correlation between slot area and performance is not found after a DoE run for a different set of input conditions, then a CNN would have to be trained and used by the GA for design optimization.

The thesis addressed the following research questions that were set at the start of the thesis:

1. **By how much can the performance (thrust, power requirement and specific impulse) of an LPM system be increased by solely (i.e. with fixed boundary conditions) optimizing the MHX geometry?**

   From the DSMC simulations done on a variety of rectangular shapes spanning the entire allowable areal range, thrust varies from $5.1916 \times 10^{-}5$ to 0.0453 N, I_{sp} ranges varies from 32.86 to 51.22 s, power requirement varies from 0.3914 to 16.9434 W, and thrust efficiency varies from 0.02132 to 0.656201. From the simulations, it is clear that the area of the slot matters more than the exact shape of the slot, at least for the input conditions used in the case study. For higher pressures, the shape may matter more as the effect of the boundary layer becomes more pronounced.

   (a) **Can the GA method supported by a neural network surrogate of a DSMC solver be used to optimize the design of the MHX geometry?**
   Yes, however, as observed in this thesis, it depends on the accuracy of the CNN. Using the CNN as the response surface, GA runs resulted in higher thrust efficient designs which were better than the designs in the DoE run. However, on simulating these designs with the DSMC method, it was found that the CNN has overestimated the performance of these designs by about 11% which meant that the most thrust efficient design in the polar dataset was not surpassed by the GA optimization owing to the inaccuracy of the CNN models. A bigger dataset might be needed to boost the accuracy of the CNN. As a ballpark figure, a CNN with an average error of around 3% will be more suited as a response surface. To do this, changes can be made to the architecture and the dataset can be increased as well. As several architectures were tried in this thesis, the recommendation is to increase the dataset.

   (b) **What should the architecture of the CNN method look like such that it efficiently (takes less time to train and is $> 90\%$ accurate) maps an image of the MHX geometry and input parametric values to the performance of the MHX?**
   On experimenting with several architectures, a 5-layer deep convolutional layer and 2-layer deep fully connected layer with max-pooling layers and ReLU activation function was found to be most suited to the task.

(c) **How close are the results by the CNN to the DSMC results?**

The mean absolute error on the validation dataset is $< 10\%$.

(d) **Is a CNN model a sufficient replacement of a DSMC solver for comparing the performance of MHX designs?**

A CNN model can be a sufficient replacement for a DSMC solver provided that the parameter space is uniformly and extensively explored by the dataset.

(e) **How large should the dataset (composed of MHX designs and their DSMC-derived performance values) used for training the CNN be such that the CNN is $> 90\%$ accurate?**

A dataset of 512×8 (increased 8-fold by rotating the designs) designs and their corresponding performance values was enough for the CNN to be $> 90\%$ accurate.

2. **How close is VSTRAP's results to the experimental results of LPM systems documented in literature?**

DSMC simulations using VSTRAP compared favourably with the experimental results. The maximum error for thrust and specific impulse was 3.28% and 0.00% respectively.

7

Recommendations

As the conclusion states, it is enough to explore the design space solely using Sobol sequences provided that the minimum and maximum area of the slot designs in the dataset is close to the allowable minimum and allowable maximum area set by the designer. One should build a response surface and proceed with design optimization only if one requires fast results to answer 'what if?' questions or if the correlation between slot area and thruster performance was not observed for a particular set of input conditions.

A response based methodology was implemented for this thesis and was executed for a specific input boundary condition: Argon propellant, inlet pressure = 50 Pa, inlet temperature = 300 K and heaterchip temperature = 600 K. The response surface hence only works for this set of input conditions. To get around this limitation, the CNN architecture would have to be changed such that along with SDF, the input conditions are also part of the dataset. It could be done by assigning a few neurons from the dense layer of the architecture to the input conditions as shown in fig. 3.11. Of course, this would require a bigger dataset with the performance of designs evaluated at various input conditions such that the neural net understands the link between the input conditions and the performance of the thruster.

The mean error for predicting thrust efficiency sits around 10% for the trained CNN model. Improving the accuracy of the prediction can be done by trying out different architectures, and by increasing the dataset size. This dataset could include the following features that were not considered in this thesis:

1. Different base geometry: A new base geometry could have different rules governing the motion of control points based on parametric values.
2. Include 'weights' (associated with each control point) and the curve 'degree' as design parameters. In this thesis, all weights were set as 1 and the curve degree was set as 2. Only control points were treated as design parameters.

3. Recording energy gained by the particles colliding with the slot walls every timestep will provide a better estimate of the power required by the heaterchip to maintain its temperature. Currently, an analytical equation is being used that relates mass flow rate to power requirement, which is an overestimation.

Heat transfer by radiation has not considered in this thesis or in preceding related works dealing with LPM systems, however, according to Frohn, Roth, and Anders 2010, the influence of heat transfer by radiation increases with increasing rarefaction and in high-vacuum, radiation becomes the dominant type of energy transfer. Future studies should explore if the effect of radiation heat transfer between slot walls and the fluid is low enough to be ignored.

Furthermore, experimental studies need to be carried out with different propellants at different boundary conditions for validating simulations and also testing out the optimized geometry.

References

Aggarwal, Charu C. 2018. *Neural Networks and Deep Learning*. Cham: Springer International Publishing. ISBN: 978-3-319-94462-3. https://doi.org/10.1007/978-3-319-94463-0. http://link.springer.com/10.1007/978-3-319-94463-0.

Alihosseini, Yousef, Mohammad Zabetian Targhi, Mohammad Mahdi Heyhat, and Nima Ghorbani. 2020. "Effect of a micro heat sink geometric design on thermo-hydraulic performance: A review." *Applied Thermal Engineering* 170 (April). ISSN: 13594311. https://doi.org/10.1016/j.applthermaleng.2020.114974.

Andric, Nemanja, Daniel W. Meyer, and Patrick Jenny. 2019. "Data-based modeling of gas-surface interaction in rarefied gas flow simulations." *Physics of Fluids* 31, no. 6 (June). ISSN: 10897666. https://doi.org/10.1063/1.5094768.

Baganoff, D., and J. D. McDonald. 1990. "A collision-selection rule for a particle simulation method suited to vector computers." *Physics of Fluids A* 2 (7): 1248–1259. ISSN: 08998213. https://doi.org/10.1063/1.857625.

Baodong, Shao, Wang Lifeng, Li Jianyun, and Cheng Heming. 2011. "Multi-objective optimization design of a micro-channel heat sink using adaptive genetic algorithm." *International Journal of Numerical Methods for Heat and Fluid Flow* 21 (3): 353–364. ISSN: 09615539. https://doi.org/10.1108/09615531111108512.

Bhatnagar, P. L., E. P. Gross, and M. Krook. 1954. "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems." *Physical Review* 94, no. 3 (May): 511–525. ISSN: 0031-899X. https://doi.org/10.1103/PhysRev.94.511. https://link.aps.org/doi/10.1103/PhysRev.94.511.

Bijster, R.J.F. 2014. "Design, Verification and Validation of a Micropropulsion Thrust Stand." PhD diss., Delft University of Technology. http://resolver.tudelft.nl/uuid:5c3f0eb3-511c-4443-90b2-46eed9a6b184.

Binder, T., P. C. Boldini, F. Romano, G. Herdrich, and S. Fasoulas. 2016. "Transmission probabilities of rarefied flows in the application of atmosphere-breathing electric propulsion." *AIP Conference Proceedings* 1786 (November 2016). ISSN: 15517616. https://doi.org/10.1063/1.4967689.

Bingol, Onur R. n.d. "Python library for NURBS." Accessed May 5, 2021. https://nurbs-python.readthedocs.io/en/5.x/.

Bird, G A. 1994a. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, chap. 5, 119. Oxford University Press. ISBN: 978-0-19-856195-8. https://app.knovel.com/hotlink/khtml/id:kt0088INWB/molecular-gas-dynamics/molecules-with-rotational.

———. 1970. "Direct simulation and the Boltzmann equation." *Physics of Fluids* 13 (11): 2676–2681. ISSN: 10706631. https://doi.org/10.1063/1.1692849.

———. 1994b. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford University Press. ISBN: 9780198561958.

———. 2005. "The DS2V/3V Program Suite for DSMC Calculations." In *AIP Conference Proceedings*, 762:541–546. June 2005. AIP. ISBN: 0735402477. https://doi.org/10.1063/1.1941592. http://aip.scitation.org/doi/abs/10.1063/1.1941592.

Bouwmeester, J., S. Radu, M. S. Uludag, N. Chronas, S. Speretta, A. Menicucci, and E. K.A. Gill. 2020. "Utility and constraints of PocketQubes." *CEAS Space Journal*, ISSN: 18682510. https://doi.org/10.1007/s12567-020-00300-0. https://doi.org/10.1007/s12567-020-00300-0.

Bratley, Paul, and Bennett L. Fox. 1988. "Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator." *ACM Transactions on Mathematical Software (TOMS)* 14 (1): 88–100. ISSN: 15577295. https://doi.org/10.1145/42288.214372.

Cambridge Dictionary. n.d. "Optimization." Accessed April 27, 2021. https://dictionary.cambridge.org/dictionary/english/optimization.

Cavazzuti, Marco. 2013. *Optimization Methods: From Theory to Design*. 272. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-31186-4. https://doi.org/10.1007/978-3-642-31187-1. http://link.springer.com/10.1007/978-3-642-31187-1.

Cercignani, C., and M. Lampis. 1971. "Kinetic models for gas—surface interactions." *Transport Theory and Statistical Physics* 1 (2): 101–114. ISSN: 15322424. https://doi.org/10.1080/00411457108231440.

Chandra, Anant. 2021a. *DSMC simulation results for Polar and rectangular microchannel designs for the LPM propulsion system*. https://doi.org/10.4121/14723472.v1.

———. 2021b. *Trained CNN model to predict performance of an LPM thruster from the 32 x 32 px image of its microchannel cross-section*. https://doi.org/10.4121/14740353.v1.

Charpentier, Éric, Annick Lesne, and Nikolaï K. Nikolski, eds. 2007. *Kolmogorov's Heritage in Mathematics*. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-36349-1. https://doi.org/10.1007/978-3-540-36351-4. http://link.springer.com/10.1007/978-3-540-36351-4.

Chen, Yongping, and Ping Cheng. 2002. "Heat transfer and pressure drop in fractal tree-like microchannel nets." *International Journal of Heat and Mass Transfer* 45, no. 13 (June): 2643–2648. ISSN: 00179310. https://doi.org/10.1016/S0017-9310(02)00013-3.

Colin, Stéphane. 2013. "Single-Phase Gas Flow in Microchannels." *Heat Transfer and Fluid Flow in Minichannels and Microchannels*, 11–102. https://doi.org/10.1016/B978-0-08-098346-2.00002-8.

De Boer, Meint J., J. G.E. Gardeniers, Henri V. Jansen, Edwin Smulders, Melis Jan Gilde, Gerard Roelofs, Jay N. Sasserath, and Miko Elwenspoek. 2002. "Guidelines for etching silicon MEMS structures using fluorine high-density plasmas at cryogenic temperatures." *Journal of Microelectromechanical Systems* 11 (4): 385–401. ISSN: 10577157. https://doi.org/10.1109/JMEMS.2002.800928.

Didominic, Dominic, Emma Gist, Jonathan Fitzgerald, and Mark N. Glauser. 2020. "Complex nozzle optimization techniques using machine learning." *AIAA Scitech 2020 Forum* 1 PartF (January): 1–12. https://doi.org/10.2514/6.2020-1866.

Dietrich, Stefan, and Iain D Boyd. 1996. *Scalar and Parallel Optimized Implementation of the Direct Simulation Monte Carlo Method.* Technical report.

Duchi, John C., Peter L. Bartlett, and Martin J. Wainwright. 2012. "Randomized smoothing for (parallel) stochastic optimization." *Proceedings of the IEEE Conference on Decision and Control* 12:5442–5444. ISSN: 01912216. https://doi.org/10.1109/CDC.2012.6426698.

Fang, Xiaopeng. 2007. "Engineering design using genetic algorithms." PhD diss., Iowa State University. https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=16942%7B%5C&%7Dcontext=rtd.

Fattahi, Mehdi, Kourosh Vaferi, Mohammad Vajdi, Farhad Sadegh Moghanlou, Abbas Sabahi Namini, and Mehdi Shahedi Asl. 2020. "Aluminum nitride as an alternative ceramic for fabrication of microchannel heat exchangers: A numerical study." *Ceramics International* 46 (8): 11647–11657. ISSN: 02728842. https://doi.org/10.1016/j.ceramint.2020.01.195. https://doi.org/10.1016/j.ceramint.2020.01.195.

Finger, George Wayne. 2005. "Estimation of Tangential Momentum Accommodation Coefficient Using Molecular Dynamics Simulation." PhD diss., University of Central Florida. https://stars.library.ucf.edu/cgi/viewcontent.cgi?article1552&contextetd.

Frohn, Arnold, Norbert Roth, and Klaus Anders. 2010. "Heat Transfer and Momentum Flux in Rarefied Gases." Chap. M10 in *VDI Heat Atlas.* Stuttgart, Germany: Springer-Verlag Berlin Heidelberg. ISBN: 978-3-540-79999-3. https://www.springer.com/de/book/9783540778769.

Furtney, Jason. n.d. "Signed Distance Function using the Fast Marching Method." Accessed May 5, 2021. https://github.com/scikit-fmm/scikit-fmm.

Gao, Da, Chonglin Zhang, and Thomas E. Schwartzentruber. 2011. "Particle simulations of planetary probe flows employing automated mesh refinement." *Journal of Spacecraft and Rockets* 48 (3): 397–405. ISSN: 15336794. https://doi.org/10.2514/1.52129.

"Gmsh." n.d. Accessed May 11, 2021. https://gmsh.info/.

Guerrieri, Daduí C, Marsil de Athayde Costa e Silva, Barry Zandbergen, and Angelo Cervone. 2016. "Heater Chip with Different Microchannels Geometries for a Low Pressure Free Molecular Micro-Resistojet." In *Space propulsion 2016*, 1–5. TU Delft. https://repository.tudelft.nl/islandora/object/uuid%7B%5C%%7D3A00545557-b420-4634-a143-399434da899b.

Guerrieri, Daduí C, Marsil A.C. Silva, Henk van Zeijl, Angelo Cervone, and Eberhard Gill. 2017. "Fabrication and characterization of low pressure micro-resistojets with integrated heater and temperature measurement." *Journal of Micromechanics and Microengineering* 27, no. 12 (October). ISSN: 13616439. https://doi.org/10.1088/1361-6439/aa90fb.

Guerrieri, Daduí C., Marsil A.C. Silva, Angelo Cervone, and Eberhard Gill. 2018. "An analytical model for characterizing the thrust performance of a Low-Pressure Micro-Resistojet." *Acta Astronautica* 152 (November): 719–726. ISSN: 00945765. https://doi.org/10.1016/j.actaastro.2018.09.008.

Guo, Xiaoxiao, Wei Li, and Francesco Iorio. 2016. "Convolutional neural networks for steady flow approximation." In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Augu, 481–490. Association for Computing Machinery, August. ISBN: 9781450342322. https://doi.org/10.1145/2939672.2939738.

Hadjiconstantinou, Nicolas G. 2006. "The limits of Navier-Stokes theory and kinetic extensions for describing small-scale gaseous hydrodynamics." 18 (11). ISSN: 10706631. https://doi.org/10.1063/1.2393436.

Hall, P. M. 1968. "The effect of expansion mismatch on temperature coefficient of resistance of thin films." *Applied Physics Letters* 12 (6): 212. ISSN: 00036951. https://doi.org/10.1063/1.1651956.

Hamidnia, Mohammad, Yi Luo, Xiaodong Wang, and Gang Jing. 2018. "Thermal performance optimization of Si micro flat heat pipes by Box–Behnken design." *Microsystem Technologies* 24, no. 7 (July): 3085–3094. ISSN: 09467076. https://doi.org/10.1007/s00542-018-3829-y.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. "Multilayer feedforward networks are universal approximators." *Neural Networks* 2 (5): 359–366. ISSN: 08936080. https://doi.org/10.1016/0893-6080(89)90020-8.

Hui, Xinyu, Junqiang Bai, Hui Wang, and Yang Zhang. 2020. "Fast pressure distribution prediction of airfoils using deep learning." *Aerospace Science and Technology* 105:105949. ISSN: 12709638. https://doi.org/10.1016/j.ast.2020.105949. https://doi.org/10.1016/j.ast.2020.105949.

Ighalo, F. U., T. Bello-Ochende, and J. P. Meyer. 2009. "Mathematical optimization: Application to the design of optimal micro-channel heat sinks." In *Proceedings - 2009 3rd Southern Conference on Computational Modeling, MCSUL 2009*, 1–6. ISBN: 9780769539768. https://doi.org/10.1109/mcsul.2009.8.

Ivanov, M. S., A. V. Kashkovsky, P. V. Vashchenkov, and Y. A. Bondar. 2011. "Parallel Object-Oriented software system for DSMC modeling of high-altitude aerothermodynamic problems." *AIP Conference Proceedings* 1333 (PART 1): 211–218. ISSN: 0094243X. https://doi.org/10.1063/1.3562650.

Ivanov, M. S., G. N. Markelov, and S. F. Gimelshein. 1998. "Statistical simulation of reactive rarefied flows: Numerical approach and applications." *7th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, https://doi.org/10.2514/6.1998-2669.

James, F. 1990. "A review of pseudorandom number generators." *Computer Physics Communications* 60 (3): 329–344. ISSN: 00104655. https://doi.org/10.1016/0010-4655(90)90032-V.

Jansen, E.H.W. 2016. "Improvement and validation of test stand performance for novel micropropulsion systems." PhD diss., TU Delft. http://resolver.tudelft.nl/uuid:28f0b7fa-1288-4e32-88fd-2e138745714a.

Jiang, Jui-Chin, Cheng-Hsing Hsu, Thi-Anh-Tuyet Nguyen, and Hung-Son Dang. 2017. "Applied System Innovation for Modern Technology : proceedings of the 2017 IEEE International Conference on Applied System Innovation (IEEE-ICASI 2017) : Sapporo, Japan, 13-17 May 2017." In *IEEE International Conference on Applied System Innovation*. ISBN: 9781509048977.

Kaiming, He, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. 2016. "Deep Residual Learning for Image Recognition." https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Kargaran, H., A. Minuchehr, and A. Zolfaghari. 2017. "An improved combinatorial geometry model for arbitrary geometry in DSMC." *Journal of Statistical Mechanics: Theory and Experiment* 2017, no. 3 (March). ISSN: 17425468. https://doi.org/10.1088/1742-5468/aa5d1a.

Ketsdever, Andrew D., Amanda A. Green, E. P. Muntz, and Stephen E. Vargo. 2000. "Fabrication and testing of the free molecule micro- resistojet: Initial results." In *35th Intersociety Energy Conversion Engineering Conference and Exhibit*. https://doi.org/10.2514/6.2000-3672.

Ketsdever, Andrew D., Riki H. Lee, and Taylor C. Lilly. 2005. "Performance testing of a microfabricated propulsion system for nanosatellite applications." *Journal of Micromechanics and Microengineering* 15, no. 12 (December): 2254–2263. ISSN: 09601317. https://doi.org/10.1088/0960-1317/15/12/007.

Ketsdever, Andrew D., Dean C. Wadsworth, and E. P. Muntz. 2001. "Gas-Surface Interaction Model Influence on Predicted Performance of Microelectromechanical System Resistojet." *Journal of Thermophysics and Heat Transfer* 15 (3): 302–307. ISSN: 15336808. https://doi.org/10.2514/2.6626.

Ketsdever, Andrew D., Dean C. Wadsworth, Stephen Vargo, and E. P. Muntz. 1998. "The free molecule micro-Resistojet: An interesting alternative to nozzle expansion." In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*. American Institute of Aeronautics / Astronautics Inc, AIAA. https://doi.org/10.2514/6.1998-3918.

Khan, Asifullah, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. 2020. *A survey of the recent architectures of deep convolutional neural networks*, 53:5455–5516. 8. Springer Netherlands. ISBN: 0123456789. https://doi.org/10.1007/s10462-020-09825-6. arXiv: 1901.06032. https://doi.org/10.1007/s10462-020-09825-6.

Kinefuchi, Ikuya, Yusuke Kotsubo, Kenichi Osuka, Yuta Yoshimoto, Nobuya Miyoshi, Shu Takagi, and Yoichiro Matsumoto. 2017. "Incident energy dependence of the scattering dynamics of water molecules on silicon and graphite surfaces: the effect on tangential

momentum accommodation." *Microfluidics and Nanofluidics* 21, no. 2 (February). ISSN: 16134990. https://doi.org/10.1007/s10404-017-1850-6.

Korkut, Burak, and Deborah A. Levin. 2014. "Three dimensional DSMC-PIC simulations of ion thruster plumes with SUGAR." *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference 2014*, https://doi.org/10.2514/6.2014-3447.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet Classification with Deep Convolutional Neural Networks." *Commun. ACM* (New York, NY, USA) 60, no. 6 (May): 84–90. ISSN: 0001-0782. https://doi.org/10.1145/3065386. https://doi.org/10.1145/3065386.

Kuddusi, Lütfullah. 2007. "Prediction of temperature distribution and Nusselt number in rectangular microchannels at wall slip condition for all versions of constant wall temperature." *International Journal of Thermal Sciences* 46, no. 10 (October): 998–1010. ISSN: 12900729. https://doi.org/10.1016/j.ijthermalsci.2006.12.006.

Kuddusi, Lütfullah, and Edvin Çetegen. 2006. "Prediction of temperature distribution and Nusselt number in rectangular microchannels at wall slip condition for all versions of constant heat flux." *International Journal of Heat and Fluid Flow* 28, no. 4 (August): 777–786. ISSN: 0142727X. https://doi.org/10.1016/j.ijheatfluidflow.2006.09.002.

Kumar, Rakesh, and Arun Kumar Chinnappan. 2017. "Development of a multi-species, parallel, 3D Direct Simulation Monte-Carlo solver for rarefied gas flows." *Computers and Fluids* 159 (December): 204–216. ISSN: 00457930. https://doi.org/10.1016/j.compfluid.2017.10.006.

Kuppusamy, Navin Raja, N. N.N. Ghazali, R. Saidur, and M. E. Niza. 2015. "Optimum design of triangular shaped micro mixer in micro channel heat sink." *International Journal of Heat and Mass Transfer* 91 (August): 52–62. ISSN: 00179310. https://doi.org/10.1016/j.ijheatmasstransfer.2015.07.088.

LeBeau, GJ, and Fe Lumpkin III. 2001. "Application highlights of the DSMC Analysis Code (DAC) software for simulating rarefied flows." *Computer Methods in Applied Mechanics and Engineering* 191, nos. 6-7 (December): 595–609. ISSN: 00457825. https://doi.org/10.1016/S0045-7825(01)00304-8. https://linkinghub.elsevier.com/retrieve/pii/S0045782501003048.

Lee, Man, Man Wong, and Yitshak Zohar. 2002. "Design, fabrication and characterization of an integrated micro heat pipe." *Proceedings of the IEEE Micro Electro Mechanical Systems (MEMS)*, 85–88. https://doi.org/10.1109/memsys.2002.984096.

Liang, Tengfei, Qi Li, and Wenjing Ye. 2018. "A physical-based gas–surface interaction model for rarefied gas flow simulation." *Journal of Computational Physics* 352:105–122. ISSN: 10902716. https://doi.org/10.1016/j.jcp.2017.08.061.

Liao, Meng, Quy Dong To, Céline Léonard, and Vincent Monchiet. 2018. "Non-parametric wall model and methods of identifying boundary conditions for moments in gas flow equations." *Physics of Fluids* 30 (3). ISSN: 10897666. https://doi.org/10.1063/1.5016278. http://dx.doi.org/10.1063/1.5016278.

Luo, Lingai, Yilin Fan, and Daniel Tondeur. 2007. "Heat exchanger: From micro- to multi-scale design optimization." *International Journal of Energy Research* 31, no. 13 (October): 1266–1274. ISSN: 0363907X. https://doi.org/10.1002/er.1298.

Ma, D. D., G. D. Xia, Y. F. Li, Y. T. Jia, and J. Wang. 2016. "Design study of micro heat sink configurations with offset zigzag channel for specific chips geometrics." *Energy Conversion and Management* 127 (November): 160–169. ISSN: 01968904. https://doi.org/10.1016/j.enconman.2016.09.013.

Mancas, A, A Cervone, and B Zandbergen. 2014. "Design, Simulation and Optimisation of a Low-Pressure Micro-Resistojet for SmallSatellite Missions." In *Space propulsion 2014*. Cologne, Germany. http://resolver.tudelft.nl/uuid:31253f90-8e20-4993-96cc-b80b24b9c8c4.

Michalak, Andrew. 2019. "Generative Design Optimization of Thermal Management Systems for High Output Power Electronics." PhD diss., University of Toronto. https://tspace.library.utoronto.ca/bitstream/1807/98176/1/Michalak_Andrew_J_201911_MAS_thesis.pdf.

Misdanitis, Serafeim, and Dimitris Valougeorgis. 2008. "Couette flow with heat transfer in the whole range of the Knudsen number." *Proceedings of the 6th International Conference on Nanochannels, Microchannels, and Minichannels, ICNMM2008* (PART A): 15–22. https://doi.org/10.1115/ICNMM2008-62079.

Muñoz-Paniagua, J., and J. García. 2020. "Aerodynamic drag optimization of a high-speed train." *Journal of Wind Engineering and Industrial Aerodynamics* 204 (July). ISSN: 01676105. https://doi.org/10.1016/j.jweia.2020.104215.

Naqiuddin, Nor Haziq, Lip Huat Saw, Ming Chian Yew, Farazila Yusof, Tan Ching Ng, and Ming Kun Yew. 2018. *Overview of micro-channel design for high heat flux application*. https://doi.org/10.1016/j.rser.2017.09.110.

Naris, Steryios, and Dimitris Valougeorgis. 2008. "Rarefied gas flow in a triangular duct based on a boundary fitted lattice." *European Journal of Mechanics - B/Fluids* 27, no. 6 (November): 810–822. ISSN: 09977546. https://doi.org/10.1016/j.euromechflu.2008.01.002. https://linkinghub.elsevier.com/retrieve/pii/S0997754608000137.

Niederreiter, Harald. 1988. "Low-discrepancy and low-dispersion sequences." *Journal of Number Theory* 30, no. 1 (September): 51–70. ISSN: 0022314X. https://doi.org/10.1016/0022-314X(88)90025-X. https://linkinghub.elsevier.com/retrieve/pii/0022314X8890025X.

numpy. n.d. "Uniform random." Accessed April 28, 2021. https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html.

Okutani, Chihiro, Tomoyuki Yokota, Ryotaro Matsukawa, and Takao Someya. 2020. "Suppressing the negative temperature coefficient effect of resistance in polymer composites with positive temperature coefficients of resistance by coating with parylene." *Journal of Materials Chemistry C* 8 (22): 7304–7308. ISSN: 20507526. https://doi.org/10.1039/d0tc00702a.

Olson, Spencer E., and Andrew J. Christlieb. 2008. "Gridless DSMC." *Journal of Computational Physics* 227, no. 17 (September): 8035–8064. ISSN: 00219991. https://doi.org/10.1016/j.jcp.2008.04.038. https://linkinghub.elsevier.com/retrieve/pii/S002199910800257X.

OpenFOAM. n.d. "dsmcFOAM manual page." Accessed November 10, 2020. https://www.openfoam.com/documentation/guides/latest/man/dsmcFoam.html.

Pallichadath, V., L. Turmaine, A. Melaika, S. Gelmi, M. Vilella Ramisa, D. Rijlaarsdam, M. A.C. Silva, et al. 2019. "In-orbit micro-propulsion demonstrator for PICO-satellite applications." *Acta Astronautica* 165 (December): 414–423. ISSN: 00945765. https://doi.org/10.1016/j.actaastro.2019.09.004.

Palmer, Kristoffer, Hugo Nguyen, and Greger Thornell. 2013. "Fabrication and evaluation of a free molecule micro-resistojet with thick silicon dioxide insulation and suspension." *Journal of Micromechanics and Microengineering* 23, no. 6 (June). ISSN: 09601317. https://doi.org/10.1088/0960-1317/23/6/065006.

Pamela, Vocale, and Spiga Marco. 2014. "Asymmetrical heating in rarefied flows through circular microchannels." *Journal of Heat Transfer* 136 (9). ISSN: 15288943. https://doi.org/10.1115/1.4027786.

Peddakotla, Sai Abhishek, Kishore K. Kammara, and Rakesh Kumar. 2019. "Molecular dynamics simulation of particle trajectory for the evaluation of surface accommodation coefficients." *Microfluidics and Nanofluidics* 23 (6): 1–17. ISSN: 16134990. https://doi.org/10.1007/s10404-019-2248-4. https://doi.org/10.1007/s10404-019-2248-4.

Piegl, Les, and Wayne Tiller. 1995. *The NURBS Book.* Monographs in Visual Communications. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-97387-1. https://doi.org/10.1007/978-3-642-97385-7. http://link.springer.com/10.1007/978-3-642-97385-7.

random.org. n.d. "Random Numbers from Atmospheric Noise." Accessed April 28, 2021. https://www.random.org/.

Rieffel, Marc A. 1995. "Concurrent Simulations of Plasma Reactors for VLSI Manufacturing." PhD diss., California Institute of Technology. https://authors.library.caltech.edu/26885/2/CS_TR_95_12.pdf.

Scherer, Caio S., and Liliane B. Barichello. 2009. "A heat transfer problem in rarefied gas dynamics." In *Heat Transfer Engineering*, 30:282–291. March. https://doi.org/10.1080/01457630802381822.

Schmidt, Helmut. 1970. "Quantum‐Mechanical Random‐Number Generator." *Journal of Applied Physics* 41, no. 2 (February): 462–468. ISSN: 0021-8979. https://doi.org/10.1063/1.1658698. http://aip.scitation.org/doi/10.1063/1.1658698.

scikit-image. n.d. "Rasterization using scikit-image." Accessed May 5, 2021. https://scikit-image.org/docs/dev/api/skimage.draw.html?highlight=polygon#polygon.

Shao, Baodong, Lifeng Wang, Jianyun Li, and Zhaowei Sun. 2009. "Application of thermal resistance network model in optimization design of micro-channel cooling heat sink."

International Journal of Numerical Methods for Heat and Fluid Flow 19 (3-4): 535–545. ISSN: 09615539. https://doi.org/10.1108/09615530910938425.

Sofloo, Hadi Ghezel, Alireza Shams, and Reza Ebrahimi. 2009. "Study of temperature and velocity distribution of rarefied gas flow in micro-nano channels." *Proceedings of the ASME Fluids Engineering Division Summer Conference 2009, FEDSM2009* 1 (PART B): 1045–1050. https://doi.org/10.1115/FEDSM2009-78017.

"SPARC Industries SARL." n.d. Accessed November 10, 2020. https://www.sparc-industries.com/.

Steinhart, John S., and Stanley R. Hart. 1968. "Calibration curves for thermistors." *Deep-Sea Research and Oceanographic Abstracts* 15 (4): 497–503. ISSN: 00117471. https://doi.org/10.1016/0011-7471(68)90057-0.

Straathof, Michiel H. 2012. "Shape Parameterization in Aircraft Design: A Novel Method, Based on B-Splines." PhD diss., Technische Universiteit Delft. http://resolver.tudelft.nl/uuid:b4aee571-0489-42ff-ab55-d74e980f724a.

Tengfei, Liang. 2015. "Numerical Study of Gas-Surface Interaction Models for Micro Scale Rarefied Gas Flow Simulation." PhD diss., The Hong Kong University of Science and Technology in.

TensorFlow. n.d. "Sobol Sequence." Accessed April 28, 2021. https://www.tensorflow.org/api_docs/python/tf/math/sobol_sample.

———. n.d. "TensorFlow python library." Accessed May 7, 2021. https://www.tensorflow.org/api_docs/python/tf/keras.

Tew, Janet, Jennifer Van Den Driessche, Fred M. Lutfy, E. P. Muntz, Joyce Wong, and Andrew D. Ketsdever. 2000. "A thrust stand designed for performance measurements of the free molecule micro-resistojet." In *36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit.* American Institute of Aeronautics / Astronautics Inc. https://doi.org/10.2514/6.2000-3673.

Thulukkanam, Kuppan. 2013. "Heat Exchanger Design Handbook: Second Edition," 22–25. CRC Press. ISBN: 978-1-4398-4213-3.

Trott, Wayne M., Jaime N. Castaeda, John R. Torczynski, Michael A. Gallis, and Daniel J. Rader. 2011. "An experimental assembly for precise measurement of thermal accommodation coefficients." *Review of Scientific Instruments* 82 (3). ISSN: 00346748. https://doi.org/10.1063/1.3571269.

Turansky, Craig, and Brian Argrow. 2015. "Volumetric geometry for DSMC and the Voldipar code." *Computers and Fluids* 121 (October): 114–132. ISSN: 00457930. https://doi.org/10.1016/j.compfluid.2015.08.009.

Twiggs, Bob. 2009. "Making it Small." *2009 Cal Poly Developers' Workshop*, http://mstl.atl.calpoly.edu/~workshop/archive/2009/Spring/Session%207%20-%20CubeSat%20Alternatives/0900-Twiggs-Making%20it%20Smaller.pdf.

Van Zwol, P. J., D. F. Vles, W. P. Voorthuijzen, M. Péter, H. Vermeulen, W. J. Van Der Zande, J. M. Sturm, R. W.E. Van De Kruijs, and F. Bijkerk. 2015. "Emissivity of freestanding membranes with thin metal coatings." *Journal of Applied Physics* 118 (21). ISSN: 10897550. https://doi.org/10.1063/1.4936851. http://dx.doi.org/10.1063/1.4936851.

Vocale, Pamela, Gian Luca Morini, and Marco Spiga. 2014. "Dilute gas flows through elliptic microchannels under H2 boundary conditions." *International Journal of Heat and Mass Transfer* 71 (April): 376–385. ISSN: 00179310. https://doi.org/10.1016/j.ijheatmasstransfer.2013.12.028.

Vytla, V. V., P. G. Huang, and R. C. Penmetsa. 2010. "Multi objective aerodynamic shape optimiaztion of high speed train nose using adaptive surrogate model." *28th AIAA Applied Aerodynamics Conference* 1 (July): 1–12. https://doi.org/10.2514/6.2010-4383.

Wadsworth, D. C., D. B. VanGilder, L. J. Wysong, C. Kaplan, and D. Mott. 2001. "The design of SUPREM-DSMC - A scalable reacting flow direct simulation monte carlo code." *39th Aerospace Sciences Meeting and Exhibit*, no. January, https://doi.org/10.2514/6.2001-591.

Wagner, Wolfgang. 1992. "A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation." *Journal of Statistical Physics* 66, nos. 3-4 (February): 1011–1044. ISSN: 0022-4715. https://doi.org/10.1007/bf01055714. http://link.springer.com/10.1007/BF01055714.

White, C, M K Borg, T J Scanlon, S M Longshaw, B John, D R Emerson, and J M Reese. 2018. "dsmcFoam+: An OpenFOAM based direct simulation Monte Carlo solver." *Computer Physics Communications* 224:22–43. https://doi.org/10.17632/7b4xkpx43b.1. https://github.com/MicroNanoFlows.

Wong, Joyce, Helen Reed, and Andrew Ketsdever. 2001. "Temperature and Power Predictions for Flight-testing the Free Molecule Micro-Resistojet." In *27th International Electric Propulsion Conference*. http://electricrocket.org/IEPC/232_1.pdf.

Xie, Yanlin, Daxiang Deng, Guang Pi, Xiang Huang, and Chenyang Zhao. 2020. "Fabrication of silicon carbide microchannels by thin diamond wheel grinding." *International Journal of Advanced Manufacturing Technology* 111 (1-2): 309–323. ISSN: 14333015. https://doi.org/10.1007/s00170-020-06085-0.

Yang, J., J. J. Ye, J. Y. Zheng, I. Wong, C. K. Lam, P. Xu, R. X. Chen, and Z. H. Zhu. 2010. "Using direct simulation monte carlo with improved boundary conditions for heat and mass transfer in microchannels." *Journal of Heat Transfer* 132, no. 4 (April): 1–9. ISSN: 00221481. https://doi.org/10.1115/1.4000880.

Yousefi-Nasab, Sadegh, Jaber Safdari, Javad Karimi-Sabet, Ali Norouzi, and Elham Amini. 2019. "Determination of momentum accommodation coefficients and velocity distribution function for Noble gas-polymeric surface interactions using molecular dynamics simulation." *Applied Surface Science* 493 (March): 766–778. ISSN: 01694332. https://doi.org/10.1016/j.apsusc.2019.07.033.

Yu, Shiping, and Timothy A Ameel. 2001. "Slip-flow heat transfer in rectangular microchannels." *International Journal of Heat and Mass Transfer* 44, no. 22 (November): 4225–4234. ISSN: 00179310. https://doi.org/10.1016/S0017-9310(01)00075-8. www.elsevier.com/locate/ijhmt%20https://linkinghub.elsevier.com/retrieve/pii/S0017931001000758.

Zhang, Chang, Mathieu Giroux, Thea Abdul Nour, and Raphael St-Gelais. 2020. "Radiative Heat Transfer in Freestanding Silicon Nitride Membranes." *Physical Review Applied* 14 (2): 1. ISSN: 23317019. https://doi.org/10.1103/PhysRevApplied.14.024072. https://doi.org/10.1103/PhysRevApplied.14.024072.

Zhang, Chonglin. 2010. "Adaptive Mesh Refinement and Cut-cell Algorithms for DSMC Simulation of Hypersonic Flows."

Zhang, Yao, Woong Je Sung, and Dimitri Mavris. 2018. "Application of convolutional neural network to predict airfoil lift coefficient." *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, no. 210049, 1–9. https://doi.org/10.2514/6.2018-1903. arXiv: 1712.10082.

A

Fitting equations for performance parameters vs Slot area plots

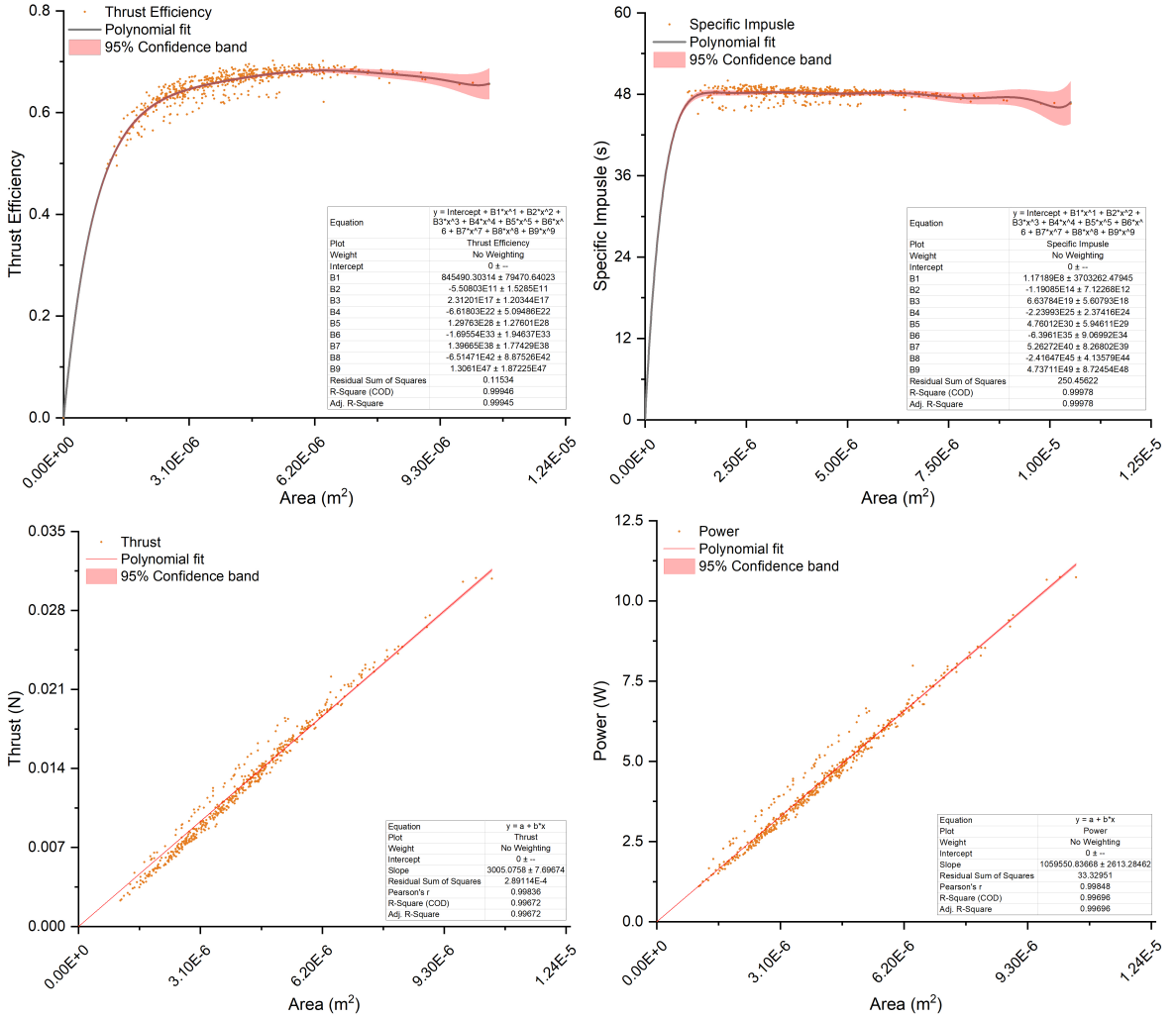


Figure A.1: Fitting equations for the polar dataset.

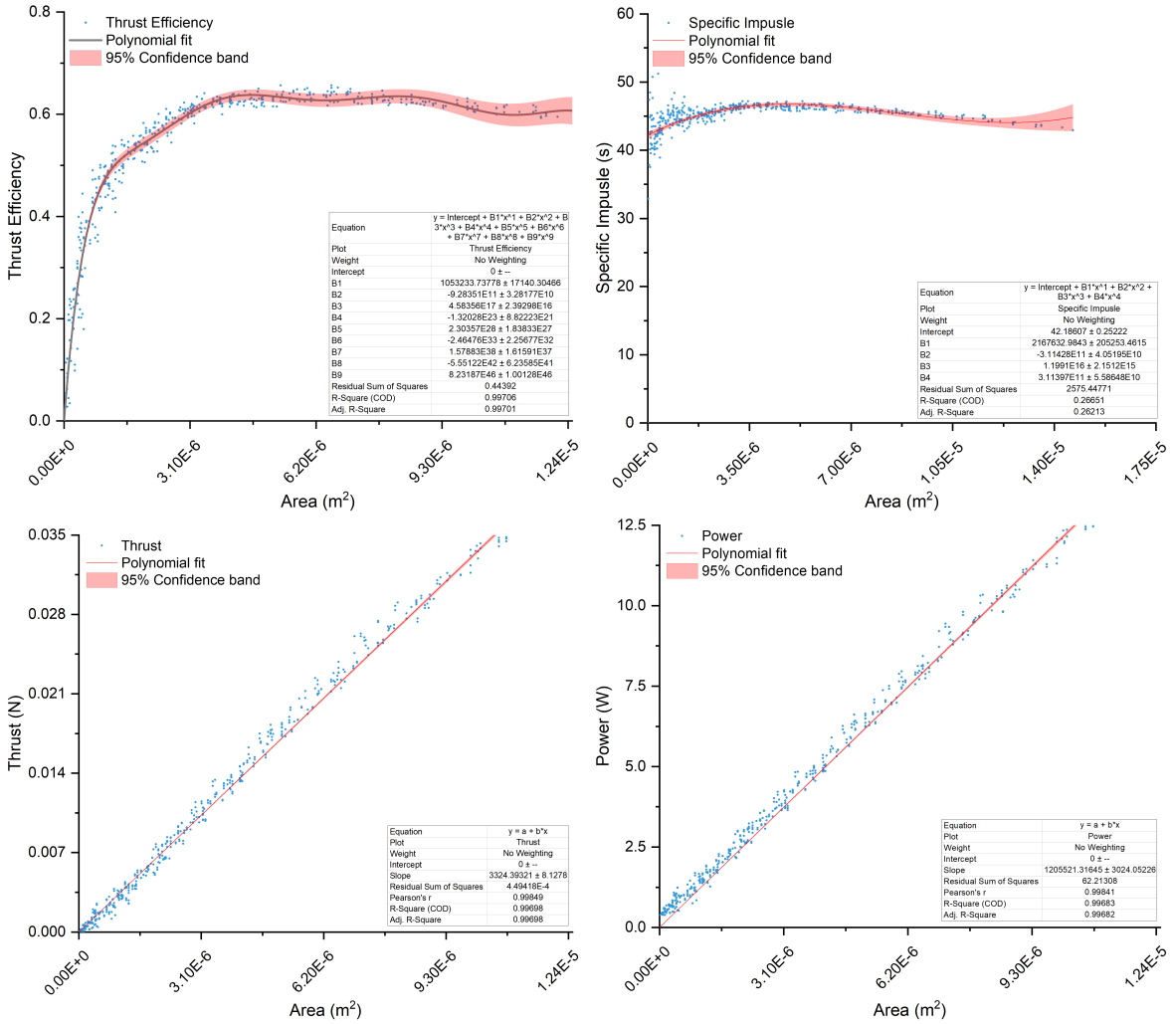


Figure A.2: Fitting equations for the rectangular dataset.