Convergent-Divergent Nozzle Contour Optimization Using Genetic Algorithms

Nathan Rand
Virginia Polytechnic Institute and State University
Advisor: Dr. Christopher Roy

The optimization of converging-diverging (CD) nozzles is critical towards a variety of applications throughout the aviation and space industries — areas closely inline with NASA's mission. This study specifically focuses on the application of machine learning (specifically Genetic Algorithms) and computational fluid dynamics (CFD) software towards the optimization problem surrounding CD nozzle geometry. By manipulating the position of control points connected by cubic splines it is possible to create an open design space and drive the best performing individual CD nozzle towards producing an isentropic flow field as computed through the Euler equations ($\Delta S = 0.0\frac{J}{kgK}$). The optimal case produced from this paper took an initial guess with $\Delta S = 0.935\frac{J}{kgK}$ to a local minimum geometry of $\Delta S = 0.395\frac{J}{kgK}$. The foundation developed in this project opens the door to further work in the application of genetic algorithms towards optimization of CD nozzles and other subsonic/supersonic fluid components.

Nomenclature

α	=	Divergent half angle (deg)
β	=	Tanh dist. parameter (dimensionless)
ΔS	=	Change in Entropy ($J/(kg \cdot K)$)
ϵ	=	Expansion ratio (dimensionless)
ϵ_c	=	Chamber contraction ratio (dimensionless)
γ	=	Specific heat ratio (dimensionless)
κ	=	Geometry Curvature
ϕ	=	Fitness ($J/(kg \cdot K)$)
ξ	=	Non-dimensional coordinate
A_e	=	Nozzle exit area (m^2)
A_i	=	Nozzle exit ring area (m^2)
A_t	=	Throat area (m^2)
b_p	=	Crossover probability (%)
C_p	=	Specific heat at constant P ($J/(kg \cdot K)$)
k	=	Generation number (dimensionless)
k_{max}	=	Number of generations
L_n	=	Nozzle length (m)
m_p	=	Mutation probability (%)
MW	=	Mach Number (dimensionless)
MW	=	Molecular weight (g/mol)
n	=	Number of Individuals
p	=	Number of control points (dimensionless)
P_c	=	Chamber pressure (Pa)
P_e	=	Nozzle exit pressure (Pa)
Pr	=	Prandtl number (dimensionless)
R	=	Universal gas constant ($J/(mol \cdot K)$)
R_e	=	Nozzle exit radius (m)
R_t	=	Throat radius (m)
T_c	=	Chamber temperature (K)

1. Introduction

THIS study aims to investigate the applicability of machine learning techniques coupled with computational fluid dynamics (CFD) software in order to optimize the axisymmetric contour of a convergent-divergent (CD) nozzle. CD nozzles are frequently utilized in aerospace applications ranging from experimental wind tunnels to aircraft and rocket engines. With this wide breadth of applicability the performance of such devices is critical, with even small improvements to their overall efficiency or produced flow quality leading to drastic positive effects on the systems in which they are integrated. Typically when it comes to designing a nozzle three options present themselves to the modern engineer:

1) Utilize basic empirical relations akin to those outlined in undergraduate propulsion courses or as touched on by Huzel and Huang specifically

for application in liquid rocket engines [1].

2) Apply the Method of Characteristics (MOC) in order to produce a shock-free nozzle based on methods outlined by many authors including Anderson [2] and in more modern software related papers such as that by Mishra and Kumar [3]. An example of this technique is displayed below in Figure 1.

3) Employ CFD software in order to produce the most optimal nozzle for a given application while simultaneously validating the design at high fidelity.

The aforementioned options have been ordered by increasing level of fidelity and cost. Typically the MOC is employed with modifications made after the fact through a thorough CFD analysis. Not much work has been done into development of a method that can produce a truly optimal nozzle for a given unique objective function while still keeping cost (in this case, time) low. Therefore, the aim of this study is to investigate the potential for leveraging modern computing techniques in the form of genetic algorithms in order to fill the existing gap in the field of CD nozzle design.

An optimal CD nozzle can mean a variety of things for a variety of different applications. The goal overall is to produce an axisymmetric geometry that can best obtain some objective function for a supersonic exit flow. Typically this involves driving the process towards an isentropic one by minimizing losses; however, it can also include other factors such as minimizing nozzle weight, maximizing thrust, or improving flow uniformity/quality. This wide range of definitions for what an "optimal nozzle" is further dictates the development of a more flexible and effective approach to designing CD nozzles.

Research has been conducted on the optimization of CD nozzles in the past including by the Air Force [4], Quintao [5], and Bahamon and Martinez [6]. The variance between this study and those previously performed is the more open design-space application combined with the inclusion of modern machine-learning algorithms. Most previous research has investigated if the existing methods such as those derived from empirical relations or MOC can be optimized. In contrast to that standard approach, this paper aims to see if an optimal solution can be reached by the model organically given some design space surrounding an initial geometry.

This study began by outlining an initial software framework that allows for a set of input geometric variables to be evaluated as a nozzle geometry for some defined objective function (in this case entropy change ΔS). Next, a variety of machine-learning and numerical methods were applied in order to determine which one would be best suited for producing an optimal nozzle geometry. Specifically, Gaussian-Process models and Reinforcement Learning Neural Networks were implemented first with some, but not great, success. However, based off the success of Genetic Algorithms in other papers (such as Bahamon and Martinez [6]) they were utilized here and found to be suitable for the given problem. Finally, a wide sweep of cases and hyperparameter configurations were utilized to find a best case given the time and resources available. The methods and collected results and conclusions will be discussed throughout the remainder of this paper.

A. Relevance to NASA Mission Directorates

The research conducted within this study is applicable to two of the main NASA Mission Directorates: Aeronautics and Space Technology [7]. CD nozzles, as previously mentioned, have a wide range of applications that are directly applicable to NASA's missions. Aeronautics frequently employs CD nozzles for wind-tunnels (both subsonic and supersonic) along with for research surrounding high-speed in-atmosphere flight noise reduction (such as for the X-59 project). The research into CD nozzle optimization is even more critical for the space technology directorate as NASA begins to ramp up the Artemis missions and the push to return humans to the Moon and then on to Mars. CD nozzles are in almost every propulsion system, from launch vehicle engines to attitude control systems such as cold gas thrusters. Improving the performance of propulsion systems through better performing CD nozzles can allow for both an increased payload capacity while simultaneously increasing system reliability and safety margins.

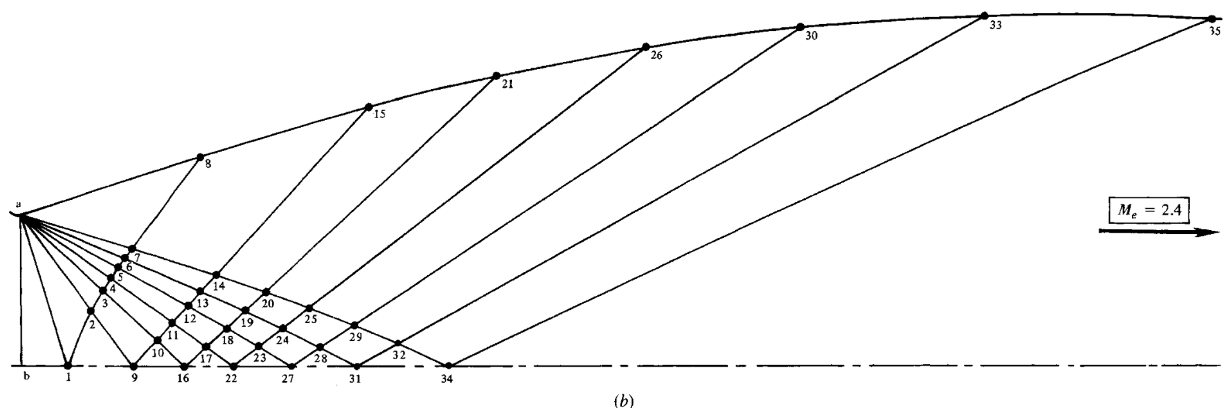

Figure 1. Method of Characteristics Approach to Supersonic Nozzle Design reproduced from Anderson [2]

2. Methodology

The approaches taken throughout this study are critical to discuss as they directly determine the validity of the presented results and drawn conclusions. Therefore, this section of the paper will be focused on analyzing the utilized methods, techniques, and software/algorithms utilized throughout the project.

A. Initial Geometry

The first step for the optimization process is to produce an initial set of geometrical control points based on empirical relationships and basic steady state analysis as outlined by Huzel and Huang for liquid rocket engines [1]. This allows for the determination of the position of critical points for a 80% bell nozzle, such as the nozzle exit position, which is defined based on the equations 2 and 3 below for a given expansion ratio (ϵ). Note that a conical divergent half angle of ($\alpha = 15°$) is used in this study as is recommended by Huzel and Huang [1].

$$\epsilon = \frac{A_e}{A_t} \tag{1}$$

$$L_n = \frac{R_t\left(\sqrt{\varepsilon - 1}\right) + R\left(\sec\alpha - 1\right)}{\tan\alpha} \tag{2}$$

$$R_e = \sqrt{\epsilon R_t} \tag{3}$$

Other points are defined based on other basic geometric relations defined by Huzel and Huang as shown in Figure 2 for reference. Note that a throat position of ($x_t = 0.0,\ y_t = 1.0$) was used in this study for simplicity. Upstream of the throat the positioning of control points was less critical and chosen somewhat arbitrarily with a chamber contraction ratio of ($\epsilon_c = 2.50$) with a suitably long overall chamber section. Effectively no losses were reported in the chamber section throughout the project and, therefore, these assumptions were validated as reasonable for the given initial configuration.

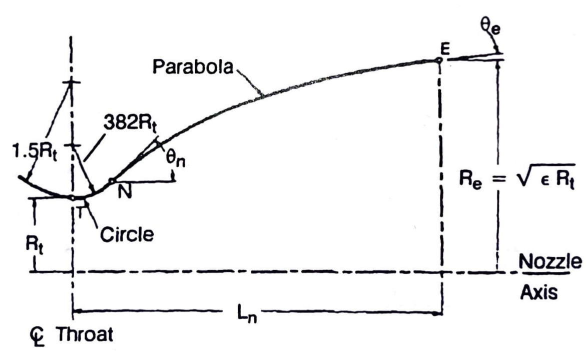

Figure 2. Initial Geometry Empirical Nozzle Design reproduced from Huzel and Huang [1]

These initial empirical points were fit to a cubic spline and then a linearly spaced set of $p - 1$ control points were laid out based on it. An extra control point was added exactly at the throat position leading to the number of total control points to be p.

B. Software Infrastructure (FORGED)

Supporting software was developed in Python in order to support the optimization process surrounding this project. This software is the First Order Rocket Geometry Enhancement and Design tool or (FORGED). The FORGED algorithmic structure and code flow is displayed below in Figure 3.

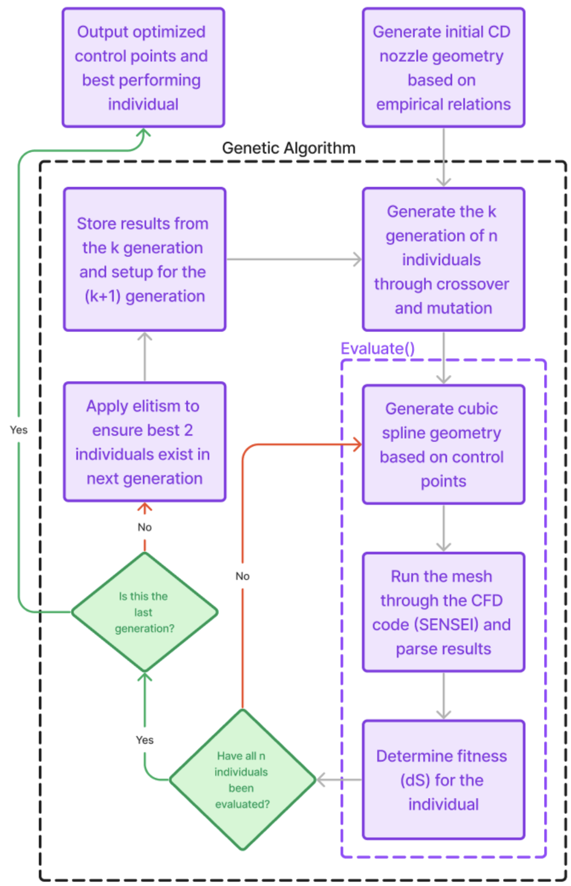

Figure 3. FORGED Algorithm Flow

The main steps that the FORGED software manages is the first 2 parts of the "Evaluate()" routine. First, a cubic spline geometry and corresponding mesh must be generated based on the control points. This is done using Python's 'scipy.interpolate.CubicSpline' library [8] and the mesh generation is done using GMSH's Python wrapper library [9]. The mesh features hyperbolic tangent distributions for its cell spacing with higher density at the throat, inlet, and exit planes. These distributions are based on those outlined by Samuel in his work on the subject [10] and are shown below in equations 4 and 5.

$$x = \frac{L}{2}\left[1 - \frac{\tanh\left[\beta(1-2\xi)\right]}{\tanh\beta}\right] \quad (4)$$

$$x = L\left[1 - \frac{\tanh\left[\beta(1-\xi)\right]}{\tanh\beta}\right] \quad (5)$$

The ΔX values were blended across the connected distributions by matching the derivative values of each distribution with its neighboring ones through numerical convergence of the parameter β.

Second a Python wrapper was developed for Virginia Tech's research CFD Code: SENSEI (finite-volume) that was developed by the research group led by Dr. Christopher Roy. This allows for the SENSEI code to be ran directly from within FORGED on each iteration for an individual's given mesh and geometry in order to determine the flow field and properties relevant to an individual's fitness. SENSEI was run with an inviscid axisymmetric case (to minimize computational time) that features a subsonic inflow pressure ($P_c = 5.0MPa, T_c = 3000K$) and temperature boundary condition at the inlet and a supersonic outlet condition ($P_e = 50kPa$) at the nozzle exit. Slip wall and axisymmetric boundary conditions were also enforced in order to complete the problem setup. The working fluid was defined to have properties ($\gamma = 1.40$, $Pr = 0.72$, $MW = 28.97\frac{g}{mol}$). The flow is calorically perfect and therefore has constant specific heat with constant pressure ($C_p = 1004.45\frac{J}{kgK}$).

C. Genetic Algorithm

The genetic algorithm for this study was implemented using the popular DEAP library in Python [11]. The optimization problem was configured as an effort to minimize the objective function, thus maximizing performance, in each generation with the general form shown in equation 6.

$$\min_{(x,y)\in\mathbb{R}^P} \phi(\vec{x},\vec{y}) \quad (6)$$

The objective function can then be numerically determined based on equations 7 - 9.

$$\phi(\vec{x},\vec{y}) = \Delta S = \text{Fitness} \quad (7)$$

$$\Delta S = A\left(C_p \log\frac{T_e}{T_c} - R\log\frac{P_e}{P_c}\right) \quad (8)$$

$$A = \pi R_t^2\left(y_{i+1} - y_i\right) \quad (9)$$

The goal is to minimize the entropy change (ΔS) from the chamber to the nozzle exit weighted based on the area for each ring (i to $i+1$). This involves taking the area weighted value of quantities such as entropy in ring-shaped regions defining the exit plane of the CD nozzle based on the axisymmetric geometry. The fitness for each individual $\phi(\vec{x},\vec{y})$ is computed based on these outlined equations based on the SENSEI CFD outputs. Each generation in the genetic algorithm has n individuals and runs for k generations. Elitism, which is carrying over the best performers from the previous generation to the next one, is implemented in order to ensure that the seeded initial guess and strong performers do not get diluted out of the population. It is known that the optimal solution is similar to the initial guess case and it is therefore critical to ensure that some individuals similar to this guess persist in the population pool until a more optimal geometry is discovered.

Mutation and crossover is performed when creating the $k+1$ generation and has defined mutation probability (m_p) and crossover probability (b_p). The action space on which mutations can occur is defined based on perturbation ranges surrounding the initial geometry as displayed in Figure 4.

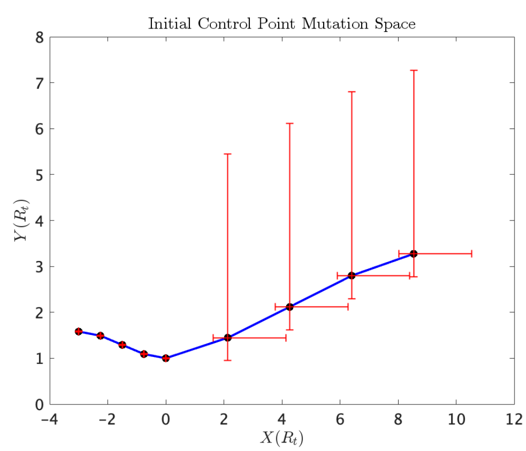

Figure 4. Mutation Action Space for Control Points in the Genetic Algorithm Process

This process persists until the defined final generation (k_{max}) is reached. At this point the best performing individual (with the minimum amount of entropy change) is selected as the resulting local minimum of the optimization process.

A variety of penalty method constraints are implemented that aid in allowing the genetic algorithm to converge including:

1) Equality constraint such that all points are increasing in x and y (ensuring all control points p are sequential in the design space). The applied penalty is based upon a scaled summation of a fixed factor and the amount of non-sequentiality summed across all points as shown below in Equation 10 as the non-sequential penalty (p_{ns}). The difference is set to zero if the difference returned by $diff()$ is positive (or sequential such that position $i+1$ > position i) for each point set. This penalty is only applied if any set of points is non-sequential.

$$p_{ns} = 1000 * (25 + diff(\vec{x}_{i+1} - \vec{x}_i)) \tag{10}$$

2) Penalty factors applied for excessively non-smooth geometry as determined by a second derivative of the cubic spline and a basic curvature relation shown below in equation 11. This is to try and prematurely avoid failed/non-converging CFD runs.

$$\kappa = \frac{|f''(\vec{x})|}{(1+(f'(\vec{x}))^2)^{\frac{3}{2}}} \tag{11}$$

3) Penalty factors for geometric cases that are sequential and smooth but still lead to a failure in either GMSH or SENSEI.

These penalty factors and try-catch cases ensure that the code can effectively converge even given a very wide range of potential CD nozzle geometries.

3. Results & Discussion

A wide range of optimization cases were ran with varying mutation spaces as well as differing hyper-parameters (primarily p, k_{max}, n, m_p, and b_p). The optimal case was run with the following parameters: $p = 9$, $k_{max} = 50$, $n = 100$, $m_p = 0.25$, and $b_p = 0.5$. The final optimized control point geometry (along with its produced cubic splines) can be seen in Figure 6 along with the original initial guess control point geometry in Figure 5.

The optimization process progressed through all 50 generations, consistently approaching a local minimum of ($\phi(\vec{x},\vec{y}) = \Delta S = 0.395 \frac{J}{kgK}$). Rapid convergence appears to occur over the first 5 iterations objectives of the work presented here.

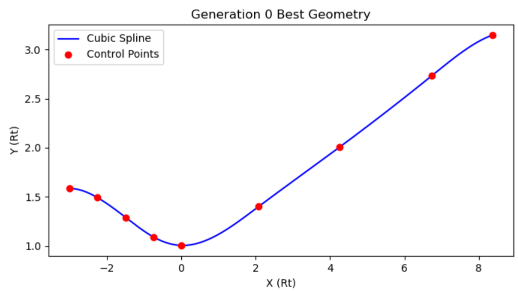

Figure 5. Initial Guess CD Nozzle Control Points and Corresponding Cubic Splines

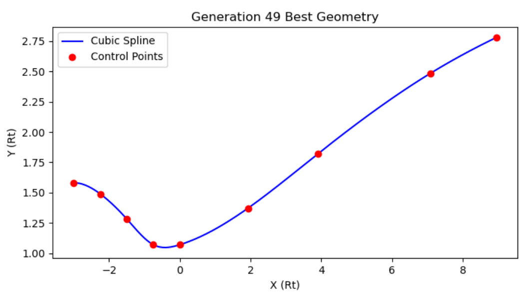

Figure 6. Optimized CD Nozzle Control Points and Corresponding Cubic Splines

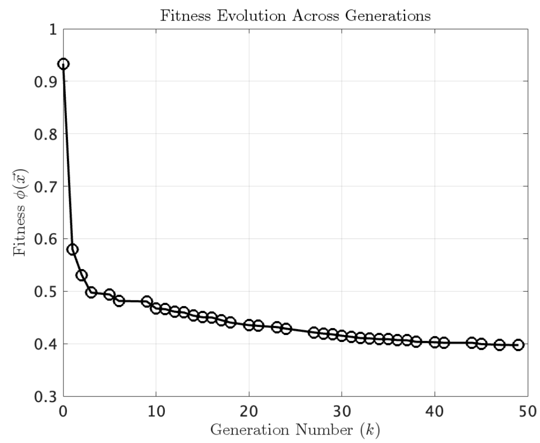

Figure 7. Best Performing Individual's Fitness in Each Generation

and levels off with smaller relative fine adjustments for the remaining 45 iterations. By opening up the design space with more individuals and a higher mutation probability it is likely possible to more effectively explore the design space and determine a geometry that produces true isentropic flow such that ($\Delta S_{ideal} = 0.00 \frac{J}{kgK}$). The optimization process fitness for the best performer in each generation is displayed below for reference in Figure 7.

Flow fields for both entropy change (ΔS) and Mach number (M) were also determined for three cases: the initial guess ($k = 0$), the best performer in one of the middle generations ($k = 27$), and the optimal case ($k = 49$). Both of these contour plot progressions can be seen in Figures 8 and 9. It can be seen clearly that the optimization process creates a flow field that has less losses while maintaining uniformity. The original large entropy spike near the nozzle wall at the exit plane has been entirely removed by the optimization process. Overall, as the optimization process progresses the nozzle flow field becomes more isentropic while maintaining an acceptable and even more uniform Mach field. This was the original goal of the study and therefore validates the primary

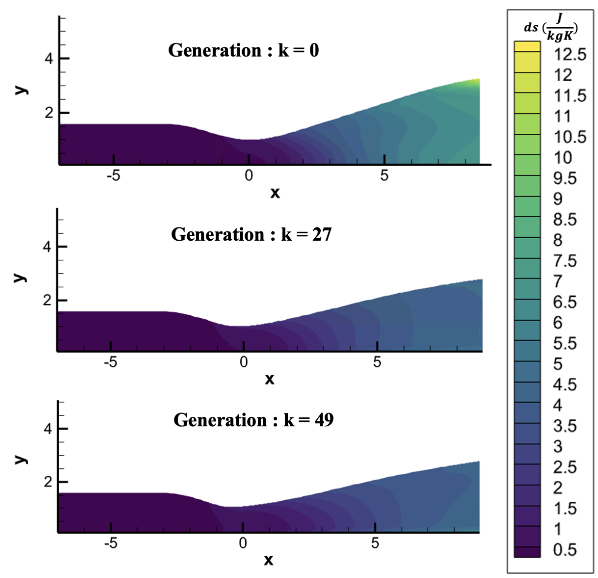

Figure 8. Entropy Change (ΔS Contour Plots Throughout Optimization Process

4. Conclusion

Throughout the course of this study an optimization technique and the corresponding optimal results for axisymmetric CD nozzle geometry presented in order to minimize entropy gain and losses throughout the nozzle flow field. Through a thorough investigation genetic algorithms were determined to be the best option for the presented problem. Developing the FORGED software in Python allowed for the overall

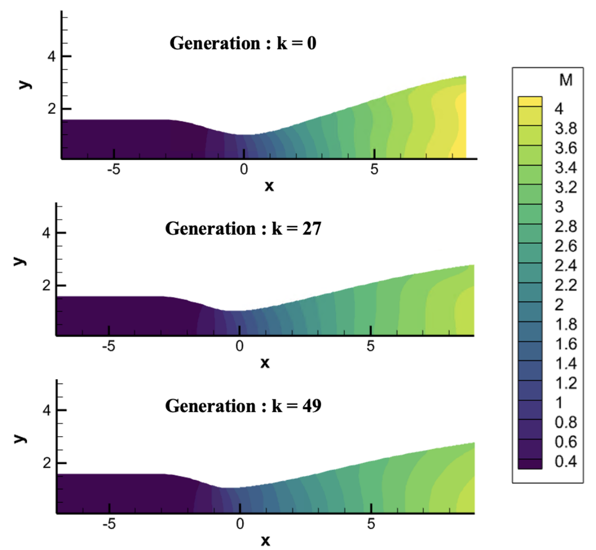

Figure 9. Mach Number (M Contour Plots Throughout Optimization Process

meshing and CFD process to be seamlessly integrated into DEAP's genetic algorithm structure. An optimal result was produced that reduced the entropy gain from $\Delta S = 0.934\frac{J}{kgK}$ in the initial guess to $\Delta S = 0.395\frac{J}{kgK}$, a significant change of 57.71% as the optimization problem drives the flow towards an isentropic state. This study has also set up a strong foundation for future work in the field of organic and open-design space CD nozzle optimization using machine learning techniques. By leveraging more computational resources and parallel computing it could be possible to drive the nozzle towards a global minimum where ($\Delta S_{ideal} \rightarrow 0.0\frac{J}{kgK}$). The work can then be even further expanded by implementing more control points for fine control of the geometry, non-inviscid CFD conditions to incorporate more realistic areas of loss, and other more accurate (but expensive) cases. The organic nature of the FORGED + Genetic Algorithm approach taken in this paper leaves the possibility for also integrating additional terms into the objective-function to optimize for a variety of applications such as weight-reduction, flow-uniformity, or thermal loading.

Acknowledgments

The author would like to thank Dr. Roy for his support, insight, and the opportunity to conduct this research throughout the course of my senior year. I also would like to thank William Jordan and all the other members of the research group. Finally, I extend my gratitude to VSGC for the opportunity to present my research and the funding they provided to conduct it.

References

[1] Huzel, D. K., and Huang, D. H., *Modern Engineering for Design of Liquid-Propellant Rocket Engines, Volume 147*, American Institute of Aeronautics and Astronautics, Washington, D.C., 1992.

[2] Anderson, J. D., *Modern Compressible Flow: With Historical Perspective*, 3rd ed., McGraw-Hill Education, 2003, Chap. 9.

[3] Mishra, N. K., Kumar, A. S., Ganesh, M., and Alam, M. A. R., "Design of Minimum Length Supersonic Nozzle using the Method of Characteristics," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Vol. 9, No. 2, 2019, pp. 1370–1374. doi: 10.35940/ijitee.B6183.129219.

[4] Shope, F. L., "Design Optimization of Hypersonic Test Facility Nozzle Contours Using Splined Corrections," , March 2005. Final Report for Period October 1, 2001 - September 30, 2004.

[5] Quintao, K. K., "Design Optimization of Nozzle Shapes for Maximum Uniformity of Exit Flow," Master's Thesis, Florida International University, November 2012. URL `https://digitalcommons.fiu.edu/etd/779`, available at FIU Digital Commons.

[6] Bahamon, J., and Martinez, M., "Study of Fluid-Dynamic Behavior in a Convergent-Divergent Nozzle by Shape Optimization Using Evolutionary Strategies Algorithms," *Proceedings of the IMechE Part G: Journal of Aerospace Engineering*, Vol. 237, No. 12, 2023, pp. 2844–2862. doi: 10.1177/09544100231163372.

[7] National Aeronautics and Space Administration, "NASA Directorates," `https://www.nasa.gov/directorates`, 2024. Accessed: 2024-03-27.

[8] SciPy developers, "SciPy Interpolate CubicSpline documentation," `https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html`, 2023.

[9] Gmsh developers, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," `https://gmsh.info/`, 2023.

[10] Indian Institute of Technology Kanpur, "Grid Generation Techniques," `https://home.iitk.ac.in/~roshanj/grid.html`, 2023.

[11] DEAP developers, "DEAP: Distributed Evolutionary Algorithms in Python Documentation," , 2023.