

Wright State University

CORE Scholar

---

[Browse all Theses and Dissertations](#)

[Theses and Dissertations](#)

---

2018

## Coupling Computational Fluid Dynamics Analysis and Optimization Techniques for Scramjet Engine Design

Nathan T. McGillivray  
*Wright State University*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/etd\\_all](https://corescholar.libraries.wright.edu/etd_all)



Part of the [Mechanical Engineering Commons](#)

---

### Repository Citation

McGillivray, Nathan T., "Coupling Computational Fluid Dynamics Analysis and Optimization Techniques for Scramjet Engine Design" (2018). *Browse all Theses and Dissertations*. 2027.  
[https://corescholar.libraries.wright.edu/etd\\_all/2027](https://corescholar.libraries.wright.edu/etd_all/2027)

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact [library-corescholar@wright.edu](mailto:library-corescholar@wright.edu).

# **Coupling Computational Fluid Dynamics Analysis and Optimization Techniques for Scramjet Engine Design**

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

By:

Nathan T. McGillivray  
B.S.M.E., Wright State University, 2016

2018  
Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

July 25, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Nathan T. McGillivray ENTITLED Coupling Computational Fluid Dynamics Analysis and Optimization Techniques for Scramjet Engine Design BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Mechanical Engineering.

---

Mitch Wolff, Ph.D.  
Thesis Director

---

Joseph C. Slater, Ph.D., P.E.  
Chair, Department of Mechanical  
and Materials Engineering

Committee of Final Examination:

---

Dean Eklund, Ph.D.

---

Scott Thomas, Ph.D.

---

Mitch Wolff, Ph.D.

---

Barry Milligan, Ph.D.  
Interim Dean of the Graduate School

## **ABSTRACT**

McGillivray, Nathan T., M.S.M.E. Department of Mechanical and Materials Engineering, Wright State University, 2018. Coupling Computational Fluid Dynamics Analysis and Optimization Techniques for Scramjet Engine Design.

Various aspects of hypersonic vehicles are being rapidly explored for improved functionality.

One of the main areas of consideration is the fueling of a Supersonic Combusting Ramjet (scramjet) engine. Using Computational Fluid Dynamics (CFD), computer simulations can be performed to analyze the flow physics of a scramjet. In this research, an optimization code, Dakota, is integrated with the CFD to optimize a set of parameters to maximum thrust. In this study, the fuel injection and combustion is replaced with heat sources. This simplification greatly reduces the computational requirements. Additionally, the 3D geometry is reduced to an axisymmetric 2D geometry because three dimension effects like mixing and combustion are not being modeled. With this simplified model, the optimization and CFD algorithm is executed to find the heat addition for maximum thrust. Different optimization methods have been explored to reduce computational times. A genetic algorithm was selected because of its robust abilities. Additionally, a sampling algorithm was selected because of its abilities to explore the whole design space. Furthermore, the sampling method enables additional studies, such as sensitivity studies, to be completed. In addition to optimization studies, calibration studies are performed to obtain the heat source values that correspond to a given experimental wall pressure distribution. Knowledge of the optimized heat distribution will assist in the optimization of fueling splits and injector locations for a more detailed combustion investigation in which similar optimization techniques can be applied.

# TABLE OF CONTENTS

1. INTRODUCTION .....	1
1.1. Problem Overview.....	1
1.2. Approach Taken .....	2
1.3. Thesis Organization.....	3
2. BACKGROUND .....	3
2.1. Ramjet .....	3
2.2. Scramjet.....	5
2.2.1. Research Cell 22 (RC22).....	6
2.3. Computational Analysis .....	8
2.3.1. Computational Fluid Dynamics (CFD) .....	8
2.3.2. Metacomp's CFD++ .....	9
2.3.3. Dakota Toolkit.....	14
3. METHODOLOGY .....	20
3.1. Geometry.....	21
3.2. Grid Sensitivity .....	22
4. RESULTS .....	27
4.1. Initial Heat Source Study .....	27
4.2. Fixed Heat Load Optimization-Genetic Algorithm .....	32
4.3. Preliminary Heating Optimization Study-Sampling Method.....	35
4.4. Wall Pressure Calibration Study .....	40
5. CONCLUSION.....	45
6. FUTURE WORK.....	46
7. REFERENCES .....	48
APPENDIX A: Script/ Input File Examples.....	50
APPENDIX B: Additional Plots.....	66

## TABLE OF FIGURES

Figure 1: Quickstat Concept Vehicle [3] .....	1
Figure 2: Ramjet Schematic [4] .....	4
Figure 3: Open Loop Brayton Cycle [5] .....	4
Figure 4: Scramjet Schematic [4] .....	5
Figure 5: Engine Performance Comparison [5] .....	6
Figure 6: RC22 Testbed Schematic [6] .....	7
Figure 7: Scramjet Design Configurations [6] .....	7
Figure 8: Complete Algorithm Flow Chart .....	15
Figure 9: Location for Mach Number Extraction & Location of Heat Sources .....	21
Figure 10: 2D Axisymmetric to Rotated 3D Geometry .....	22
Figure 11: Locations of Increased Y+ Values .....	24
Figure 12: Baseline Grid (Top) and 2x Globally Refined Grid (Bottom) .....	24
Figure 13: Grid Sensitivity Heating Configuration .....	25
Figure 14: Grid Sensitivity Contour Comparison .....	26
Figure 15: Comparison of Heating Configurations .....	28
Figure 16: Wall Pressure Comparison for Single Heat Configuration .....	29
Figure 17: Thrust Production Comparison for Single Heat Configuration .....	30
Figure 18: Wall Pressure Comparison for Distributed Heat Configuration .....	31
Figure 19: Thrust Production Comparison for Distributed Heat Configuration .....	32
Figure 20: SOGA Two Variable Heat Sources .....	33
Figure 21: Evolution of SOGA Two Heat Population .....	33
Figure 22: Isolator Mach Number Variation .....	34
Figure 23: MOAT Three Heat Sources .....	36
Figure 24: PSUADE MOAT Results .....	37
Figure 25: Reduction of Heat Source Design Space .....	38
Figure 26: Reduced Design Space MOAT Results .....	39
Figure 27: Wall Pressure Design Space Reduction .....	42
Figure 28: Initial Optimal Wall Pressure Calibration .....	42
Figure 29: Revised Optimal Wall Pressure Calibration .....	43
Figure 30: Revised Optimal Wall Pressure Error .....	44
Figure 31: Thrust Curve for Reduced Design Space MOAT Optimization Study .....	66
Figure 32: Additional Convergence Check .....	67

## TABLE OF TABLES

Table 1: Grid Sensitivity Study.....	26
Table 2: Distributed Configuration Heating Summary.....	28
Table 3: Summary of Optimal Solutions in Original Design Space.....	37
Table 4: Summary of Optimal Solutions in Reduced Design Space .....	40

## NOMENCLATURE (TEXT)

AFRL	=	Air Force Research Laboratories
CFD	=	Computational Fluid Dynamics
COLIN	=	Common Optimization Library Interface
CPU	=	Central Processing Unit
DACE	=	Design and Analysis of Computer Experiment
DoD	=	Department of Defense
DoE	=	Design of Experiment
DSRC	=	DoD Supercomputing Resource Center
EA	=	Evolutionary Algorithm
MOAT	=	Morris One-at-a-Time
MOGA	=	Multi-Objective Genetic Algorithm
PBS	=	Portable Batch System
PSUADE	=	Problem Solving Environment for Uncertainty Analysis and Design Exploration
RANS	=	Reynolds-Averaged Navier Stokes
RC22	=	Research Cell 22
RMS	=	Root-Mean-Square
RQH	=	High Speed Systems Division – Aerospace Systems Directorate
SOGA	=	Single Objective Genetic Algorithm
WSU	=	Wright State University

## NOMENCLATURE (EQUATIONS)

$e$	=	Total Specific Energy
$\vec{F}$	=	Inviscid Flux Vector
$f_\mu$	=	Low-Reynolds Number Function
$\vec{G}$	=	Viscous Flux Vector
$I_{sp}$	=	Specific Impulse
$k$	=	Turbulent Kinetic Energy
$K$	=	Thermal Conductivity
$n$	=	Number of Design Parameters
$p$	=	Static Pressure
$P$	=	Number of Partitions
$P_k$	=	Production Rate of Turbulent Energy
$\vec{Q}$	=	Vector of Dependent Conservation Variables
$R$	=	Gas Constant
$R_t$	=	Turbulent Reynolds Number
$s$	=	Specific Entropy
$S$	=	Dimensional Fluid Strain Magnitude
$\dot{S}$	=	Source Terms
$S_{ij}$	=	Mean Strain Tensor
$T$	=	Static Temperature
$T_t$	=	Realizable Turbulence Timescale



$u$	=	Velocity in X-Direction
$u_*$	=	Friction Velocity
$v$	=	Velocity in Y-Direction
$w$	=	Velocity in Z-Direction
$y^+$	=	Non-Dimensional Wall Distance
$\gamma$	=	Ratio of Specific Heats
$\delta_{ij}$	=	Kronecker Delta Function
$\varepsilon$	=	Rate of Energy Dissipation
$\mu$	=	Dynamic Viscosity
$\mu_t$	=	Eddy Viscosity
$\nu$	=	Kinematic Viscosity
$\rho$	=	Density
$\tau$	=	Viscous Stress

## **ACKNOWLEDGEMENTS**

I am thankful for the guidance and direction of Mitch Wolff, my thesis director and Mark Hagenmaier, my AFRL sponsor. I appreciate the time they have dedicated to assist my research performed. Without either of them, I would not have been able to complete this work. Not only did they provide their time, but their knowledge and resources which was critical in the completion of this work.

I would also like to thank my parents and brothers for the support they have provided during my academic career. Their moral support provided the motivation to further my academic career which has already begun paying off.

Lastly, I would like to thank Dayton Area Graduate Studies Institute (DAGSI) and Wright State University for the opportunities they have provided. This work would not be possible without the collaboration and financial support of both institutions. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

# 1. INTRODUCTION

## 1.1. Problem Overview

There has been a change in how outer space is being viewed over the past couple of decades. As the number of space technologies increase, so does the interest in inexpensive, responsive space access. The current systems available for access to space are very costly and are unable to provide quick response. Space access is of great importance to the United States military.

Although space may never be a true battleground in the same way as land, sea, and air; it is not a far stretch to say that space will play a key role to the warfighter. The infrastructure in space plays a very important role in surveillance and reconnaissance missions and therefore it is easy to see the importance of protecting these assets [1]. In addition, weapons themselves will also use space in their trajectories. Responsive access to space is critical to the protection of United States assets in space and is necessary because of the extent to which the United States depends on these assets [2]. The Commission to Assess United States National Security Space Management and Organization, commonly referred to as the “Space Commission”, has issued some findings as to the importance of space to national security.

A new space vehicle is needed to reduce operating expense and increase the number of sorties for rapid response. Reusable launch vehicles, such as Quickstat, are being considered. Quickstat is a Two Stage to Orbit reusable launch vehicle, shown in Figure 1. This research analyzes a scramjet combustor geometry being studied by

AFRL/RQH for vehicles, such as the



*Figure 1: Quickstat Concept Vehicle [3]*

Quicksat, to enable the improved performance of the vehicle [3]. Optimization of the heat release distribution of the fuel was performed with the CFD code CFD++ from Metacomp Technologies, Inc. Axisymmetric flow paths (i.e. circular) in a scramjet combustor provide benefits not found in a typical rectangular flow path. The benefits of circular compared with rectangular combustors of the same cross-sectional area include elimination of corner-flow effects, increased structural efficiency, and reduced weight; however, circular combustors also present challenges related to effective fuel injection and flame propagation, especially at large scales. The use of CFD++ software will aid in the optimization of a new fuel delivery system, which will significantly improve the combustion efficiency. This work targeted the coupling of the popular CFD code, CFD++, and a powerful optimization toolkit, Dakota. This provided the foundation to a dynamic computational analysis tool to enhance both the design time and design performance.

## **1.2. Approach Taken**

The initial phase of the research involved performing baseline CFD simulations of the AFRL/RQH experimental geometry in a 2-D analysis without combustion using CFD++. This used the current scramjet geometry without fuel injection, combustion, or heat addition. The simulation provided a general understanding of the computational tools and the flow field. Next, volumetric sources were added to allow for heat addition to the flow. These volumetric heat sources provided a representation of the combustion that takes place in the flow. Although fuel injection and combustion was neglected, the heat sources provide an appropriate approximation to perform optimization studies. However, prior to an optimization studies, a sensitivity study of the computational grid was performed to identify the resolution needed to obtain accurate solutions while remaining computationally efficient. Next, the CFD software was utilized to perform an optimization of the heating distribution, for a fixed total heating, to give the highest combustor performance while ensuring the shock system would remain in the isolator. Once the

optimization studies were completed, a calibration study was completed to determine how to distribute the heat to match the simulation wall pressure to experimental data. Additionally, different optimization methods and techniques were considered throughout the heating studies to develop an efficient optimization routine specifically for 2-D scramjet combustion cases.

### **1.3. Thesis Organization**

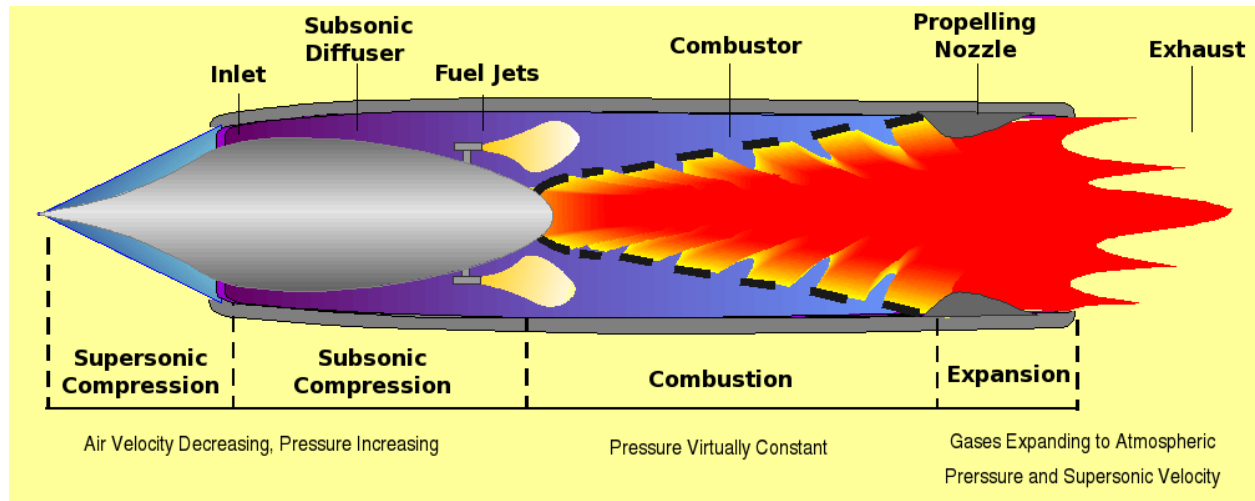
This thesis is organized as follows. Section 2 covers important background information necessary for the work completed in this thesis. This includes discussion of supersonic engines, CFD and equations used, optimization tools, and the coupling of CFD analysis with optimization tools. Following is Section 3, discussing the methods used in preparing the CFD model for optimization. Next, Section 4 covers the results of various studies performed for the work in this thesis. Specifically, this section discusses the progression of the algorithm to successfully couple a simplified CFD model of a scramjet and optimization tools. Section 5 covers the finding of this work and the application of the algorithm. Lastly, Section 6 concludes this thesis and discusses how this work can be extended in the future.

## **2. BACKGROUND**

### **2.1. Ramjet**

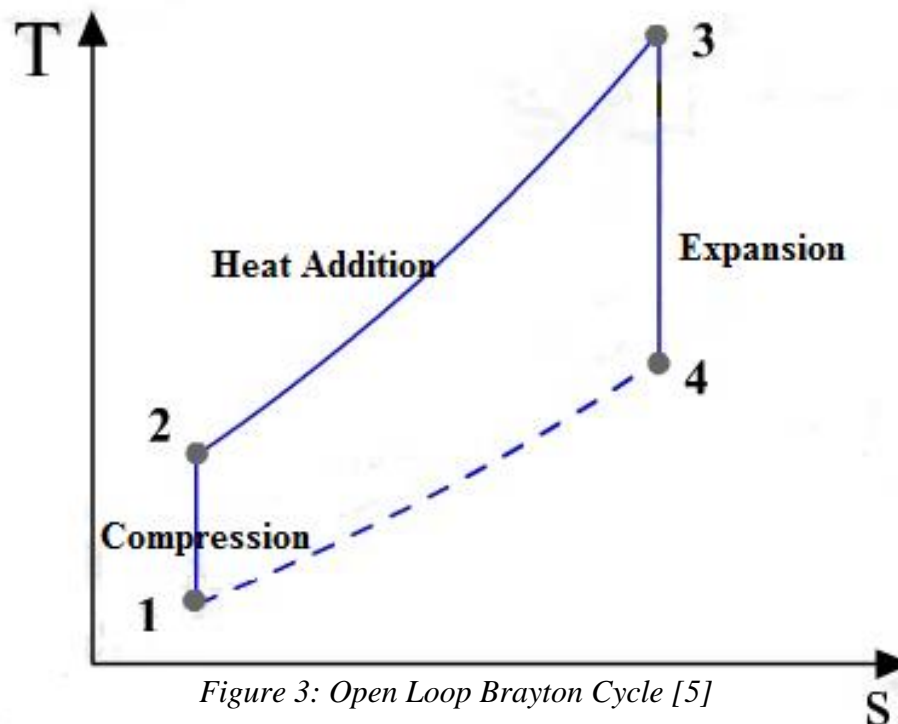
A ramjet is a type of jet engine that utilizes air being forced into a duct to compress the air rather than a mechanical compressor to compress the air. The air being forced into the inlet duct is typically done by the forward motion of the engine. During this process, the flow velocity decreases to subsonic conditions and the pressure increases. The decrease to subsonic conditions restricts the operability to a freestream Mach number between 2 and 5. Following the compression process, fuel, typically a hydrocarbon or hydrogen, is injected to the flow to be combusted. During combustion, the flow in the combustor is controlled so that the pressure is nearly constant. Finally, the flow is expanded to the atmosphere through a nozzle. In the nozzle,

the pressure decreases while the velocity increases to supersonic conditions [4]. As a result of this acceleration, thrust is produced. Figure 2 below shows a schematic of ramjet and its components.



*Figure 2: Ramjet Schematic [4]*

The ramjet illustrates an open loop Brayton cycle. Figure 3 shows the process on a temperature-entropy diagram, where the supersonic and subsonic compression take place from 1 to 2, combustion during 2 to 3, and expansion through the nozzle during 3 to 4 [5].

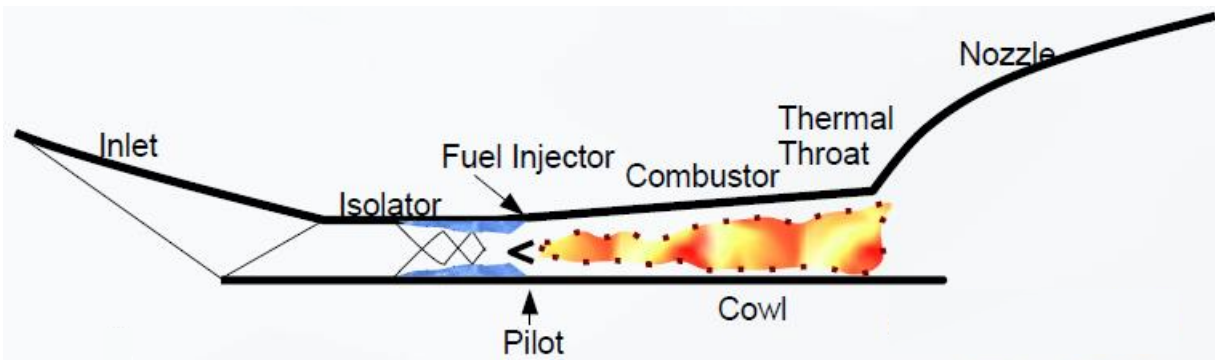


*Figure 3: Open Loop Brayton Cycle [5]*

## 2.2. Scramjet

A scramjet is a supersonic combustion ramjet engine. Similar to a ramjet, supersonic air is forced into the engine inlet which is then compressed through shock waves, prior to combustion.

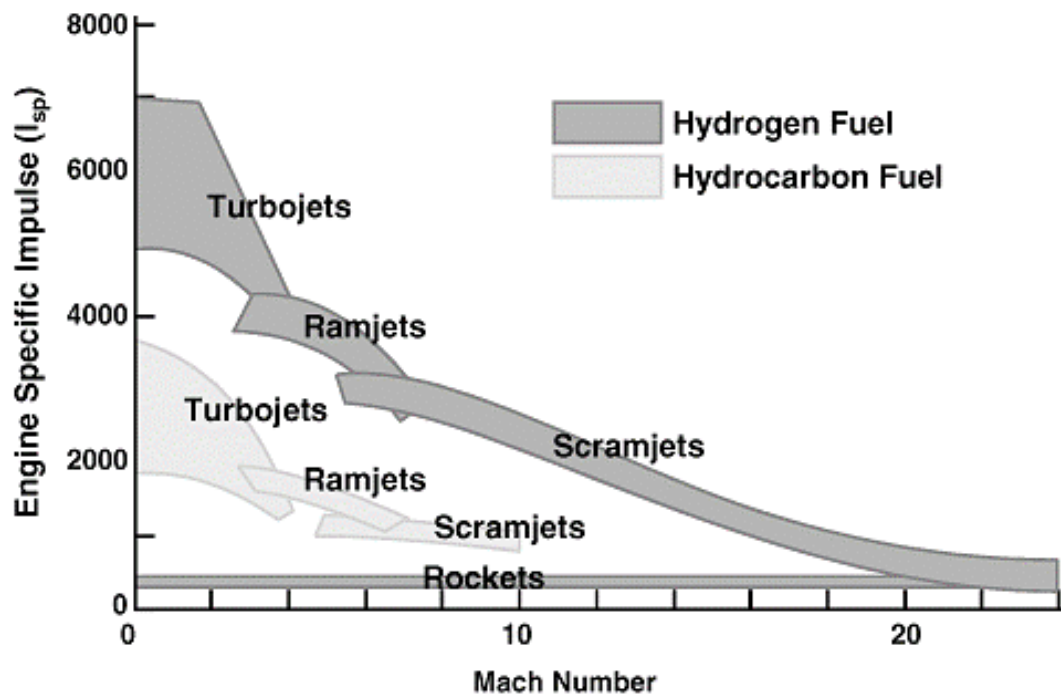
However, in a scramjet, the air being compressed slows down, but remains supersonic rather than slowing to subsonic conditions. As a result, scramjets require higher speed freestream flow for operation. Scramjets typically operate at Mach numbers greater than 7. This also means that the flow throughout the engine is supersonic; therefore, the engine does not require a converging-diverging nozzle at the exit [4]. A nominal scramjet is shown below in Figure 4.



*Figure 4: Scramjet Schematic [4]*

Scramjets present a much more complex shock structure. Initially, an oblique shock is formed from the inlet of the scramjet. This provides an initial compression of the flow. Additionally, a series of oblique shocks are present in the isolator of the scramjet. These shocks are commonly referred to as the shock train and provide additional compression. This is similar to the diffuser seen in a ramjet. The fuel injection and combustion in the supersonic flow causes a sudden increase in the pressure. In turn, this pushes the shock train forward in the scramjet. As a result, the strength and location of the shock train is extremely sensitive to the heat release during the combustion process. The shock train is a crucial component to the scramjet performance. The work of the engine can be illustrated by the area under the T-s curve shown in Figure 3. One way to increase the area, i.e. increase the thrust of the engine, is to increase the heat addition.

However, the increase of heat addition pushes the shock train towards the inlet. If the heat addition is too much, unstart will occur. Unstart is a result of the shock train standing in or near the inlet, resulting in a restriction of flow to the engine. Operation at higher Mach numbers allow for increased heat addition as the shock train is forced away from the inlet as the freestream Mach number is increased. A comparison of the operational ranges and specific impulse of a typical turbojet, ramjet, and scramjet engine is shown below in Figure 5.



Courtesy of the Air Force Propulsion Directorate (circa 1990's)

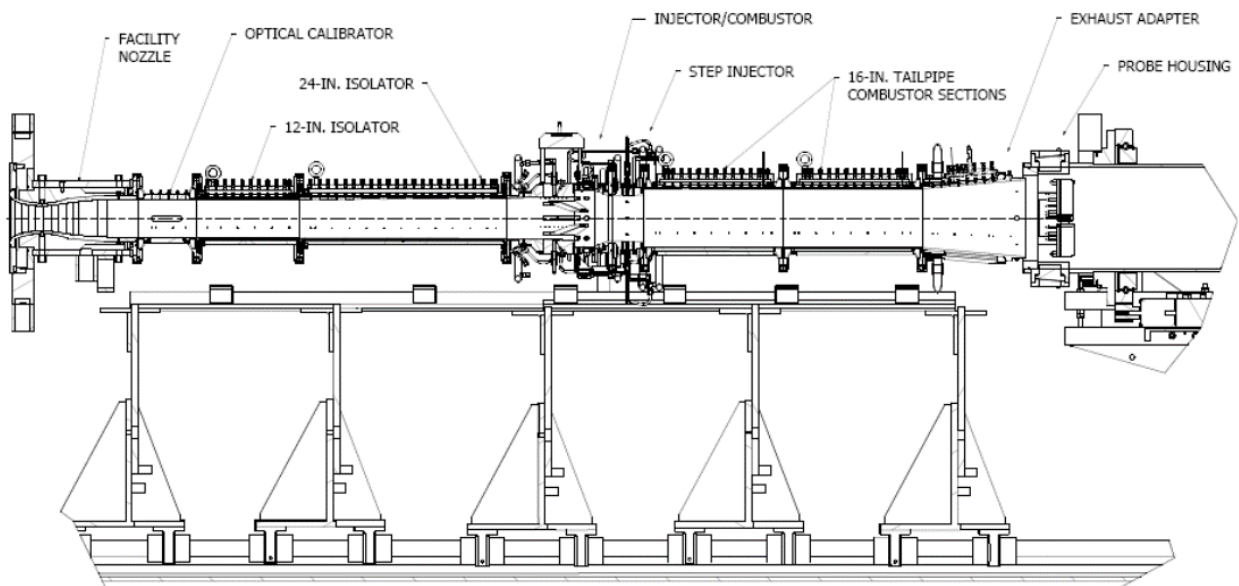
Figure 5: Engine Performance Comparison [5]

### 2.2.1. Research Cell 22 (RC22)

A test facility located at Wright Patterson Air Force Base houses a scramjet testbed used for various aspects of scramjet design and analysis [6]. This research facility provides extensive data on hypersonic propulsion. However, the experimental data collected is limited to the capabilities

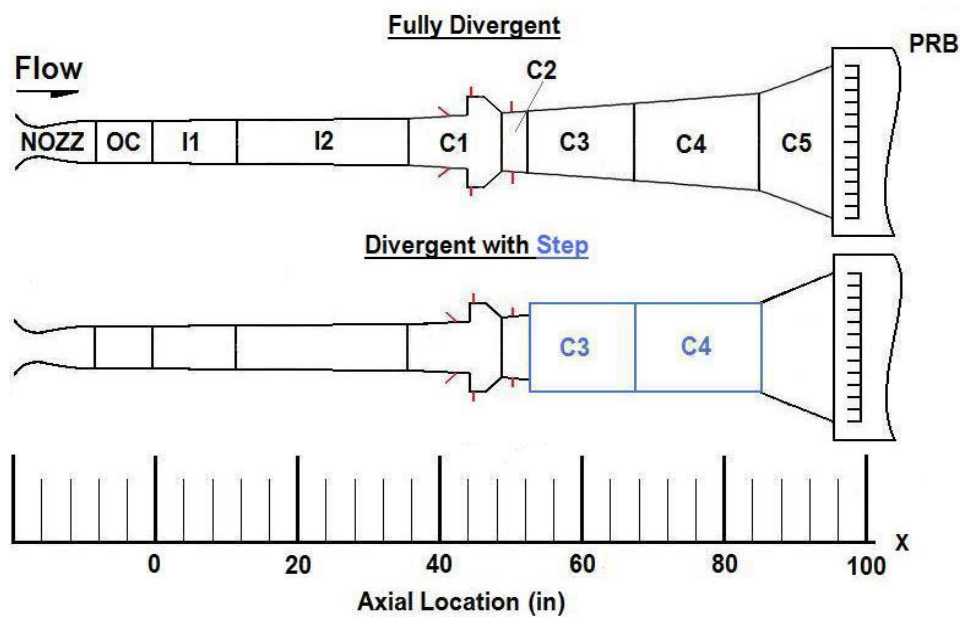


of the test equipment. In addition, these tests require numerous hours of preparation, as well as significant funding. Figure 6 below shows the components of the RC22 testbed.



*Figure 6: RC22 Testbed Schematic [6]*

The research by Milligan [6] reported on two scramjet configurations. These configurations include a divergent with a step configuration and a fully divergent configuration. The two configurations can be seen below in Figure 7.



*Figure 7: Scramjet Design Configurations [6]*

The two configurations are identical except for the combustion sections (C3 & C4) following the combustion cavity (C1). Accompanying the two different configurations, RC22 can change the facility nozzle to provide a wide range of inlet conditions. Between the limited testing equipment and the financial requirements of this test-stand, the Air Force Research Lab (AFRL) is relying more and more on the powerful computational tools available.

## **2.3. Computational Analysis**

### **2.3.1. Computational Fluid Dynamics (CFD)**

CFD has grown to be a powerful tool in the world of engineering, especially over the past 5-6 years as computational power has grown immensely. With current computer hardware and CFD software, high fidelity CFD simulations have become more practical for analysis and design. In the past, computations would take weeks, if not months, to complete leaving CFD analysis an engineering tool that was under-utilized. However, many detailed computations can now be performed in less than one week. On this short timeline, engineers can couple computational analysis into the design process and even couple it with experimental work. This allows for the computational analysis to complement experiments in various aspects, such as cost, time, and equipment restrictions. Typically, the cost for CFD licenses are comparable to the cost of one or two experiments in terms of scramjet engine testing. In terms of time, the DoD Supercomputing Resource Center provides sufficient computational resources to run multiple simulations at one time which would simulate the various configurations and inlet conditions. The time to complete these multiple simulations would represent a fraction of the time required to setup the scramjet test and reconfigure the system for the various inlet conditions. Additionally, this would require a whole team to complete whereas the multiple simulations could be completed by a single engineer. Lastly, CFD simulations provide an abundance of data throughout the whole flow field. Essentially, CFD simulations will provide any property at any location within the flow. On the

other hand, experiments are restricted to specific locations and properties that can be observed. This aspect of experimental restrictions is more prevalent in scramjet engine testing due to the high temperatures and high speeds of the flow.

### **2.3.2. Metacomp's CFD++**

Metacomp Technologies provides a wide range of software suites for modeling and simulating aero-acoustics, fluid dynamics, and solid structures. Specifically, Metacomp's computational fluid dynamics suite, CFD++, has the capabilities to simulate a wide variety of flows. The software is based on the following three unifying principles: unified-grid, unified-physics, and unified-computing. For large problems, CFD++ is able to efficiently run on thousands of CPU cores. The unified-grid principle allows for structured and unstructured grids to be used in both two- and three-dimensions. For two-dimensional cases, this includes quadrilateral and triangular cells. For three-dimensional cases, this includes hexahedral, tetrahedral, pyramid, and triangular prism cells [7].

#### **2.3.2.1. Governing Equations**

CFD++ offers the ability to solve a wide range of flow types. This requires various types of flow equations. The available flow equation sets are listed below:

- I. Compressible Perfect Gas Navier-Stokes/Euler
- II. Preconditioned Compressible Perfect Gas Navier-Stokes/Euler
- III. Compressible Real Gas Navier-Stokes/Euler
- IV. Preconditioned Compressible Real Gas Navier-Stokes/Euler
- V. Incompressible Navier-Stokes/Euler
- VI. Incompressible Multi-fluid Navier-Stokes/Euler

Flow Equation Sets II and IV are the same as I and III, respectively. The preconditioned equation option can improve convergence by altering the time evolution of the problem to avoid numerical issues. These issues arise when numerical schemes for the compressible flow equations encounter a low velocity fluid. This causes a large difference in the eigenvalues of

Jacobian, which makes the numerical diffusion and the condition number to become very large. Preconditioning of the equations reduces the “stiffness” by multiplying the Jacobian by a matrix containing a preconditioning parameter. This parameter reduces the deviation of the Jacobian eigenvalues by reducing the speed of sound terms found in the eigenvalues. This effective speed of sound is much lower than the actual speed of sound.

For all CFD simulations performed for this work, Flow Equation Set I was used. The governing equations in three-dimensions can be written in conservation form as:

$$\frac{\partial \vec{Q}}{\partial t} + \frac{\partial}{\partial x}(\vec{F}_1 + \vec{G}_1) + \frac{\partial}{\partial y}(\vec{F}_2 + \vec{G}_2) + \frac{\partial}{\partial z}(\vec{F}_3 + \vec{G}_3) = \dot{S} \quad (1)$$

where  $\vec{Q}$  is the vector of dependent conservation variables,  $\vec{F}$  are the inviscid flux vectors,  $\vec{G}$  are the viscous flux vectors, and  $\dot{S}$  is the vector of source terms. The subscripts 1, 2, and 3 represent the three spatial directions. The dependent conservation vector and inviscid flux vectors are given as:

$$\vec{Q} = \begin{bmatrix} e \\ \rho \\ \rho u \\ \rho v \\ \rho w \end{bmatrix} \quad \vec{F}_1 = \begin{bmatrix} (e + p)u \\ \rho u \\ \rho u^2 + p \\ \rho vu \\ \rho wu \end{bmatrix} \quad \vec{F}_2 = \begin{bmatrix} (e + p)v \\ \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho wv \end{bmatrix} \quad \vec{F}_3 = \begin{bmatrix} (e + p)w \\ \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \end{bmatrix} \quad (2)$$

where  $e$  is the total energy,  $\rho$  is the mixture density,  $p$  is the pressure, and  $u$ ,  $v$ , and  $w$  are the velocity components in the  $x$ ,  $y$ , and  $z$  directions respectively. The viscous flux vectors are given as:

$$\vec{G}_1 = \begin{bmatrix} K \frac{\partial T}{\partial x} - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} \\ 0 \\ -\tau_{xx} \\ -\tau_{xy} \\ -\tau_{xz} \end{bmatrix} \quad \vec{G}_2 = \begin{bmatrix} K \frac{\partial T}{\partial y} - u\tau_{yx} - v\tau_{yy} - w\tau_{yz} \\ 0 \\ -\tau_{yx} \\ -\tau_{yy} \\ -\tau_{yz} \end{bmatrix} \quad (3)$$

$$\vec{G}_3 = \begin{bmatrix} K \frac{\partial T}{\partial z} - u\tau_{zx} - v\tau_{zy} - w\tau_{zz} \\ 0 \\ -\tau_{zx} \\ -\tau_{zy} \\ -\tau_{zz} \end{bmatrix} \quad (4)$$

where  $K$  is the thermal conductivity of the mixture,  $T$  is the temperature, and  $\tau_{ij}$  terms are the fluid stresses and strains. The source term vector is typically zero, unless CFD++ is run with the Eulerian Dispersed Phase option, the P-1 Radiation model, a Porosity boundary condition, or any of the physics source terms options. For this work, the volumetric source term option within the physics source terms was used to provide heat addition to the flow. The volumetric source term option allows the user to input a given mass, body force, and energy to the flow [7].

The fluid stresses and strains seen in Equations 3 and 4, are given as:

$$\begin{aligned} \tau_{xx} &= 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\ \tau_{yy} &= 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\ \tau_{zz} &= 2\mu \frac{\partial w}{\partial z} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\ \tau_{xy} &= \tau_{yx} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \tau_{xz} &= \tau_{zx} = \mu \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\ \tau_{yz} &= \tau_{zy} = \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \end{aligned} \quad (5)$$

where  $\mu$  is the dynamic viscosity of the mixture. Next the pressure is coupled to the density and temperature using the following equation of state derived from the perfect gas equation of state:

$$p = (\gamma - 1) \left( e - \frac{1}{2\rho} ((\rho u)^2 + (\rho v)^2 + (\rho w)^2) \right) \quad (6)$$

where  $\gamma$  is the ratio of specific heats. To complete the governing equations, the perfect gas equation of state is used to derive temperature using the following equation:

$$T = \frac{p}{\rho R} = \left( \frac{e}{\rho} - \frac{(\rho u)^2 + (\rho v)^2 + (\rho w)^2}{2\rho^2} \right) \frac{(\gamma - 1)}{R} \quad (7)$$

where  $R$  is the gas constant [7].

### 2.3.2.2. Turbulence Modeling

For this work, the realizable  $k - \varepsilon$  turbulence model was used. The  $k - \varepsilon$  model is a very commonly used turbulence model in Reynolds-Averaged Navier Stokes (RANS) flow simulations. The realizable model introduces modifications to account for certain known physical properties of the stress tensor. This is done by bounding the magnitude of the predicted tensor components. As a result, this improves predictive accuracy and stability [7]. To algebraically obtain Reynolds-stresses from the modeled eddy viscosity,  $\mu_t$  and the available mean-strain tensor, the Boussinesq relation is used as follows:

$$\rho \overline{uv} = \frac{2}{3} \delta_{xy} \rho k - \mu_t S_{xy} \quad (8)$$

where

$$S_{xy} = \left( \frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} - \frac{2}{3} \frac{\partial \bar{w}}{\partial z} \delta_{xy} \right) \quad (9)$$

The transport equations for  $k$  and  $\varepsilon$  are as follows:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial}{\partial y} (\bar{v} \rho k) = \frac{\partial}{\partial y} \left[ \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial y} \right] + P_k - \rho \varepsilon \quad (10)$$

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \frac{\partial}{\partial y}(\bar{v}\rho\varepsilon) = \frac{\partial}{\partial y}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial y}\right] + (C_{\varepsilon 1}P_k - C_{\varepsilon 2}\rho\varepsilon + E)T_t^{-1} \quad (11)$$

where  $P_k$  is the rate of production of turbulence energy and is defined as:

$$P_k = -\rho\bar{u}\bar{v}\frac{\partial\bar{u}}{\partial y} \quad (12)$$

Additionally,  $T_t$  is the realizable estimate of the turbulence timescale:

$$T_t = \frac{k}{\varepsilon}\max\{1, \zeta^{-1}\}, \quad \zeta = \sqrt{\frac{R_t}{2}} \quad (13)$$

with the turbulence Reynolds number as:

$$R_t = \frac{\rho k^2}{\mu\varepsilon} \quad (14)$$

The additional term,  $E$  in the dissipation-rate equation provides improved model response to adverse pressure-gradient flows. This term is defined as:

$$E = A_E\rho\sqrt{\varepsilon T_t}\Psi\max\left\{\frac{1}{k^{\frac{1}{2}}}, (\nu\varepsilon)^{\frac{1}{4}}\right\}, \quad \Psi = \max\left\{\frac{\partial k}{\partial y}\frac{\partial\tau}{\partial y}, 0\right\}, \quad \tau = \frac{k}{\varepsilon} \quad (15)$$

The eddy viscosity,  $\mu_t$  is obtained from:

$$\mu_t = \min\left\{\frac{C_\mu f_\mu \rho k^2}{\varepsilon}, \frac{2}{3}\frac{\rho k}{S}\right\} \quad (16)$$

where  $S$  is the dimensional strain magnitude defined as:

$$S = \frac{S_{kl}}{\sqrt{2}} \quad (17)$$

and  $f_\mu$  is a low-Reynolds number function to account for viscous and inviscid damping of turbulent fluctuations near solid surfaces:

$$f_{\mu} = \frac{1 - e^{-0.01R_t}}{1 - e^{-\sqrt{R_t}}} \max \left\{ 1, \left( \frac{2}{R_t} \right)^{\frac{1}{2}} \right\} \quad (18)$$

Lastly, the constants for the model are given by:

$$C_{\mu} = 0.09, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_{\varepsilon} = 1.3, \quad A_E = 0.3 \quad (19)$$

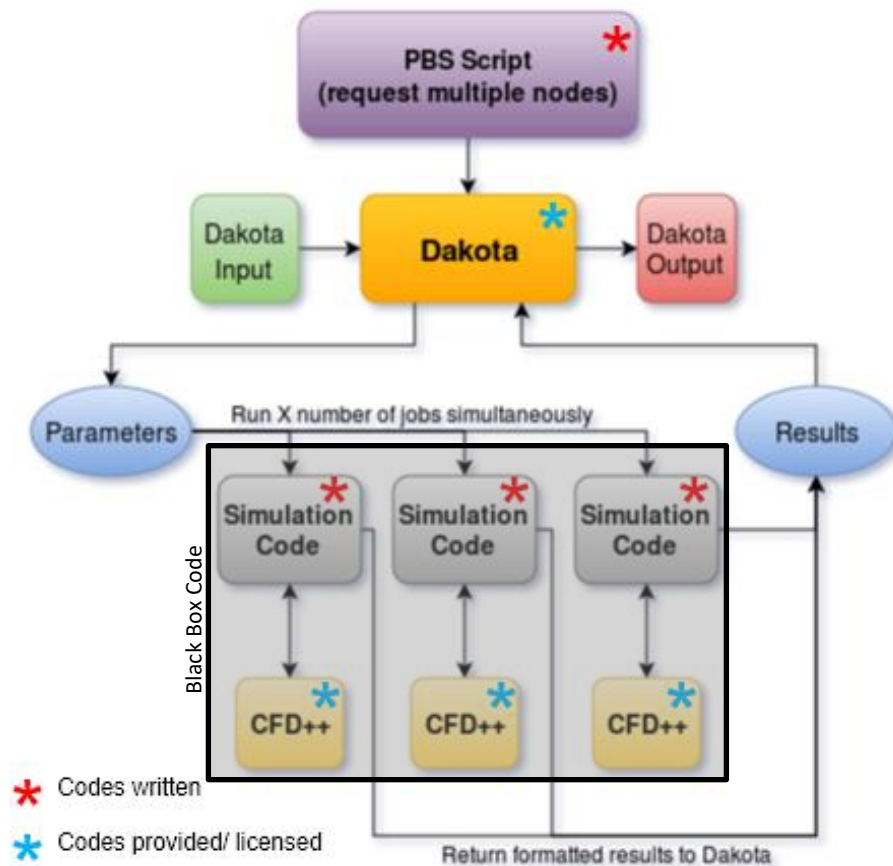
The realizable  $k - \varepsilon$  model is either integrated directly to walls or used in conjunction with wall function [7]. For this work, the grid has been refined at the wall to allow for the  $k - \varepsilon$  model to be integrated directly to the wall, also known as solving to the wall.

### 2.3.3. Dakota Toolkit

Dakota is a project led by Sandia National Laboratories to provide an optimization toolkit that can be used for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. The toolkit can easily be interfaced with simulation codes of all types. Dakota's intended use targets the computational methods used in the engineering field to enhance analyses. Specifically, Dakota enables the results to achieve a broader impact rather than simple single-point solutions to simulations. With this, questions like "What is the best design?", "How safe is it?", and "How much confidence do I have in my answer?" can be addressed. These types of questions are represented in the optimization problem by an objective function. In general, Dakota is used to either minimize or maximize the objective function by varying a set of design parameters. For example, if the answer to "How safe is it?" is desired, the calculated factor of safety would be the objective function. Dakota then is used to maximize the objective function (factor of safety). Dakota is an on-going effort in which many areas of the toolkit are still being debugged and developed [8]. Therefore, only basic optimization methods and tools were used.



A simulation code is used to communicate with CFD++. This code acts as a “black box” code; it takes inputs, executes some specified code, and then returns an output. Dakota’s asynchronous capabilities allow for the use of the DoD Supercomputing Resource Center (DSRC). This allowed for each optimization evaluation to run CFD++ on its own set of computing nodes. Figure 8 illustrates the flow of the complete algorithm. The PBS Script is simply a code that is used to access the DSRC and is not a part of a standard Dakota execution. For this work the PBS script obtains a node from the DSRC and begins running Dakota. Once Dakota has initialized



*Figure 8: Complete Algorithm Flow Chart*

using the Dakota input file, parameters are sent out to the simulation code. This begins by setting up a working directory and applies the parameters to the CFD++ input file(s). Following this, the simulation code submits another PBS script unique to the working directory. This PBS script

obtains nodes for the CFD++ to run the specific evaluation. Once the CFD++ simulation has reached convergence, the results are returned to the simulation code to be formatted and sent back to Dakota. Once Dakota receives the results, another parameter file is sent out. This process is repeated until all Dakota evaluations have been completed. An example PBS Script, simulation code, Dakota input, and CFD++ input can all be seen in Appendix A.

### 2.3.3.1. Evolutionary Algorithm

Dakota provides a wide variety of optimization methods ranging in complexity. Initially, an Evolutionary Algorithm (EA) was used to build the overall CFD-optimization routine. An EA was selected because they are very well documented and required minimal background knowledge to get started. Dakota offers the following EA's: The Common Optimization Library INterface (COLIN), Single Objective Genetic Algorithm (SOGA), Multi-Objective Genetic Algorithm (MOGA). All of these methods follow the same theory behind EA's, which is based on survival of the fittest. To begin, a population of randomly selected design points in the parameter space is produced. These values are uniquely represented in the population by forming a "genetic string." Following the evaluation of these points, the EA selects the best design points from the population, based on a fitness function. The fitness function is defined as:

$$D = \sqrt{\sum_{j=1}^m \{\max[0, g_j(\bar{x})]\}^2 + \sum_{k=1}^l |h_k(\bar{x})|^2} \quad (20)$$

where  $\bar{x}$  is the design variable vector,  $m$  is the number of inequality constraints, and  $l$  is the number of equality constraints [9]. These points are considered to be the most "fit" and are known as the parents of the population. Next, the EA carries out mathematically simulated evolutionary processes such as natural selection, breeding, and mutation. The breeding stage is

commonly known as the crossover stage and uses two parents to generate a number of children defined by the user. The arithmetic crossover is defined by two equations. For this example, the user has selected two children per crossover:

$$Child1 = (r)(Parent1) + (1 - r)(Parent2) \quad (21)$$

$$Child2 = (1 - r)(Parent1) + (r)(Parent2) \quad (22)$$

where  $r$  is  $U[0,1]$ , which ensures the children are the result of a convex combination of the parents. The next process involves randomly selecting children from the crossover for mutation. The mutation process is random bit mutation. This involves randomly selecting a design variable and switching its bit makeup. For example, if there is a 0 then it becomes a 1 and vice versa. The total number of mutations is determined by the mutation rate selected by the user [9]. This evolutionary process produces a new population to be evaluated. The process is repeated to identify a design point that represents an optimal objective function [10]. This type of optimization method tries to learn as it works through the design space. The repetitive evolution process, through learning and survival of the fittest, makes EA's a robust optimization method. However, being robust doesn't mean the method will be efficient. In some cases, the EA will find the optimal solution, but it will take much more computational power and time than another optimization method.

The efficiency of the EA can often be improved if there is knowledge about the optimal solution. With some knowledge, the EA can be started with the initial population being selected, or seeded, from a "region" of design parameters to ensure a high number of "fit" design points to be reproduced in the evolving populations that follow. Additionally, a penalty function can be applied to EA. This function is a formulation to penalize infeasible designs to improve the

population reproduction. Typically, the penalty function is simply a multiplier that is applied to the design points in the fitness function to ensure this point is not selected to be a parent in the next population. For simplicity, the default Dakota settings for the EA were used. This includes a mutation rate of 0.08 and 50 parents.

Referring back to Figure 8, when Dakota is initiated, the input file is read in to setup the optimization environment. This includes the design parameter(s), the objective function(s), the optimization method, and the various options for the method selected. When an EA method is selected, Dakota produces the initial population. These parameters are then passed to the simulation code to process the parameters and format the input file of CFD++ with these parameters. The simulation code then executes the CFD++ run. Once CFD++ has completed the evaluation, the simulation code processes the CFD++ results, extracts the needed outputs, and returns the set of results to Dakota. Dakota then creates the new population, or set of parameters, based on the previous results.

#### **2.3.3.2. Morris One-at-a-Time**

The second optimization method used in this work was the Morris One-at-a-Time (MOAT) algorithm. It was produced by Lawrence Livermore National Laboratory as a part of the Problem Solving Environment for Uncertainty Analysis and Design Exploration (PSUADE) toolkit. The PSUADE toolkit can be used for metamodeling, sensitivity analysis, uncertainty quantification, and optimization. The MOAT method is one of the many sampling optimization methods offered in the PSUADE toolkit and was originally proposed by Max D. Morris. The method is specifically designed for evaluating input variables of computational models and their effects on the outputs. This is done by varying one design parameter at a time to create a sample of its elementary effects [10].

Generally, MOAT is a design and analysis of computer experiment (DACE) which is a more modern version of a design of experiment (DoE). Both of these target extracting trend data from a parameter spaces with limited sample points. However, DACE methods neglect the component of non-repeatability due to the consistency of computer simulations. The MOAT method creates an evaluation grid based on  $n$  design parameters and  $P$  partitions. For simplicity, it is assumed the design parameters are normalized and range from zero to one. The number of partitions must be odd and is selected by the user. The size of the evaluation grid is the number of partitions plus one multiplied by the number of design parameters. Each design parameter may take on the following values:

$$\{0, \frac{1}{p}, \frac{2}{p}, \dots, \frac{p-1}{p}, 1\} \quad (23)$$

This evaluation grid represents all of the possible values for each design parameter. For the case of normalized parameters, all columns will be the same. To create a series of function evaluations, a random permutation matrix containing ones and zeros is used to select points from the grid. Each column in the permutation matrix contains one element equal to one with no two columns with the ones in the same position. This process is repeated to meet the number of samples, or function evaluations selected by the user. As a result, the equal probability of the random permutation matrices can result in duplicate columns [11]. However, Dakota checks for duplicates to prevent repeated simulations.

For a sampling method, Dakota will fill the design space and send all of the parameters at once. Then, once CFD++ simulations have completed, the simulation code sends the results back to Dakota and inputs the next set of parameters for CFD++. For both methods, an EA and sampling

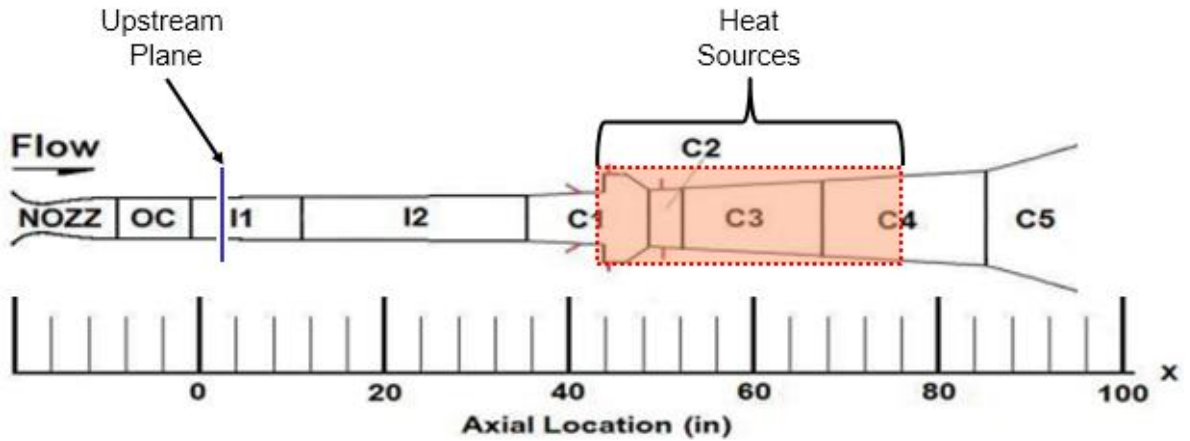
method, Dakota returns a set of output files once all results have been obtained and processed by Dakota.

### **3. METHODOLOGY**

To integrate the Dakota toolkit with CFD++, a well-developed CFD model is required. To simplify the process, only one geometry will be used as the integration can be applied to various CFD models that obtain similar parameters. Initially, the Divergent with Step geometry was selected to run initial studies because it required less time to produce a grid. However, for all sampling method optimization studies, the Fully Divergent setup was used. This geometry results in a more simple flow by removing the step. The step causes circulating flow regimes and restricts the expansion of the flow following combustion. In turn, this would require a more in-depth grid sensitivity study which is not favorable, as the focus of this work is on the integration of CFD and optimization tools. Additionally, the complexity of running a full CFD simulation of a scramjet engine is considered and was addressed by simplification of the combustion. The model was simplified by removing the fuel injection and replicating the combustion process with a variety of heat sources to provide the heat addition the flow would experience through the combustion process. The heat source is achieved with the volumetric source, a physics tool provided within CFD++. This feature allows for any number of sources to be added to the domain. The user defines the volume size and location of the source through a span of x, y, and z locations. Therefore, this restricts the volumetric source to a cuboid. The volumetric source feature also requires the user to provide an input file for each source. Within this input file, the user can define the following parameters as a function of time: total energy, mass, x-momentum, y-momentum, and z-momentum. All reference quantities used were from Case F09175AK, which corresponds to an inlet condition of Mach 1.8.

### 3.1. Geometry

Although both geometries were modeled in Metacomp's CFD++, only one of the two RC22 geometries were refined for a sensitivity study. The geometries are very similar and can be analyzed at a later time. The two similar geometries can be seen above in Figure 7. The difference in the geometries begin in the aft combustion section, specifically combustion sections 3 and 4 (C3 and C4). At the junction of C2 and C3, the second configuration introduces a step to match the diameter of C5. The Mach number at an x-location in the isolator was used as a constraint to prevent the optimization code from allowing results that resemble un-start conditions. The fully divergent geometry with the upstream plane for the Mach measurement and the location of the heat sources can be seen in Figure 9 below. Additionally, the nozzle (NOZZ) and the optical calibration (OC) section were removed. The nozzle and OC section were modeled



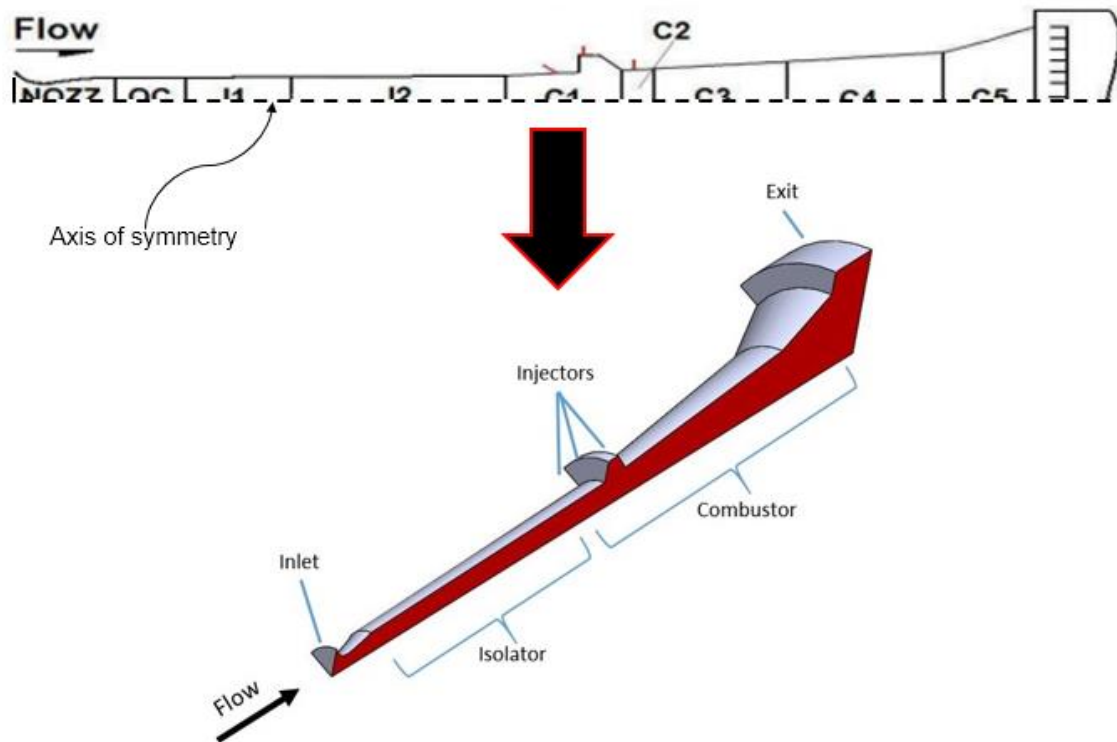
*Figure 9: Location for Mach Number Extraction & Location of Heat Sources*

separately to obtain a flow profile that was applied to the inlet of isolator 1 (I1). Lastly, the circular geometry was reduced to a two-dimensional cross-section as seen in Figure 10. Figure

10 shows how the 2D-axisymmetric geometry can be rotated into the full three-dimensional geometry.

### 3.2. Grid Sensitivity

For accurate modeling of the flow field and the surface interaction, a grid sensitivity study was



*Figure 10: 2D Axisymmetric to Rotated 3D Geometry*

performed. When solving to the wall, the grid must be refined near the wall to capture the viscous interaction with the wall to ensure the proper boundary layer is calculated. The grid sensitivity study is typically a multi-step process involving various grid refinements following by the evaluation of the different solutions obtained by the different grids. For this work, the computational time required to obtain a solution was considered in addition to the solution itself. The focus of this work is the integration of the Dakota toolbox and CFD++, therefore a balance between computational time and computational accuracy is desired. The baseline grid used contained approximately 22,500 cells with the cells at the core of the flow (near axis of



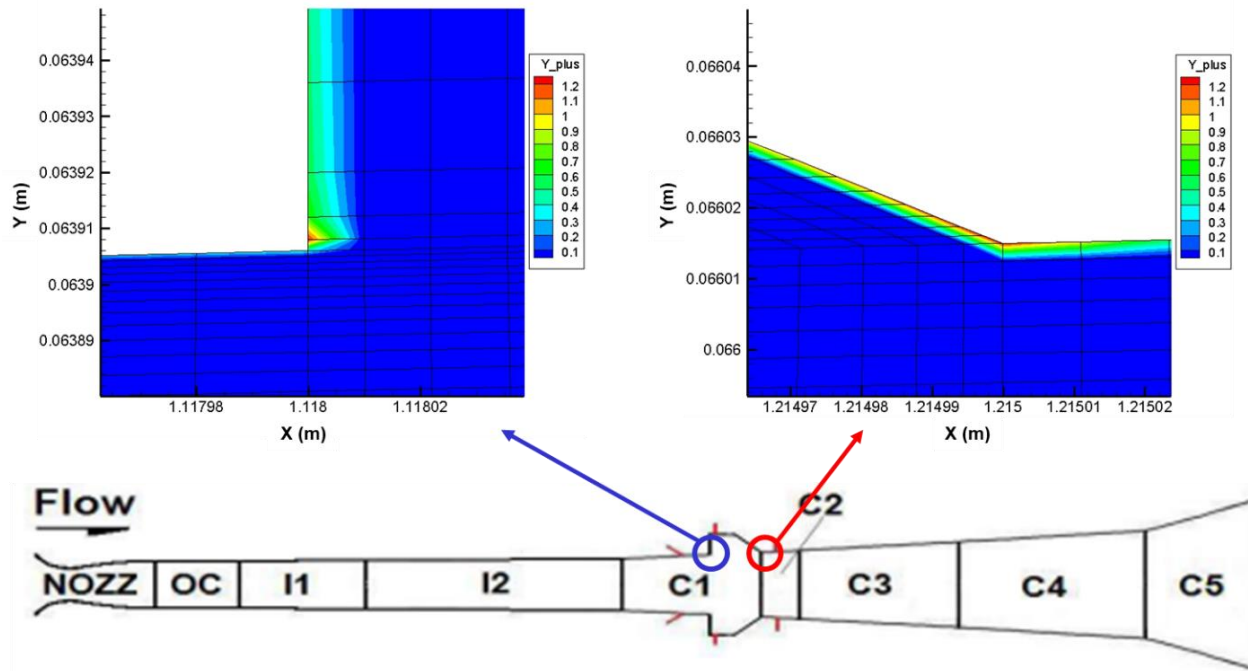
symmetry) having equal lengths in the axial and radial direction. Additionally, the maximum growth ratio of any cell in any direction does not exceed two. For example, starting at the wall, the cells increase in size in the negative radial direction. From one cell to another, the larger cell's radial dimension will be no larger than two times the radial dimension of the smaller adjacent cell. Furthermore, this baseline grid was to result in a  $Y^+$  value near one. Typically, a value of less than one is desired when solving to the wall, but for this case, this is just a baseline grid. The  $Y^+$  value is a non-dimensional wall distance used to define grid spacing for flows bounded by a wall.  $Y^+$  is defined as:

$$y^+ \equiv \frac{u_* y}{\nu} \quad (24)$$

where  $y$  is the distance from the wall and the nearest node,  $\nu$  is the kinematic viscosity, and  $u_*$  is the friction velocity at the nearest wall, which is defined as:

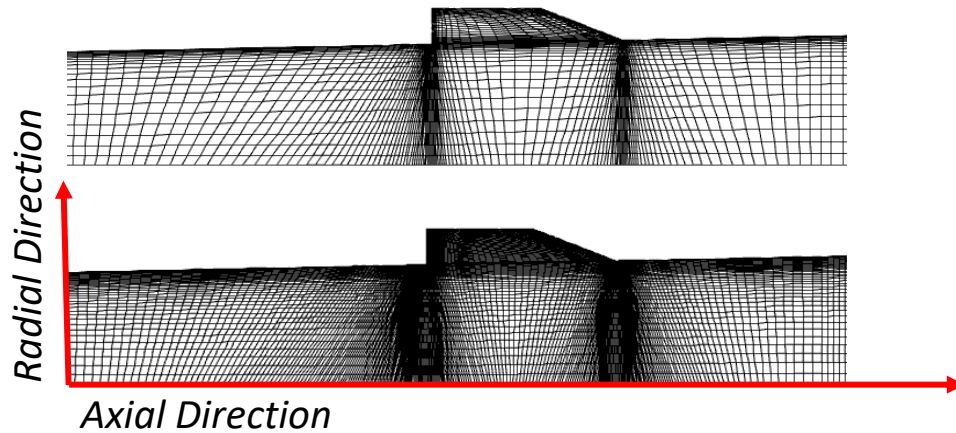
$$u_* \equiv \sqrt{\frac{\tau_w}{\rho}} \quad (25)$$

The baseline grid created resulted in a maximum  $Y^+$  value of 1.3. There were two locations in the grid where spikes in  $Y^+$  values occurred. The two locations were at the beginning of the combustion chamber (C1). Contour plots shown in Figure 11 show the two locations of increased  $Y^+$  values.



*Figure 11: Locations of Increased  $Y^+$  Values*

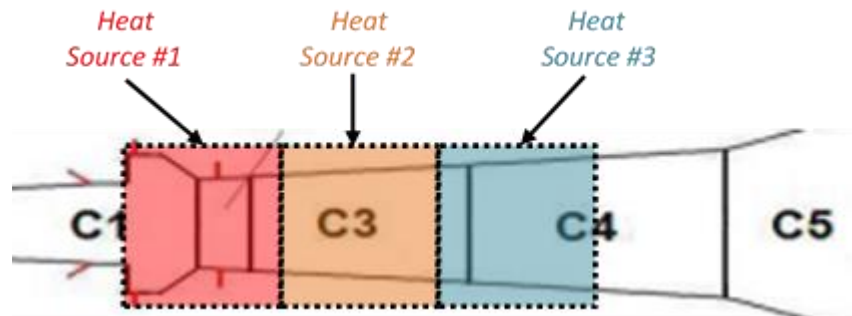
Following the creation of the baseline grid, three additional grids were produced: 2x axial direction refinement, 2x radial direction refinement, and 2x global (radial and axial directions) refinement. The baseline and 2x globally refined grids can be seen below in Figure 12.



*Figure 12: Baseline Grid (Top) and 2x Globally Refined Grid (Bottom)*

It can be seen in Figure 12 above, the Fully Divergent geometry was used in this grid refinement study. Additionally, three heat sources were applied to the combustion cavity and diverging

combustion chamber. Heat sources one, two, and three were set to 1 MW, 2 MW, and 2 MW, respectively. The heat source configuration can be seen below in Figure 13.



*Figure 13: Grid Sensitivity Heating Configuration*

Each grid was run until convergence was achieved. The convergence was based on the residual of the system's energy. For this sensitivity study, the convergence criteria was ten orders of magnitude drop of the energy residual. For all optimization and calibration studies, the convergence criteria used was five orders of magnitude drop of the energy residual after the restart of the solution. Once all four solutions were obtained, the contour plots of Mach number and pressure were compared to evaluate any irregularities from one grid to another. The comparison of the contour plots is shown below in Figure 14. The Mach number contours show little to no global variation from one grid to another. The pressure contours show little variation as well. However, there is some differences in the shock located in the isolator. The primary difference is the location of the shock. The shock can be seen at the discontinuity of pressure illustrated by the light blue contour abruptly changing to green in the isolator. Additionally, the shape of the pressure discontinuity is different at the wall. This illustrates the boundary layer interaction with the standing shock in the isolator.

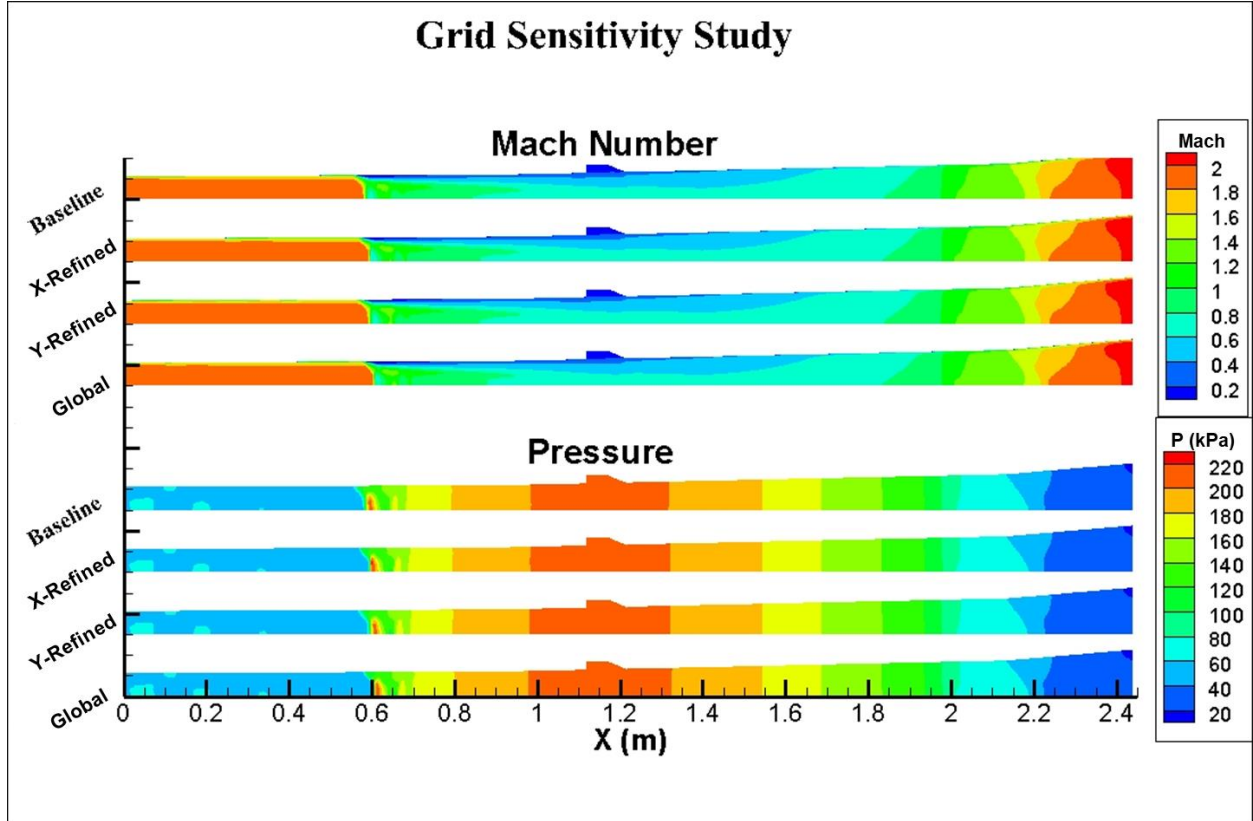


Figure 14: Grid Sensitivity Contour Comparison

Overall, there are no global differences in the simulations on the different grids. However, the optimization studies will be selecting individual values for objective functions. Therefore, the shock location, thrust production, and the computational time were all evaluated and compared in Table 1 below.

Table 1: Grid Sensitivity Study

Grid	Number of Cells	Shock Location (m)	Thrust (N)	Time on 10 Nodes (Minutes)	Difference in Shock Location (mm)	% Difference
Baseline	22.5k	0.5752	5645	2.45	24.8	4.23
Refine X	45k	0.5865	5643	5.82	13.5	2.30
Refine Y	45.5k	0.5854	5643	4.40	14.6	2.49
Global	91.5k	0.6000	5641	5.42	REFERENCE	

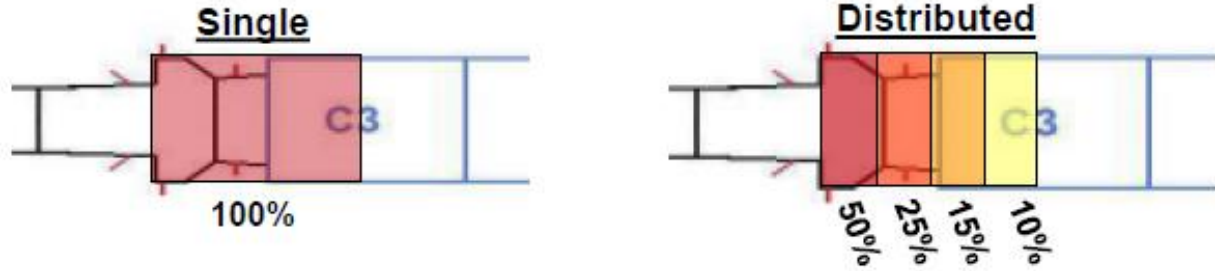
For all four of the grids, the thrust production showed strong agreement. However, there was a significant difference in the required computational time. This was a measure of time for the

solution to reach convergence on ten compute nodes. As expected, the baseline grid required much less time than the refined grids. The significant reduction of computational time is favorable for this work because multiple evaluations will be required to reach an optimal solution. The globally refined grid was used as reference to calculate the difference in the shock location in the isolator. This difference was then used to calculate a percent difference. The percent difference in the shock location showed to be minimal for the x-refined grid and the y-refined grid. However, the baseline grid showed almost double, when compared to the x-refined grid and the y-refined grid. For the fidelity of this work, a 4.23% difference from the reference is acceptable, considering the assumptions that are already being made. Additionally, the focus is on the integration of the Dakota toolbox with CFD++. For this reason, the baseline grid was selected to conduct the optimization studies discussed later.

## **4. RESULTS**

### **4.1. Initial Heat Source Study**

Before integrating the Dakota toolbox with CFD++, a simple heat source study was done to validate the results of the simulation. Two different heating configurations were used: a single fixed heat source and four fixed heat sources. The four source configuration was varied by taking a percentage of the total heat applied. From the first heat source to the last, the percentages used were 50%, 25%, 15%, and 10%. Each source in the distributed configuration was sized by taking the axial distance covered by the single configuration and dividing it into 4 equal sections. Both heating configurations can be seen below in Figure 15.



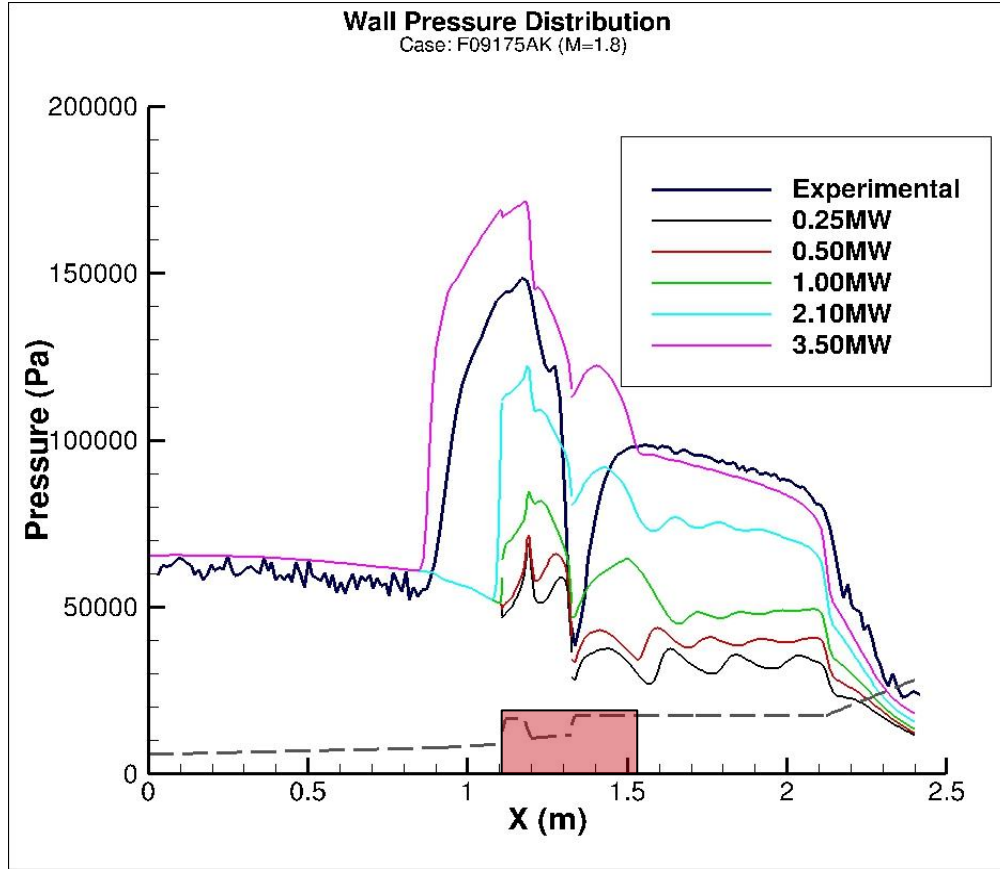
*Figure 15: Comparison of Heating Configurations*

In addition to the different configurations, the amount of heat addition was varied. The heat addition values used were 0.25 MW, 0.50 MW, 1.00 MW, 2.10 MW, and 3.50 MW. Table 2 is a summary of the different heating values for the distributed configuration.

*Table 2: Distributed Configuration Heating Summary*

Run #	Total Heat (MW)	Source 1 (MW)	Source 2 (MW)	Source 3 (MW)	Source 4 (MW)
1	0.25	0.125	0.0625	0.0375	0.025
2	0.50	0.25	0.125	0.075	0.05
3	1.00	0.5	0.25	0.15	0.1
4	2.10	1.05	0.525	0.315	0.21
5	3.50	1.75	0.875	0.525	0.35

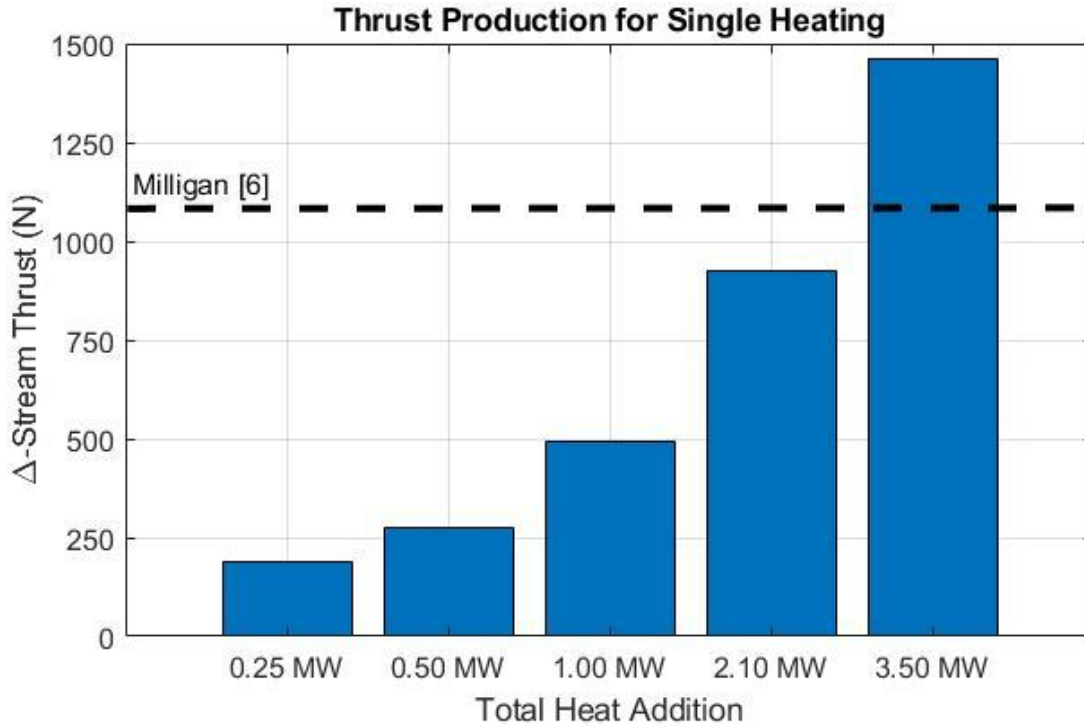
In order to validate the results, experimental wall pressure data from RC22 and stream thrust results from Milligan [6] were used as reference data. It is important to note that stream thrust is used in this study. Stream thrust is a fundamental performance parameter commonly used due to its accuracy when experimentally measured. The stream thrust represents the net momentum carried by a dynamic flow [12]. The stream thrust is a single mechanism contributing to the total thrust production and, therefore, is less than the total thrust. Each heating configuration was run for each of the total heat addition values and compiled to compare to the two reference parameters, wall pressure and thrust production. The single heat configuration wall pressure can be seen below in Figure 16.



*Figure 16: Wall Pressure Comparison for Single Heat Configuration*

Comparing the different amounts of heating, the wall pressure in the isolator is the same upstream of the standing shock. As expected, the highest heating results in the strongest and most forward shock. Additionally, the higher heating corresponds to higher wall pressure, which is also expected for the simulation. By observation, the 3.50 MW heating shows good agreement with the experimental shock location. However, the strength of the shock is stronger than that seen in the experiment. For this heating, the wall pressure at the step is significantly higher. This is a result of the lack of heating distribution within the combustion chamber. The flow downstream of the heating zone shows good agreement with the experimental data for 3.50 MW heat addition. Also, all the simulations show the sudden expansion of the flow that occurs at the step. However, the simulation, for all heat amounts, the amount of expansion is much less than

the experiment. This is caused by the single heat source, rather than realistic case of combustion where the heat addition is distributed. Next, the thrust production is compared to [6] and can be seen in Figure 17.



*Figure 17: Thrust Production Comparison for Single Heat Configuration*

Although 3.50 MW of heat addition shows agreement with the experimental wall pressure, it does not show agreement with the thrust production found by Milligan for the same geometry and conditions. Thrust for 3.50 MW shows a 29.2% difference where 2.10 MW shows only a 16.5% difference from Milligan. From the single heat configuration, it can be seen that the heat amount and location appears to agree with experimental data and other simulations performed. However, with the heat being uniformly distributed over a single source does not represent the combustion. This is seen in the expansion at the step in Figure 16 above.



Now, using the same amount of heat addition, the heat sources were broken down into four sections to be distributed, as discussed above. In doing this, the results show much better agreement with the experimental wall pressure. This can be seen below in Figure 18.

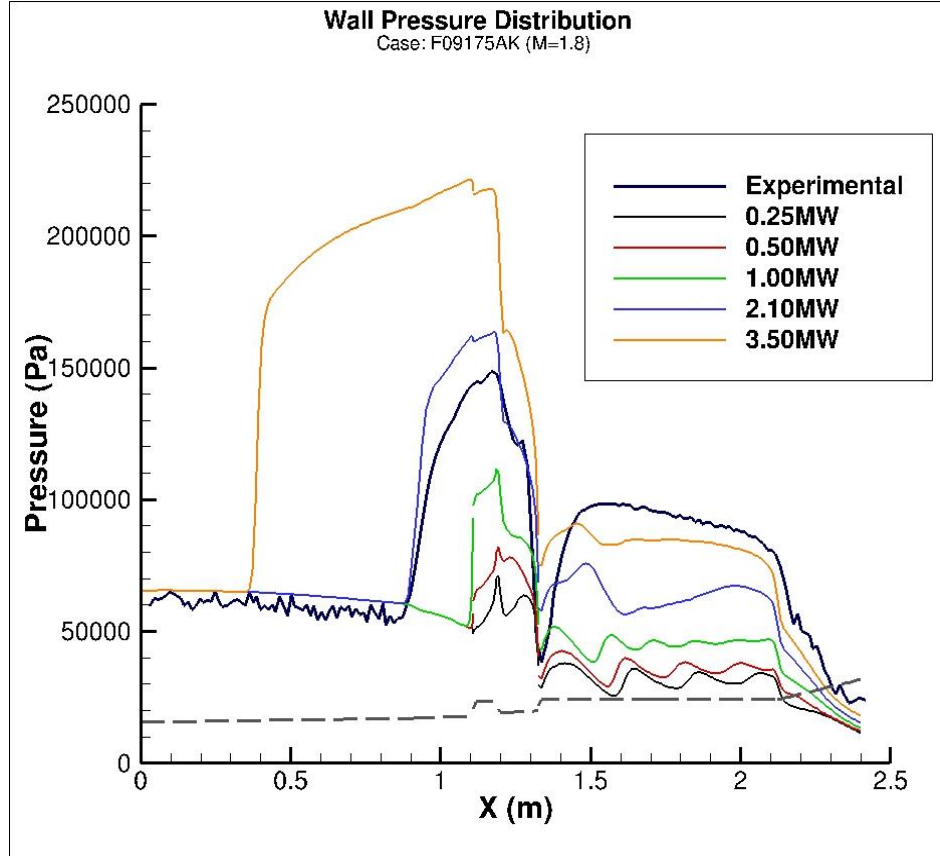
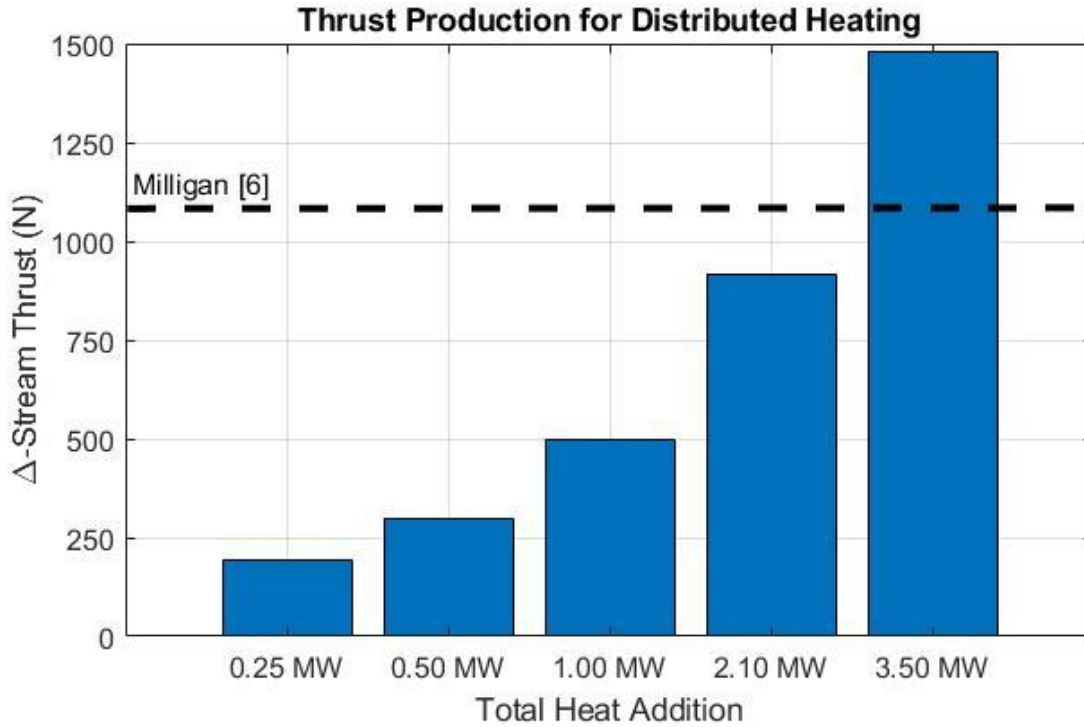


Figure 18: Wall Pressure Comparison for Distributed Heat Configuration

Similar to the single heat configuration, the isolator wall pressure is constant for all of the heat addition amounts with the shock location and strength changing. However, for the distributed heating, there is a much better agreement in the flow expansion at the step. More specifically, the magnitude and the lower bound of pressure at the step in the simulations match the experimental data much better. For this configuration, 2.10 MW heat addition shows the best agreement with the experimental wall pressure, in all aspects of the flow upstream of C3. The shock location is the same, but the strength is slightly high. Moving into the step, the expansion of the flow falls short compared to experimental data. Additionally, the compression throughout C3 and C4 is

less than what is seen in RC22. The effects of having a finite heat source can be seen at about 1.6 meters in the axial direction. At this location, the pressure drops significantly due to the end of the heat addition. Next, the thrust production can be analyzed and seen in Figure 19.



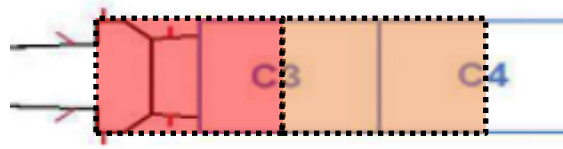
*Figure 19: Thrust Production Comparison for Distributed Heat Configuration*

Again, the heat sources are used to replicate the combustion for a simplified model to be used in the integration of CFD++ and Dakota.

#### **4.2. Fixed Heat Load Optimization-Genetic Algorithm**

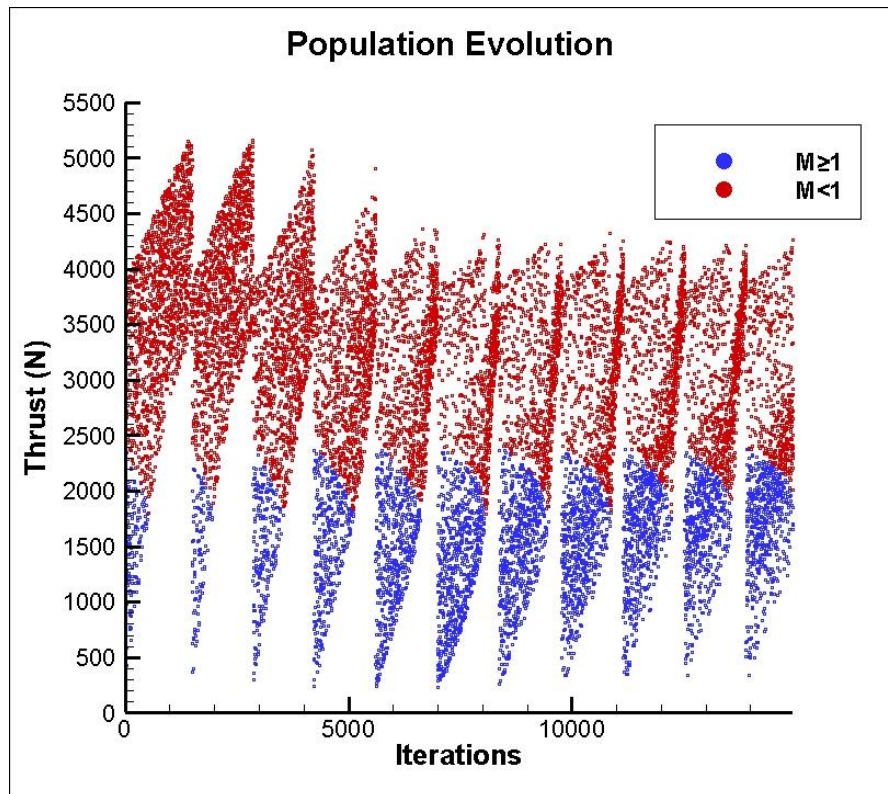
To begin the work with Dakota, a SOGA was used as an introductory study of coupling the optimization tool with CFD for scramjet applications. The objective of this study was to successfully integrate the Dakota toolbox with CFD++ to perform optimization studies. For this reason, the SOGA was selected because it is a robust method, requires minimal inputs, and is a suggested method for non-smooth, derivative-free global optimization problems. This allowed for the focus of the work to be put on the simulation code and understanding the strategy for job

management on the DSRC. The introductory study included one heating method: two variable heat sources. Each heat source was independently set and bounded from 2 MW to 20 MW. The two heat source configuration can be seen below in Figure 20.



*Figure 20: SOGA Two Variable Heat Sources*

The genetic algorithm performed 20,000 total evaluations across 11 different population sets. The fitness function of the SOGA targeted a unity Mach number extracted from the upstream plane in the isolator. However, the fitness function was weighted to drive the Mach number greater than one, as this will prevent the potential of unstart. Figure 21 below illustrates the evolution of the population.



*Figure 21: Evolution of SOGA Two Heat Population*

The initial population shows majority of the iterations resulting in an isolator Mach number less than one, indicated by red data points. However, as the populations evolve, they contain more solutions containing isolator Mach numbers of one or greater. Additionally, the evolution of populations illustrates a reduction in the upper bound of thrust. These higher thrust values correspond to lower Mach numbers at the upstream plane. Although there is an increase in acceptable Mach numbers ( $M \geq 1$ ) and a reduction of upper bound thrust, it is seen that after the 6<sup>th</sup> population set there is little to no variation of the following population sets. This is a result of the SOGA being a “brute force” style optimization routine. The sensitivity of the standing shock and the performance of the scramjet creates issues for the SOGA. To illustrate this sensitivity, Figure 22 compares the thrust production as the isolator Mach number varies.

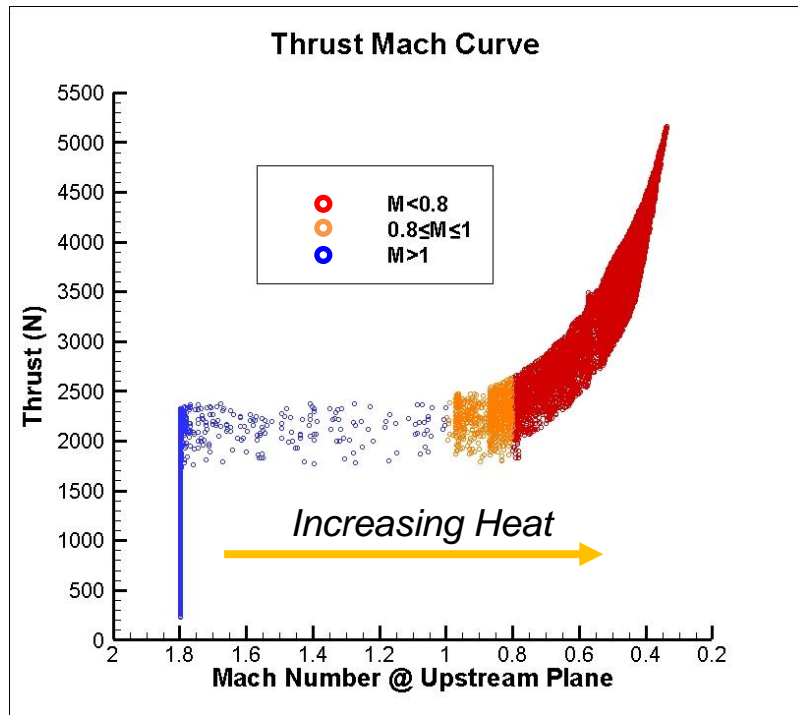


Figure 22: Isolator Mach Number Variation

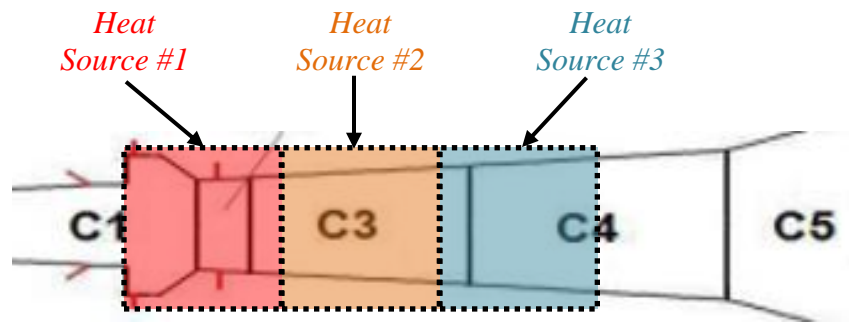
This plot includes all 20,000 iterations of the SOGA optimization, but breaks each data point into one of three groups. The x-axis is the Mach number extracted from the upstream plane and is inversely related to the amount of heat addition to the flow. There are various aspects of the

sensitivity of scramjet optimization. The first being at the lower bound of heating, which is shown as Mach 1.8. At lower amounts of heat addition, the flow in the isolator is unaffected and reflects the inflow Mach number. However, with the proper distribution of the heat addition, the geometry begins to produce thrust. The vertical line seen about Mach of 1.8 at the upstream plane is a result of increasing the amount of heat addition in the downstream heat source. This happens as a result of the flow being heated in the latter combustion chambers prior to diverging combustion chamber, where the flow is capable of expanding and producing thrust. This is the case for any vertical line in Figure 22. To summarize, the isolator Mach number may remain constant with an increase in thrust by distributing the heat addition downstream. As the total amount of heat addition increases, the thrust production remains, in general, constant. An increase in thrust is not seen until the standing shock is forced to or past the upstream plane. At this point, the Mach number at the upstream plane falls between 0.8 and 1, which is the range of accepted values to prevent unstart. Past this range, there is enough heat addition to press the shock further upstream, which also exponentially increases thrust production and the potential for unstart. The small group of orange data points illustrates the sensitivity of this optimization and the fine line of increasing performance, but preventing unstart conditions. Due to this sensitivity the current genetic algorithm was insufficient. Although, the algorithm may obtain optimal results, it requires a large number of evaluations and population evolutions, which is not ideal for this application.

#### **4.3. Preliminary Heating Optimization Study-Sampling Method**

To improve the efficiency and effectiveness of the optimization study, the optimization method was changed to the MOAT method. With a sampling method, the entire design space can be explored with a limited number of evaluations and is not dependent on a merit function for population evolution. Additionally, the MOAT method being a ‘one-at-a-time’ evaluation,

allows for sensitivity information of each design variable to be extracted while performing the optimization study. However, it is important to note that the study being performed is an optimization. Additionally, this study included a change in the scramjet geometry to the fully divergent configuration. This change was made to reduce the complexity of the flow field generated by the step configuration. The sharp corners at the step causes an increase in vorticity. Furthermore, the grid sensitivity study was performed on the fully diverging geometry as a result of the reduced flow field complexity. For these reasons, the fully divergent configuration is the preferred geometry to achieve the goals of integrating the optimization tools for scramjets. The fully divergent model for this study consisted of three heat sources, allowing each heat source to be set from zero to 6.5 MW. Figure 23 shows the placement of the three heat sources.



*Figure 23: MOAT Three Heat Sources*

This study included 1,000 function evaluations. Figure 24 shows each evaluation and its resulting thrust. The green circles are solutions that resulted in a Mach number greater than one at the upstream plane. These points represent poor performance and low resulting thrust. The blue circles are solutions that resulted in a Mach number less than one at the upstream plane. Although these solutions indicate higher resulting thrust, these solutions indicate unstart conditions.

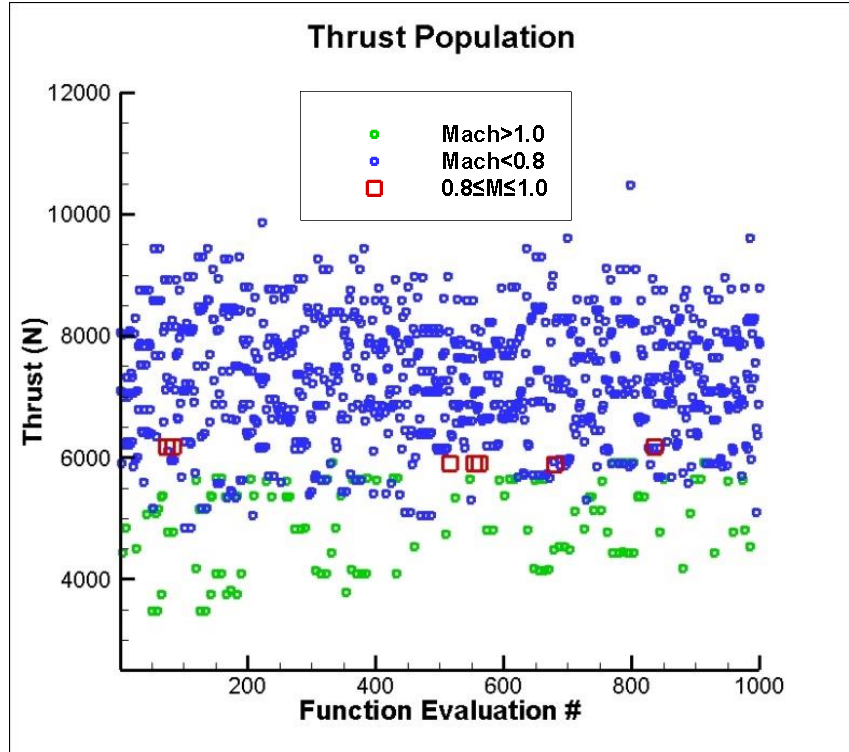


Figure 24: PSUADE MOAT Results

The best heating parameters, based on a balance of maximum thrust and minimum shock location difference to the upstream plane, were found to be at evaluation 516 and heat source #1, heat source #2, and heat source #3 were found to be zero, 2.889 MW, and 2.889 MW, respectively. To summarize the solutions show as red squares in Figure 24, Table 3 lists the amount of heat for each source, the total amount of heat addition, thrust production, and Mach number at the upstream evaluation plane. Duplicate solutions were neglected from the table. The row highlighted in yellow indicates the solution that Dakota selected to be the best.

Table 3: Summary of Optimal Solutions in Original Design Space

Source 1 (MW)	Source 2 (MW)	Source 3 (MW)	Total Heat (MW)	Thrust (N)	Mach
0.000	0.000	6.500	6.500	6175	0.818
0.000	2.889	2.889	5.778	5904	0.846
1.444	0.722	3.611	5.778	5886	0.801

As expected, the more heat added results in more thrust. Additionally, when more heat is added further downstream the thrust increases. The best solution selected by Dakota shows a balance of both thrust and shock location in the isolator. The selected solution is the second highest thrust while having the highest Mach number at the evaluation plane. The overlap of the blue and green circles shows how the distribution of heating affects the production of thrust and unstart conditions. In general, the overlapping points have similar amounts of heating, but different distributions of the heating. As a result, the standing shock in the isolator resides at a different location. Overall, the MOAT method performed well for the very large design space and sensitivity of the problem. The information from this large design space study can be used to refine the design space for further studies. By observation, Figure 24 shows the majority of the function evaluations resulting in upstream Mach numbers less than 0.8, which indicates there is too much heat addition. Therefore, the design space reduction targets the upper bound of each heat source. To determine, how the upper bound of each heating parameter was adjusted, Figure 25 shows the upstream Mach number plotted against the heat addition for each heat source.

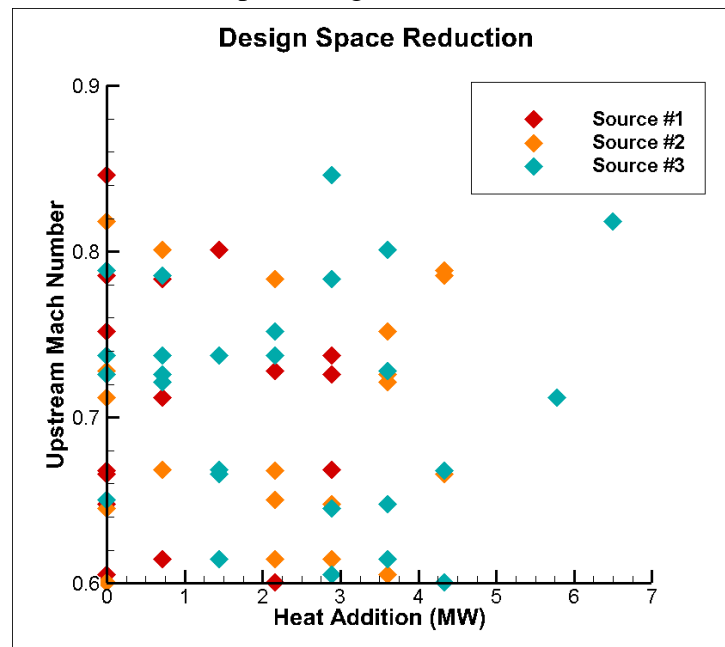
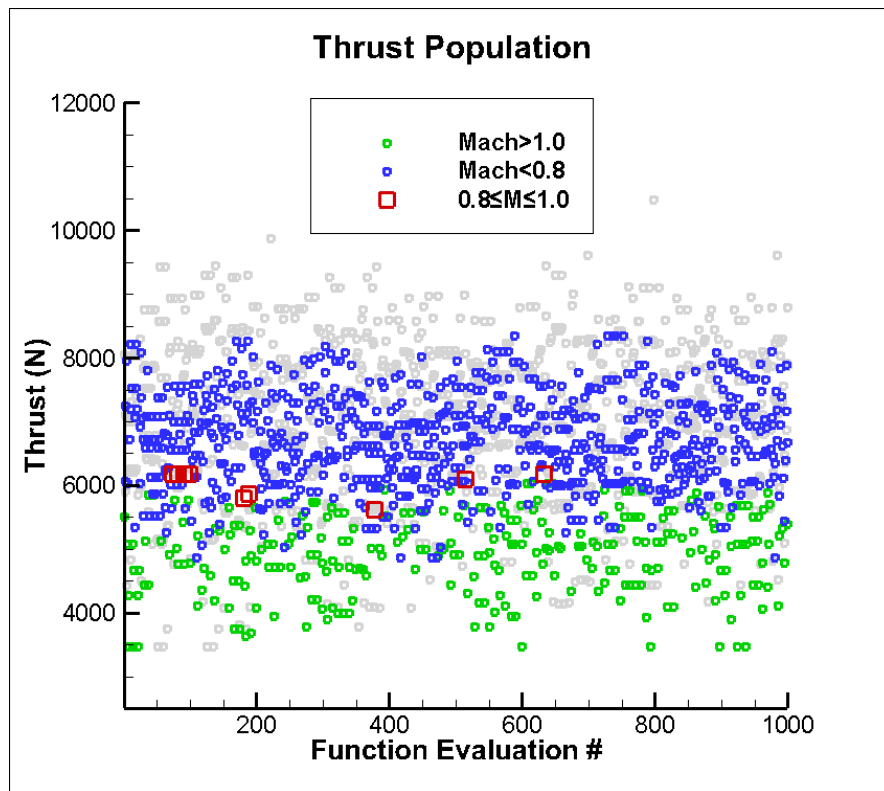


Figure 25: Reduction of Heat Source Design Space



First evaluating heat source #1, there highest heat addition that results in an upstream Mach number of 0.6 is 2.925 MW, or 45% of the original upper bound (6.5 MW). The highest heat addition for heat source #2 that results in an upstream Mach number of 0.6 is 4.55 MW, or 70% of the original upper bound. Lastly, heat source #3 shows an acceptable upstream Mach number with the upper bound set to 6.5 MW. Therefore, the upper bound of the heat addition was modified to 2.925 MW and 4.55 MW for heat sources #1 and #2, respectively, while heat source #3 remained the same. Again, the MOAT algorithm was ran for 1,000 function evaluations, but with the new heat addition bounds. Figure 26 shows the thrust production plotted against the function evaluation.



*Figure 26: Reduced Design Space MOAT Results*

The color of the points follow the same as Figure 24, however, the original thrust population has been added in the background and colored gray. The original points in the background allow for easy comparison to the new population. As expected, the reduced design space results in a more

dense population by lowering the upper end of thrust. The best function evaluation was found at evaluation number 181. This function evaluation resulted in 5,800 N of thrust and an upstream Mach number of 0.87. The heat addition values were 0.878 MW, zero, and 3.611 MW for heat source #1, #2, and #3, respectively. To summarize the solutions show as red squares in Figure 26, Table 4 lists the amount of heat for each source, the total amount of heat addition, thrust production, and Mach number at the upstream evaluation plane.

*Table 4: Summary of Optimal Solutions in Reduced Design Space*

Source 1 (MW)	Source 2 (MW)	Source 3 (MW)	Total Heat (MW)	Thrust (N)	Mach
0.000	0.000	6.500	6.500	6175	0.818
0.000	0.354	5.778	6.132	6098	0.829
0.731	0.354	3.611	4.696	5870	0.801
0.878	0.000	3.611	4.489	5801	0.867
0.731	1.416	1.444	3.591	5622	0.826

These solutions are from the optimization study using the reduced design space. Duplicate solutions were neglected from the table. The row highlighted in yellow indicates the solution that Dakota selected to be the best. Again, the most thrust is produced with the most total heat addition to the flow. Additionally, all solutions have most of the heat added to the third heat source. Dakota selected the forth solution in Table 4 due to the Mach number at the evaluation plane. Although this solution results in 5,801 N of thrust, the shock location shows a better balance for the optimal solution.

#### **4.4. Wall Pressure Calibration Study**

Similar to the previous study, the MOAT method was used in this study to start with a large design space to be reduced. However, this study's optimization objective is to calibrate the simulated wall pressure to the measured wall pressure in RC22. For the Fully Divergent

geometry, there are 151 pressure probes used to measure the wall pressure at various axial locations. The same CFD model was used from the previous study, but was configured to extract wall pressure at the same axial locations as the measurements made in RC22. Rather than having 151 objective functions in the optimization algorithm, the 151 pressures were reduced to a single term by calculating the root-mean-square (RMS) error. RMS error is often used as a measure of accuracy of different models and was calculated as follows:

$$RMS = \sqrt{\frac{\sum_{i=1}^{151} (p_{CFD,i} - p_{RC22,i})^2}{151}} \quad (26)$$

As the RMS value approaches zero, the CFD wall pressures approach the wall pressures measured in RC22. Therefore, the MOAT optimization was configured to minimize the RMS value for the objective function. The first design space used for this study allows each of the three heat sources to individually vary from zero to 6.5 MW. The design space included 1,000 samples to be evaluated. As expected, this resulted in multiple points far from the optimal solution. As in the previous study, the objective function was plotted against the heat addition for each heat source. This is used to determine the reduction of the design space for improved results of the optimization. Evaluating the lower RMS values in Figure 27, the upper bounds for heat sources 1 and 2 can be reduced to 3 MW. Additionally, the upper bound for heat source 3 can be reduced to 5.2 MW.

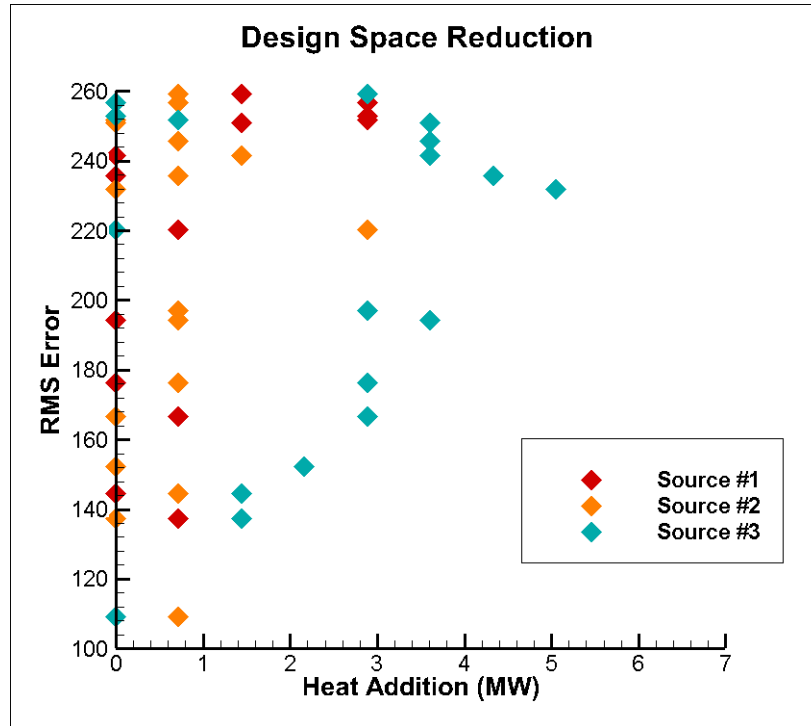


Figure 27: Wall Pressure Design Space Reduction

The optimal solution from this study was 0.72 MW, 0.72 MW and zero for heat source 1, 2, and 3, respectively. The resulting wall pressure can be seen below in Figure 28.

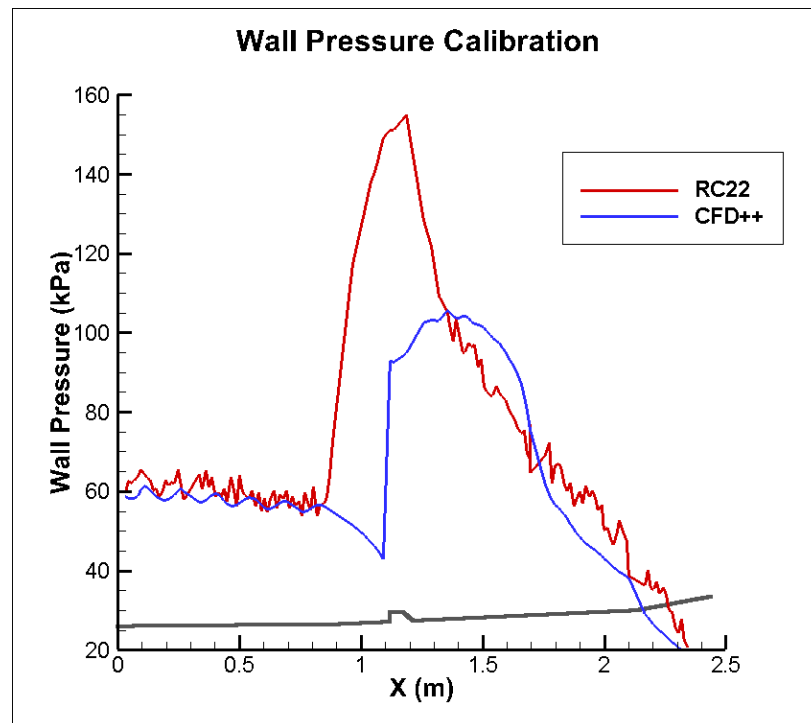
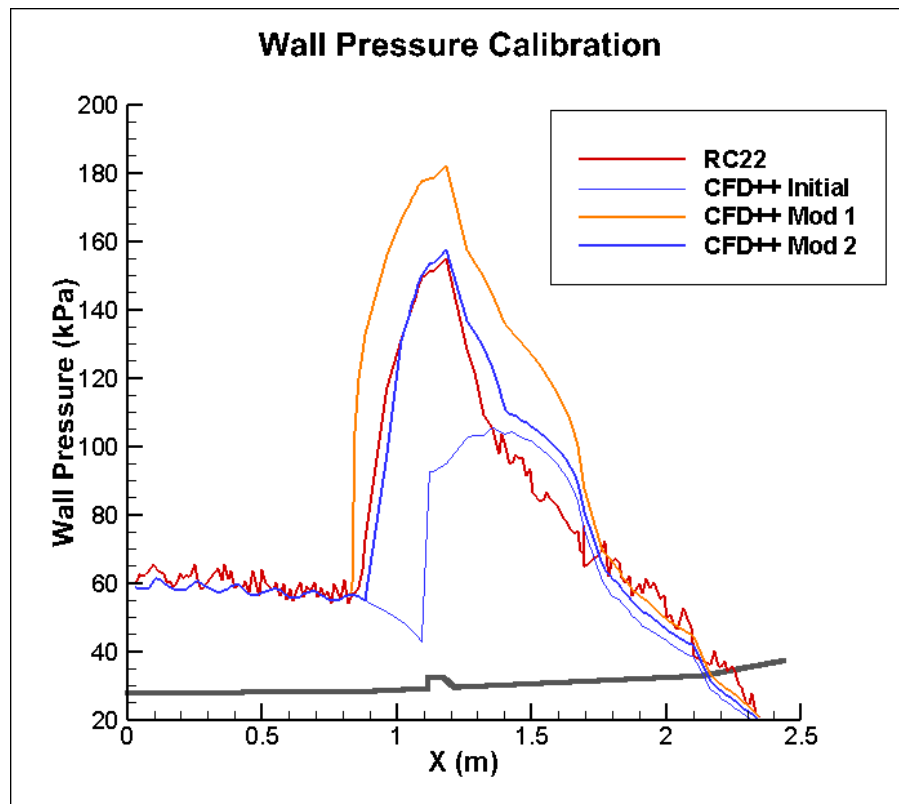


Figure 28: Initial Optimal Wall Pressure Calibration

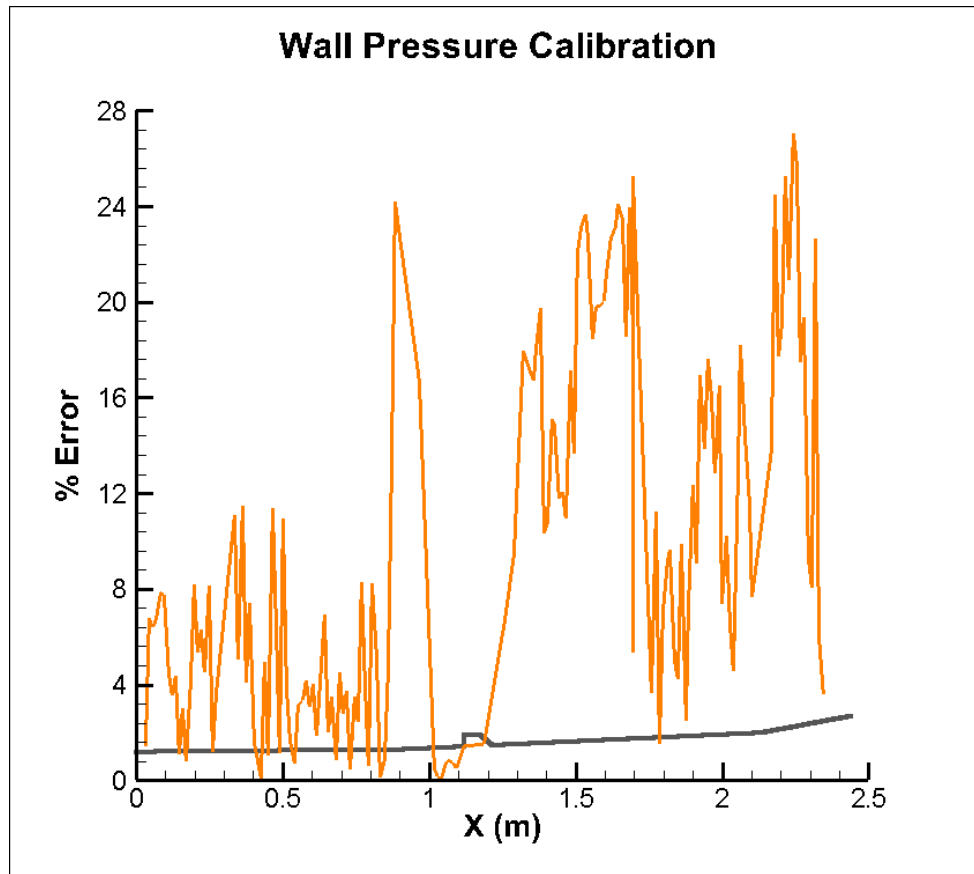
After reducing the design space and running the calibration study again, it was found that the RMS calculation could be reduced by neglecting the points upstream of the shock train for RC22 ( $X = 0.8 \text{ m}$ ). This was done due to the little variation found in the wall pressure throughout the isolator, upstream of the shock, for different amounts of heat addition. With the reduced parameters, the calibration was ran again to find the optimal solution to be 1.36 MW, 0.65 MW, and zero for the heat addition in source 1, 2, and 3, respectively. Figure 29 below shows that the revised calibration results in a much better optimal solution. The reduction of just the design



*Figure 29: Revised Optimal Wall Pressure Calibration*

space is shown in orange and labeled as Mod 1. The reduced design spaced and RMS calculation is shown in dark blue and labeled Mod 2. The revised calibration captures both the magnitude and the location of the spike in wall pressure following the shock train. There is some disagreement following the combustion cavity at the location of heat source 2. However, when calculating the percent error using the experimental wall pressure as reference, the majority of

the points fall under 15% error with the highest being 27 % error. The plot of the percent error calculated for the revised optimal calibration can be seen below in Figure 30.



*Figure 30: Revised Optimal Wall Pressure Error*

The increased percent error shown downstream of the combustion cavity is likely due to the use of heat sources rather than fuel injection and combustion. When using the heat sources, the heat addition is uniform across the source. In reality, the heat addition due to combustion is not uniform as there will be higher amounts of heat addition at locations where the air and fuel mixtures are nearest to stoichiometric combustion conditions. The large spike in percent error at axial position of 0.9 m is due to the small difference between the location of the pressure spike in the simulation and the experimental data. Overall, the revised parameters used for this calibration study resulted in an optimal solution that closely represents experimental wall pressures while using heat sources in the CFD model.

## 5. CONCLUSION

To address the current demand of hypersonic vehicles, efficient computational analysis capabilities are required. Current computational capabilities are available to provide accurate simulations. However, these capabilities are highly complex models that require large amounts of computing power and time. To enhance computational analysis of scramjet engines, Dakota was integrated with CFD++ to provide an optimization routine. This work focused on the integration of the two computational tools, which is used to evaluate a large design space rather than an individual case. Therefore, the routine utilized a simplified CFD model of a scramjet combustor tested at RC22. The simplified model was reduced to two-dimensional axisymmetric geometry. Additionally, the model replaced the scramjet's fuel injection and combustion with volumetric heat sources. These heat sources were used to replicate the addition of heat to the flow within the scramjet combustion chamber. The amount of heat added by each source was used as a design parameter in the optimization studies to maximize the production of thrust while restricting the shock train location in the isolator. Two optimization methods were used in this work: an evolutionary algorithm and a sampling algorithm. The evolutionary algorithm was used for earlier optimization studies to develop the routine. Although evolutionary algorithms exhibit robust optimization characteristics, it was found that this type of optimization method is insufficient for the application of scramjet engines. Initially, the parents selected produced improved populations. However, after three to four evolutions, the populations begin to produce similar function evaluations, which results in repeated populations. This is due to the sensitivity of the shock train within the isolator. Additionally, evolutionary algorithms perform best with some knowledge of optimal solutions. This work assumed large design spaces so that this routine can be applied to experimental designs where knowledge of optimal solutions may not be known. Therefore the Morris One-at-a-Time method, a sampling algorithm, was applied to the

optimization routine. This method samples the design space by varying one design parameter at a time. As a result, this method can also be used to extract design parameter sensitivities. This method showed to be applicable to the large design spaces used in this work by evaluating various points throughout the whole design space. Furthermore, the results of an initial optimization were used to quickly draw conclusions about the design space. This was then used to refine the design space to improve the number of optimal solutions. The same methodology was also used to evaluate the calibration of the heat sources to match the wall pressure of the scramjet. Initially, a large design space was used to minimize the difference between simulated wall pressure and experimental wall pressure data. As expected, the initial optimization study did not result in a match of wall pressures. However, by refining the design space and the objective function using knowledge from the larger space, the optimal solutions showed a better match to the experimental data. For scramjet optimization, the routine developed in this work is best used with a sampling algorithm. It requires little to no knowledge of the optimal solutions while providing details about the design space and each design variable. These details can then be used to improve the design space by narrowing the bounds of each design parameter. This knowledge of an unknown design space is particularly important to gain in new and experimental designs.

## **6. FUTURE WORK**

Advancements of this work can be made in multiple areas. The most important advancement would require the application of the optimization routine with a scramjet model with fuel injection and combustion. This will require a full three dimensional geometry to incorporate the mixing of fuel and air. With this model, the fueling can be used as the design parameters to maximize thrust. In doing this, the restriction of the shock train location in the optimization is removed. With combustion occurring in the model, unstart will occur and the thrust production will drop. Additional work can be done exploring optimization methods. A more advanced



optimization method may be required once the addition of combustion is added to the CFD model. If the simplified CFD model used in this method is desired, the number, the placement and the size of the volumetric heat sources could be studied. A study of the heat release for a full three dimensional combustion case can be used to locate areas of high heat addition to the flow. This would not only improve the accuracy of the simplified model, but also improve the calibration of the wall pressure. Lastly, with further use of the optimization routine, additional redundancies can be added to protect against the termination of the codes if errors occur.

## 7. REFERENCES

- [1] M. D. Parker, "Access to Space: Routine, Responsive and Flexible Implications for an Expeditionary Air Force," Air Command and Staff College, Air University, Maxwell Air Force Base, Alabama, April 1998.
- [2] *Report of the Commission to Assess United States National Security, Space Management, and Organization*, January 11, 2001.
- [3] J. E. Bradford, A. Charania, D. R. Eklund and J. Wallace, "Quickstat: A Two-Stage to Orbit Reusable Launch Vehicle Utilizing Air-Breathing Propulsion for Responsive Space Access," in *Space Conference and Exhibit*, San Diego, California, September 28-30, 2004.
- [4] D. Davis, "Introduction to Ramjets and Scramjets," AFRL/ RZAS, June 2010.
- [5] M. A. Hossain, M. T. Rahman, M. H. Soiket and S. Hossain, "Investigation and Improvement of Thermal Efficiency of Hypersonic Scramjet," in *ASME 2014 International Mechanical Engineering Congress and Exposition*, Montreal, Canada, November 14-20, 2014.
- [6] R. T. Milligan, J. Liu, C. J. Tamand, D. R. Eklund, M. A. Hagenmaier, D. L. Davis, D. J. Risha, M. Gruber and T. Mathur, "Dual-Mode Scramjet Combustor: Numerical Sensitivity and Evaluation of Experiments," in *50th AIAA Aerospace Sciences Meeting*, Nashville, Tennessee, January 09-12, 2012.
- [7] "CFD++ User Manual," Metacomp Technologies, INC., Agoura Hills, CA, 2013.
- [8] B. M. Adams, L. E. Bauman, W. J. Bohnhoff, K. R. Dalbey, M. S. Edeiba, J. P. Eddy, M. S. Eldred, P. D. Hough, K. T. Hu, J. D. Jakeman, J. A. Stephens, L. P. Swiler, D. M. Vigil and T. M. Wildey, "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual," Sandia Technical Report SAND2014-4633, November 2015.
- [9] J. Eddy and K. Lewis, "Effective Generation of Pareto Sets Using Genetic Programming," *ASME Design Engineering Technical Conferences*, no. DETC2001/DAC-21094, 2001.
- [10] B. M. Adams, M. S. Ebeida, M. S. Eldred, G. Geraci, J. D. Jakeman, K. A. Maupin, ... and A. Rushdi, "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.7 User's Manual," Sandia National Laboratories, SAND2014-4633, Nov. 2017.

- [11] M. D. Morris, "Factorial Sampling Plans for Preliminary Computational Experiments," *Technometrics*, vol. 33, no. 2, pp. 161-174, May 1991.
- [12] R. T. Milligan, "Dual Mode Scramjet: A Computational Investigation on Combustor Design and Operation," Wright State University, 2009.

## APPENDIX A: SCRIPT/ INPUT FILE EXAMPLES

### PBS Submission Script

An example of a PBS script running on the DSRC. This script is used to obtain a node for Dakota to run.

```
#!/bin/bash
#PBS -N PSM_Calib
#PBS -l walltime=120:00:00
#PBS -l select=1:ncpus=24:mpiprocs=24
#PBS -o PSM.o
#PBS -e PSM.e
#PBS -q standard
#PBS -V

cd $PBS_O_WORKDIR
rm -rf workdir.*
rm -f applicNodeFile.*
rm -f nodefile.txt
rm -f nodes.txt
rm -f Heat.txt
rm -f Update.txt
rm -f ResultsError.txt

export DATE=`date | cut -d ' ' -f -4 | tr ' ' '_' | tr ':' '_`
mkdir ${DATE} 2>/dev/null
mkdir PressureResults

export MPI_DISPLAY_SETTINGS=disabled

export CONCURRENCY=`grep concurr HeatCalibration.in | cut -d "=" -f2`

export APPLIC_PROCS=`grep 'mpiprocs' CFD_CONCURRENT_88.pbs | cut -d '=' -f 4`

if [ -f CFDpp_22k_Div_Calibration/mcpusin.bin.$APPLIC_PROCS ]; then
echo
else
cd CFDpp_22k_Div_Calibration/
module load cfd++
tometis kmetis $APPLIC_PROCS
cd ..
fi

DAK_RESTART=`find -iname dakota.rst -printf "%C@ %p\n" | sort -rn | sed -n '1p' | cut -d '/' -f 2-`

export EVAL_COMPLETE=0

if [ -z $DAK_RESTART ]; then
dakota -i HeatCalibration.in -w dakota.rst -o HeatCalibration.out > MaxHeatLoad.stdout
else
if [ -s $DAK_RESTART ]; then
dakota -i HeatCalibration.in -r $DAK_RESTART -w dakota.rst -o HeatCalibration.out >
MaxHeatLoad.stdout
else
dakota -i HeatCalibration.in -w dakota.rst -o HeatCalibration.out > MaxHeatLoad.stdout
fi
fi

sleep 10
mv HeatCalibration.out ${DATE}
mv HeatCalibration.dat ${DATE}
mv ResultsError.txt ${DATE} 2>/dev/null
mv dakota.rst ${DATE}
mv PressureResults ${DATE}

exit
```

## Evaluation Management Script

An example of the simulation script ran for an evaluation sent out by Dakota. This script is used to set the heat sources, begin the CFD++ simulation, and send the results back to the node running Dakota.

```
#!/bin/sh
# Simulator script for max heat load
#
#-----
#Create Temporary Working Directory
#-----
num=$(echo $1 | awk -F. '{print $NF}')
topdir=`pwd`
workdir=$topdir/workdir.$num
mkdir workdir.$num
cp $topdir/$1 $workdir/dakota_vars
cp -a CFDpp_22k_Div_Calibration/. $workdir
cp $DAKOTA_PARAMETERS_FILE $workdir
cp CFD_CONCURRENT_88.pbs $workdir
cd $workdir

# -----
# PRE-PROCESSING
# -----
EVAL_NUM=$(echo $DAKOTA_PARAMETERS_FILE | cut -c 11-)
# Find NonDimensional heats and calc heat values
NONDIMHEAT1=`sed -n '2p' $DAKOTA_PARAMETERS_FILE | cut -d 'h' -f -1`
NONDIMHEAT2=`sed -n '3p' $DAKOTA_PARAMETERS_FILE | cut -d 'h' -f -1`
NONDIMHEAT3=`sed -n '4p' $DAKOTA_PARAMETERS_FILE | cut -d 'h' -f -1`
HEAT1=`echo $NONDIMHEAT1 | awk {' printf $1 * 6500000 '}`
HEAT2=`echo $NONDIMHEAT2 | awk {' printf $1 * 6500000 '}`
HEAT3=`echo $NONDIMHEAT3 | awk {' printf $1 * 6500000 '}`
# Write heat1 to params.in
echo "${HEAT1} heat1" > params1.in
# Write heat 2 to params.in
echo "${HEAT2} heat2" > params2.in
# Write heat 2 to params.in
echo "${HEAT3} heat3" > params3.in
# Insert heating value to heat.box#
dprepro params1.in heat.box1.template heat.box1
dprepro params2.in heat.box2.template heat.box2
dprepro params3.in heat.box3.template heat.box3
export EVAL_NUM

# -----
# ANALYSIS
# -----
qsub -N Call.${EVAL_NUM} -V CFD_CONCURRENT_88.pbs

X=1
while [ "$X" -ne 0 ]
do
X=$(( [ -f ../results.out.${EVAL_NUM} ] && echo 0 ) || (echo 1 && sleep 10) )
done
sleep 5
Y=1
while [ "$Y" -ne 0 ]
do
Y=`grep -e "Evaluation ${EVAL_NUM} has completed" ../HeatCalibration.out | echo $?`
sleep 5
done

cd ..
rm -f MaxHeatLoad.stdout
JOBID=$(head -n 1 ../JOBID.${EVAL_NUM} | cut -d '.' -f1)
mv ../JOBID.${EVAL_NUM} ../JOBID.LIGHTNING/.
rm -rf ../JOBID.LIGHTNING
rm -rf workdir.${EVAL_NUM}
```

## CFD PBS Evaluation Script: Optimization Study

An example of a PBS script running on the DSRC for optimization studies. This script is used to obtain a node for CFD++ to run a simulation for a single function evaluation and post-process the results for Dakota. This script also checks for convergence. If convergence has not been met, the CFD++ simulation is restarted to perform additional iterations. If convergence is not met, a set of results are still sent back to Dakota. However, this function evaluation is noted as not reaching convergence.

```
#!/bin/bash
#PBS -l walltime=1:00:00
#PBS -l select=1:ncpus=24:mpiprocs=24
#PBS -A AFPRW00452P05
#PBS -o CFD.o
#PBS -e CFD.e
#PBS -q background

cd $PBS_O_WORKDIR
module load cfd++/15.1.1
MPI_DISPLAY_SETTINGS=disabled
CONVERGE=0
LOOP=1
APPLIC_PROCS=`grep 'mpiprocs' CFD_CONCURRENT.pbs | cut -d '=' -f 4`

# Run MPI CFD++
until [ $CONVERGE -eq 1 ]; do
if [ $LOOP -lt 5 ]; then
aprun -n $APPLIC_PROCS mpimcfd > MCFD.outjob

# -----
# POST-PROCESSING
# -----

until [ -f complete ]
do
sleep 10
done

rm complete
CONVERGE=$(grep -c 'orders of magnitude of RHS convergence achieved' MCFD.outjob)
LOOP=$(( LOOP + 1 ))
else
CONVERGE=1
fi
done

infoutlf 1
sleep 2
# extract thrust from the simulation output
sed -n '$p' minfol_e1 | cut -d ' ' -f4- | cut -c 2- > results.tmp.$EVAL_NUM
# extract upstream wall pressure from the simulation output
P=`cut -d ' ' -f 3 mcfcd.info6.nod138 | awk {' printf "%2.f",$1 '} | cut -d ' ' -f 2 | cut -c 2-`
# extract upstream velocity from the simulation output
U=`sed -n '$p' mcfcd.info37_u-velocity_pl1.mpf1d | cut -d ' ' -f 17,18,19 | awk {' printf "%12.f",$1 '}`
# extract upstream temperature from the simulation output
T=`sed -n '$p' mcfcd.info37_temperature_pl1.mpf1d | cut -d ' ' -f 17,18,19 | awk {' printf "%12.f",$1 '}`
# check pressure
if [ -z $P ]; then
mv results.out.bad ../results.out.$EVAL_NUM
echo "${EVAL_NUM}--- Error with pressure" >> ../ResultsError.txt
else
if [ $P -le 6000 ]; then
mv results.out.bad ../results.out.$EVAL_NUM
echo "${EVAL_NUM}--- Pressure < 6000 Pa" >> ../ResultsError.txt
else
```

```

        echo $P >> results.tmp.$EVAL_NUM
    fi
fi

# check temperature and calculate Mach number
if [ -z $T ]; then
    mv results.out.bad ../results.out.$EVAL_NUM
    echo "${EVAL_NUM}--- Error with temperature" >> ../ResultsError.txt
else
    if [ -z $U ]; then
        mv results.out.bad ../results.out.$EVAL_NUM
        echo "${EVAL_NUM}--- Error with velocity" >> ../ResultsError.txt
    else
        if [ $T -le 100 ]; then
            mv results.out.bad ../results.out.$EVAL_NUM
            echo "${EVAL_NUM}--- Temperature < 100 deg C" >> ../ResultsError.txt
        else
            M=`echo $U $T | awk {' printf $1 / sqrt( 1.4 * 287 * $2 ) '}`
            echo $M >> results.tmp.$EVAL_NUM
        fi
    fi
fi

if [ $LOOP -lt 5 ]; then
    mv results.tmp.$EVAL_NUM ../results.out.$EVAL_NUM
else
    mv results.tmp.$EVAL_NUM ../results.out.$EVAL_NUM
    echo "${EVAL_NUM}--- Solution did not converge" >> ../ResultsError.txt
fi

sleep 10
echo $PBS_JOBID > ../../JOBID.$EVAL_NUM
exit

```

## CFD PBS Evaluation Script: Calibration Study

An example of a PBS script running on the DSRC for calibration studies. This script is used to obtain a node for CFD++ to run a simulation for a single function evaluation and post-process the results for Dakota. This script also checks for convergence. If convergence has not been met, the CFD++ simulation is restarted to perform additional iterations. If convergence is not met, a set of results are still sent back to Dakota. However, this function evaluation is noted as not reaching convergence.

```
#!/bin/bash
#PBS -l walltime=2:00:00
#PBS -l select=1:ncpus=24:mpiprocs=24
#PBS -A AFPRW00452P05
#PBS -o CFD.o
#PBS -e CFD.e
#PBS -q standard

cd $PBS_O_WORKDIR
module load cfd++/15.1.1
MPI_DISPLAY_SETTINGS=disabled
CONVERGE=0
LOOP=1
APPLIC_PROCS=`grep 'mpiprocs' CFD_CONCURRENT.pbs | cut -d '=' -f 4`

# Run MPI CFD++
until [ $CONVERGE -eq 1 ]; do
if [ $LOOP -lt 5 ]; then

aprun -n $APPLIC_PROCS mpimcfd > MCFD.outjob

# -----
# POST-PROCESSING
# -----

until [ -f complete ]
do
    sleep 10
done

rm complete
CONVERGE=$(grep -c 'orders of magnitude of RHS convergence achieved' MCFD.outjob)
LOOP=$(( LOOP + 1 ))
else
    CONVERGE=1
fi
done

if [ $LOOP -lt 5 ]; then
    # extract wall pressure from the simulation output (mcfcd.info28_1.tec)
    sed -n '17~4p' mcfcd.info28_1.tec | cut -c 3-15 > pressure.sim

    for i in {1..151}
    do
        P_sim=`sed -n "${i}p" pressure.sim`
        P_exp=`sed -n "${i}p" pressure.exp.Div`
        delP=`echo $P_sim $P_exp | awk {' printf ($1-$2)^2 '}`
        echo $delP >> delPsquared.$EVAL_NUM
    done
    P_old=0
    # calculate RMS erro
    for i in {1..151}
    do
        P_new=`sed -n "${i}p" delPsquared.$EVAL_NUM`
        P_old=`echo $P_old $P_new | awk {' printf $1 + $2 '}`
    done

    RMS=`echo $P_old | awk {' printf sqrt($1/151) '}`
```



```

echo $RMS >> results.tmp.$EVAL_NUM
mv results.tmp.$EVAL_NUM ../results.out.$EVAL_NUM
else
# extract wall pressure from the simulation output (mcfid.info28_1.tec)
sed -n '17~4p' mcfid.info28_1.tec | cut -c 3-15 > pressure.sim
for i in {1..151}
do
P_sim=`sed -n "${i}p" pressure.sim`
P_exp=`sed -n "${i}p" pressure.exp.Div`
delP=`echo $P_sim $P_exp | awk {' printf sqrt(($1-$2)^2) '}`
echo $delP >> delPsquared.$EVAL_NUM
done
P_old=0
for i in {1..151}
do
P_new=`sed -n "${i}p" delPsquared.$EVAL_NUM`
P_old=`echo $P_old $P_new | awk {' printf $1 + $2 '}`
done

RMS=`echo $P_old | awk {' printf sqrt($1/151) '}`
echo $RMS >> results.tmp.$EVAL_NUM
mv results.tmp.$EVAL_NUM ../results.out.$EVAL_NUM
echo "${EVAL_NUM}--- Solution did not converge" >> ../ResultsError.txt
fi

sleep 10
rm -f delPsquared.$EVAL_NUM

echo $PBS_JOBID > ../../JOBID.$EVAL_NUM
exit

```

## Dakota Input File

An example of a Dakota input file. This file is used to run Dakota using the MOAT method to optimize the three heat sources to maximize the thrust while keeping the shock train at a reasonable location in the isolator.

```
# Dakota Input File:MaxHeatLoad.in (Batch PBS)
# Usage:
#       dakota -i MaxHeatLoad.in -e MaxHeatLoad.e -o MaxHeatLoad.out > MaxHeatLoad.stdout

environment
tabular_data
  tabular_data_file = 'MaxHeatLoad.dat'

method
psuade_moat
  partitions = 9
  samples = 1000

variables
  continuous_design = 3
  lower_bounds = 0      0      0
  upper_bounds = 0.45   0.7    1
  descriptor = 'heat1'  'heat2' 'heat3'

interface
  analysis_driver = 'PBS_Eval'
  fork
    asynchronous
    evaluation_concurrency = 10
  parameters_file = 'params.in'
  results_file    = 'results.out'
  file_tag

responses
  objective_functions = 2
  sense = 'max'      'min'
  descriptors 'Thrust'  'Mach'
```

## CFD++ Input File

An example of a CFD++ input file. This file is used for the inputs of the Fully Convergent geometry with three heat sources.

```
#-----
# Input file created by CFD++ 15.1.1 GUI
#-----
system begin
mc_filecopy cdepsout.bin cdepsin.bin
mc_filecopy cdaveout.bin cdavein.bin
mc_filecopy mcfd6dof.out mcfd6dof.inp
system end
#-----
iofiles begin
lm_type STANDARD
lm_cells_limit 0
lm_cpus_limit 0
lm_ser2par 0
ifinbc 0
invoke_script 0
infout28_nregrd 1
#-----
mcpusin_fn mcpusin.bin
nodesin_fn nodesin.bin
cellsin_fn cellsin.bin
exbcsin_fn exbcsin.bin
inbcsin_fn inbcsin.bin
ovsetin_fn ovsetin.bin
cdepsin_fn cdepsin.bin
pltosin_fn pltosin.bin
eqsetin_fn eqsetin.bin
zobcsin_fn zobcsin.bin
ovsetin_fn ovsetin.bin
blankin_fn blankin.bin
cgrpsin_fn cgrpsin.bin
#-----
mcpusout_fn mcpusout.bin
nodesout_fn nodesout.bin
cellsout_fn cellsout.bin
exbcsout_fn exbcsout.bin
inbcsout_fn inbcsout.bin
ovsetout_fn ovsetout.bin
cdepsout_fn cdepsout.bin
pltosout_fn pltosout.bin
eqsetout_fn eqsetout.bin
zobcsout_fn zobcsout.bin
ovsetout_fn ovsetout.bin
blankout_fn blankout.bin
cgrpsout_fn cgrpsout.bin
#---Multi-CPU IO Specified Buffer Size Controls---
#-----
iofiles end
#-----
celltypes 1
celltype 3 fluxmasks 0 0 0 0
celltype 3 vtfpmasks 0 0 0 0
celltype 3 vtxpmasks 0 0 0 0
#-----
tsteps begin
istart 1
ntstep 10000
ntstop 0
ntsmin 500
ntrmin 500
dtsmoo 0
dtsmoo_iters 4
dtsmoo_param 0.6666667
runmod 1
dtauin -1.000000e+00
```

```

dtlomx 1.000000
cflbot 1.000000e-04
cfller 0.950000
rstcfl 1
ntbclr 1
nteclr 100
cflibg 1.000000e+00
cflilen 4.000000e+01
cflglo 1.000000e+15
ilcfla 0
igcfla 0
itsync 0
ntbmdr 100
ntemdr 200
mdisbg 0.000000e+00
mdisen 0.000000e+00
ntbbfr 201
ntebfr 300
blfnbg 0.000000e+00
blfnen 1.000000e+00
cdepsave_compute 0
cdepsave_restart 1
cdepsave_ntsave 0
tsteps end
#-----
options begin
ntdsko 100
ntdskk 0
ntplto 200
ntplts 0
ntpltt 0
dtpplt 0.000000e+00
ntdskc 0
ntdskr 0
ntacou 0
ntout1 1
ntout2 0
ntout3 0
ntout4 0
ntout5 0
ntout6 10000
ntout7 0
ntout9 0
ntout10 0
ntout11 0
ntout12 0
ntout13 0
ntout14 0
ntout15 0
ntout16 0
ntout17 0
ntout18 0
ntout19 0
ntout20 0
ntout21 0
ntout22 0
ntout23 0
ntout24 0
ntout25 0
ntout26 0
ntout27 0
ntout28 0
ntout29 0
ntout37 10000
nt6dof1 1
ntoutfv 0
ntsuffv 0
ntoutes 0
ntsufes 0
ntouttp 0
ntsuftp 0

```

```

ntoptrb 0
ntsptrb 0
bcsptrb 0
mpidri 0
mcpuse 0
mcgrps 7
#-----
nodebg 0
inbczb 0
autozb 0
osetyp 0
osetdb 0
osetbr 0
osetbc 0
osetbz 0
osetll 0
osetzo 0
osetsl 0
osetir 0
osnocn 1
osnosg 1
blnkdb 0
zobcdb 0
zobcty 2
iregrd 0
iregrq 1
irezon 0
reblnk 0
igrvs1 0
igrvs7 0
igrvs9 0
igrvs10 0
igrvs13 0
igrvs14 0
#-----
viscos 1
vibnew 1
ifmdis 1
mdislh 0.250000
disson 0.100000
ifmdps 1
mdisps 0.050000
mdtype 1
mdpscf 0.300000
mdpsmx 1.000000
ifmdcb 0
mdcbfr 0.250000
#-----
xyzpol xy
ispcac 2
iblend 1
iblenz 0
blenzf 0.500000
bextac 1
bzonac 2
bzoncd 0
arcozb 0
celpol 1
cenpol 1
nodnei 1
viscrs 0
vcropt 1
tvspol 1
celpoj 0
limfac 0
rhsopt 1
rhsink 0
tolpol 1.000000e-13
limtvd 3
tvdcmp 2.0
tvdphi 0.333330

```

```

chkvol 1
tolvol 1.000000e-20
grqua1 0
grqua2 0
grqua3 1
grqua6 0
gq1dis 0.100000
gq2ang 2.000000
gq3dis 0.100000
gq6dis 0.100000
gq6ang 40.000000
#-----
rtcord 5.000000
convo1 0
convo2 0
simstr 1
dultim 0
glitrc 0.100000
gloits 1
pc_method 0
mbltim 0
mblfac 1.000000e-01
mblglt 1
geomopt1 0
method 2
itimac 2
ifunlx 1
undrlx 0.500000
implic 1
impits 16
mg_mpio 1
fg_mpfb 0
mg_mpfb 0
mg_meth 2
mg_type 2
mg_vers 1
mg_aggl 1
mg_step 5
mg_alow 1
mg_stag 1
mg_lint 2
mg_itns 1
mg_levs 20
mg_mxcg 1048576
mg_cycl 4
mg_resc 5.000000e-01
mg_terc 1.000000e-01
mg_floc 1.000000e-10
fg_resc 5.000000e-01
fg_terc 1.000000e-01
fg_floc 1.000000e-10
mg_cvis 1.0
celord 1
subdomain_mode 0
options end
#-----
infsets 14
#-----
seq.# 1 #vals 31 title eqnset_define
values 101 1 1 2 2
values 0 0 1 1 1
values 0 7 5 2 0
values 0 0 0 0 0
values 0 5 5 1 1
values 3 0 0 0 0
values 0
#-----
seq.# 2 #vals 7 title primitive_variables_2
values 65621.4 504.992 810.67 0.0 0.0
values 8.872009e+02 4.761458e+05
#-----

```

```

seq.# 3 #vals 3 title resistive_layer_heat_transfer
values 4.7510 0.003048 300.0
#-----
seq.# 4 #vals 3 title resistive_layer_heat_transfer
values 4.1004 0.003048 300.0
#-----
seq.# 5 #strs 6 title file_based_transition_tripping
strings filename.dat 0.01 1 1.0 0.0 0.0
#-----
seq.# 6 #vals 1 title infout12_inf
values 5
#-----
seq.# 7 #strs 1 title 3d_datafile(primvar2)
strings InletProfile
#-----
seq.# 8 #vals 24 title volumetric_source_by_boxes
values 1 1.107 1.398 0.0 0.088 0.0 0.0 9
values 1 1.398 1.689 0.0 0.088 0.0 0.0 10
values 1 1.689 1.98 0.0 0.088 0.0 0.0 11
#-----
seq.# 9 #strs 2 title 1d_datafile(volsource)
strings f(t) heat.box1
#-----
seq.# 10 #strs 2 title 1d_datafile(volsource)
strings f(t) heat.box2
#-----
seq.# 11 #strs 2 title 1d_datafile(volsource)
strings f(t) heat.box3
#-----
seq.# 12 #vals 15 title infout37_inf
values 1 3.025e-01 0.000000e+00 0.06808993 -0.4250828
values 0.4250828 0.0 0.0 0.0 0.0
values 0.0 0.0 0.0 0.0 0.0
#-----
seq.# 13 #vals 1 title infout6_inf
values 138
#-----
seq.# 14 #vals 3 title resistive_layer_heat_transfer
values 16.2648 0.002540 300
#-----
eqslct 101
icslct 2
#
mbcons 8
seq# type modi info
  1 186 0 3 C1
  2 186 0 3 C2
  3 186 0 4 C2C3
  4 186 0 4 C5
  5 142 0 7 Inlet
  6 186 0 14 Isolator
  7 174 0 0 Outlet
  8 6 0 0 Symm
#-----
#seq mtyp info
#-----
octree begin
toldup 1.000000e-06
tolins 1.000000e-06
toldfn 1.000000e-06
tolzco 1.000000e-06
typzco 0
trxdir 1
trydir 1
trzdir 0
dfcmax 1.000000e+00
octree end
#-----
physics begin
meq_eqsets 0
meq_eqsgrp 0

```

```

meq_icslct 0
meq_inityp 0
cht_matprp 0
absour 0
absour_selftune 1
cldriver 0
rotor_model1 0
anchor_pressure 0
moddif 0
moddif_type 2
icsrot 0
prerot 0
preacc_opt 0
preacc_dti 0.05
pretyp 1
ipreof 20
prebet 5.000000e-02
previs 5.000000e-02
prevel 1.000000e-06
prevlo 1.000000e-03
pfloor 0.000000e+00
gasnam Air
gasgam 1.400000
gasmwt 28.950000
advcon 1.000000e+00
difcon 1.000000e+00
ed_con 1.000000e+00
ed_cap 0.000000e+00
iunits 0
lenuni m
masuni kg
temuni K
timuni s
grduni m
inityp 0
qcvrti 0
qcvrto 0
ifrpow 0
ifreac 0
ifrebx 0
frctin 2
temulx 2.000000e-01
frcint 1
frclim 0
frclif 0.050000
frccfl 1
ifscon 1
iftrds 0
ifrsrc 0
ifvols 8
ifvelp 0
ifрмаi 0
frcxmn 0.000000e+00
frcmx 0.000000e+00
frcymn 0.000000e+00
frcymx 0.000000e+00
frczmn 0.000000e+00
frczmx 0.000000e+00
frsrcu 0.000000
frsrcl 1.000000
frsrct 1.000000
frsrcm 1.000000
istiff 0
sureac 0
surspe 0
surmap 0
toltem 1.000000e-05
tolfr 1.000000e-05
tnoneq_numeqns 0
edp_yesno 0
edp_model 0

```



```

edp_coefvm 0.500000
edp_coefcl 0.500000
edp_ifbuoy 0
edp_gravtx 0.000000
edp_gravty 0.000000
edp_gravtz 0.000000
sourc1 0
sourc2 0
ifaxix 1
ifaxiy 0
ifwzro 1
ifaxst 0
ifaxsw 0
ifswrl 0
swrxmn 0.000000e+00
swrxmx 0.000000e+00
swrymn 0.000000e+00
swrymx 0.000000e+00
swrzmn 0.000000e+00
swrzmx 0.000000e+00
swreta 0.500000
ifblk 0
blkxmn 0.000000e+00
blkxmx 0.000000e+00
blkymn 0.000000e+00
blkymx 0.000000e+00
blkzmn 0.000000e+00
blkzmx 0.000000e+00
ipbulk 0
bulkpr 0.000000
irbulk 0
bulkro 0.000000e+00
bulktm 2.880000e+02
presab 0.000000e+00
presmn -1.000000e+20
presmx 1.000000e+20
tempmn -1.000000e+00
tempmx 1.000000e+20
univgc 8314.000000
refmwt 1.000000
reflen 1.000000
reftem 1.000000
refden 1.000000
refvel -1.000000
refpre 1.000000
refpgf 101325.000000
ifbuoy 0
gravtx 0.000000
gravty 0.000000
gravtz 0.000000
grvcnx 0.000000e+00
grvcny 0.000000e+00
grvcnz 0.000000e+00
grvbet 0.003000
liqgas 0
vislaw 0
refmuu 1.000000e+00
refkap 1.000000e+00
prndtl 0.72
prlatu 0.800000
schmla 0.700000
schmtu 0.700000
rmuusl 1.716000e-05
tmuusl 273.110000
smuusl 111.000000
rkapsl 2.410000e-02
tkapsl 273.110000
skapsl 194.000000
liqtlo 0.0
liqtup 5000.0
yppfac 1.000000

```

```

ifporo 0
ifpmut 0
isporo 0
ifcjht 0
ifnlas 0
ls_numeqns 0
ntrbst 11
iftold 0
smagcf 0.050000
lnstyp 3
lnsdtn 1
lnsbox 0
mnfltr 0.000000e+00
sync_alpha 1
nlas_all scales 0
rfg_sample_modes 100
mulnyq 4
ininls 0
rfg_rseed 1234567
ifvspt 0
ifmuon 0
ifdpds 0
ifwfne 2
ifwfol 0
ifwfbc 3
ifcomp 1
ifmccw 1
ifskar 0
ifpope 0
ifbrad 0
iftrat 0
iftrfs 0
ifmbsl 0
iftcon 1
iftrbf 0
turbf1 1.000000
turbf2 1.000000
turbf3 1.000000
turbf4 1.000000
turbf5 1.000000
turbf6 1.000000
turbf7 1.000000
trurlx 1.000000
kmxval 1.000000e+20
tmnval 1.000000e-12
maxmut 1.000000e+10
turlim 100.000000
turxyz 0
cgtsf 0
turxmn 0.000000e+00
turxmx 0.000000e+00
turymn 0.000000e+00
turymx 0.000000e+00
turzmn 0.000000e+00
turzmx 0.000000e+00
shaper 0
physics end
#-----
probe begin
infout6_inf 13
infout37_inf 12
probe end
#-----
debug begin
debug end
#-----
guiopts begin
turbi_lev 1
turbi_len 1
turbi_mutmu 5.000000e+01
turbi_jet 0

```

```

turbi_int 1
turbi_tlev 3.000000e-02
turbi_tlen 5.550000e-02
turbi_tfrs 1.000000e-04
turbi_yp 1.000000e+00
turbi_replen 1.000000e+00
auto_pres 1.013250e+05
auto_temp 2.880000e+02
auto_u 3.000000e+01
auto_v 0.000000e+00
auto_w 0.000000e+00
auto_tlvl 2.000000e-02
auto_tlnl 1.000000e-02
auto_tl 1
auto_lt 0
auto_eqinf 0
auto_ininf 0
auto_bpinf 0
auto_mutmu 10.000000
aero_intyp 1
aero_unit 1
aero_pres 1.013250e+05
aero_temp 2.880000e+02
aero_deltat 0.000000e+00
aero_u 3.000000e+01
aero_v 0.000000e+00
aero_w 0.000000e+00
aero_tlvl 2.000000e-03
aero_tlnl 1.000000e-01
aero_tl 1
aero_lt 0
aero_eqinf 0
aero_ininf 0
aero_bcinf 0
aero_altid 1.000000e+01
aero_ma 8.000000e-01
aero_alpha 0.000000e+00
aero_beta 0.000000e+00
aero_plane 0
aero_re 1.000000e+06
aero_mutmu 10.000000
incomp_tlvl 5.000000e-02
incomp_tlnl 1.000000e-02
incomp_eqinf 0
incomp_ininf 0
incomp_bpinf 0
guiopts end
#-----
system begin
echo > complete
system end
#-----

```

## APPENDIX B: ADDITIONAL PLOTS

### MOAT Optimization Study

This is an additional plot that shows the solutions from the reduced design space MOAT optimization study. It can be seen that the optimal solutions fall around 6 MW of total heat addition. There is a region of overlap for the green and blue points. This overlap indicates similar amounts of total heat addition, but with a different distribution. This is the ideal region to target for optimal solutions. This region also illustrates the sensitivity of the shock train in the isolator.

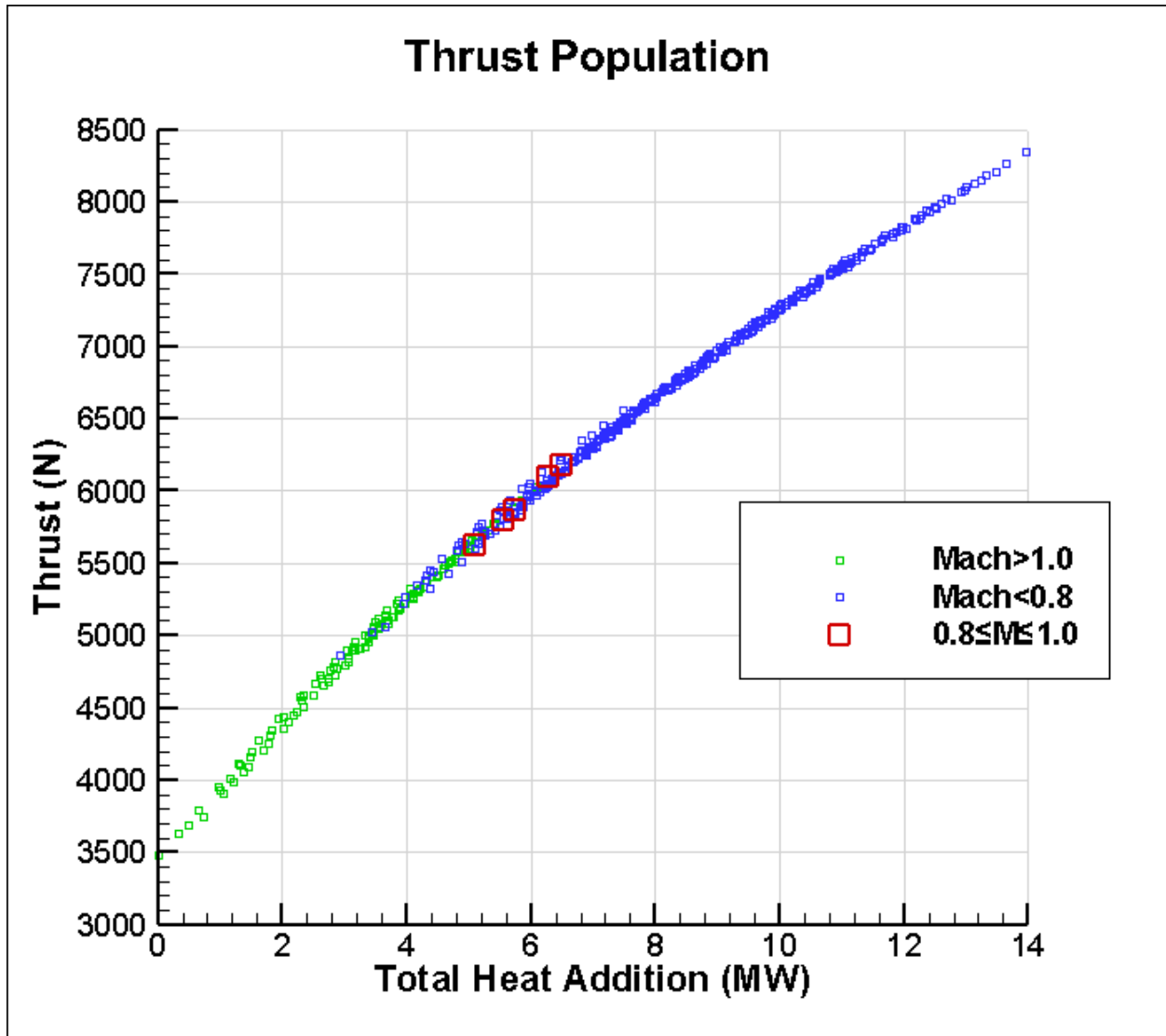


Figure 31: Thrust Curve for Reduced Design Space MOAT Optimization Study

## Convergence Check

To confirm that the solutions from the optimization studies have fully converged, a simulation was restarted to run for additional iterations. The plot shows the residuals for the energy and the mass of the system. The three steps illustrate the three separate evaluations ran. At iteration zero, the simulation was restarted from the grid sensitivity solution and ran for 40,000 iterations. The convergence criteria of  $10^{-5}$  for the energy residual was not met. Therefore the simulation was restarted and ran for another 40,000 iterations in which the residual of the energy fell below  $10^{-5}$ . The final step shows another restart for another 50,000 iterations to show that the residual of both energy and mass did not decrease.

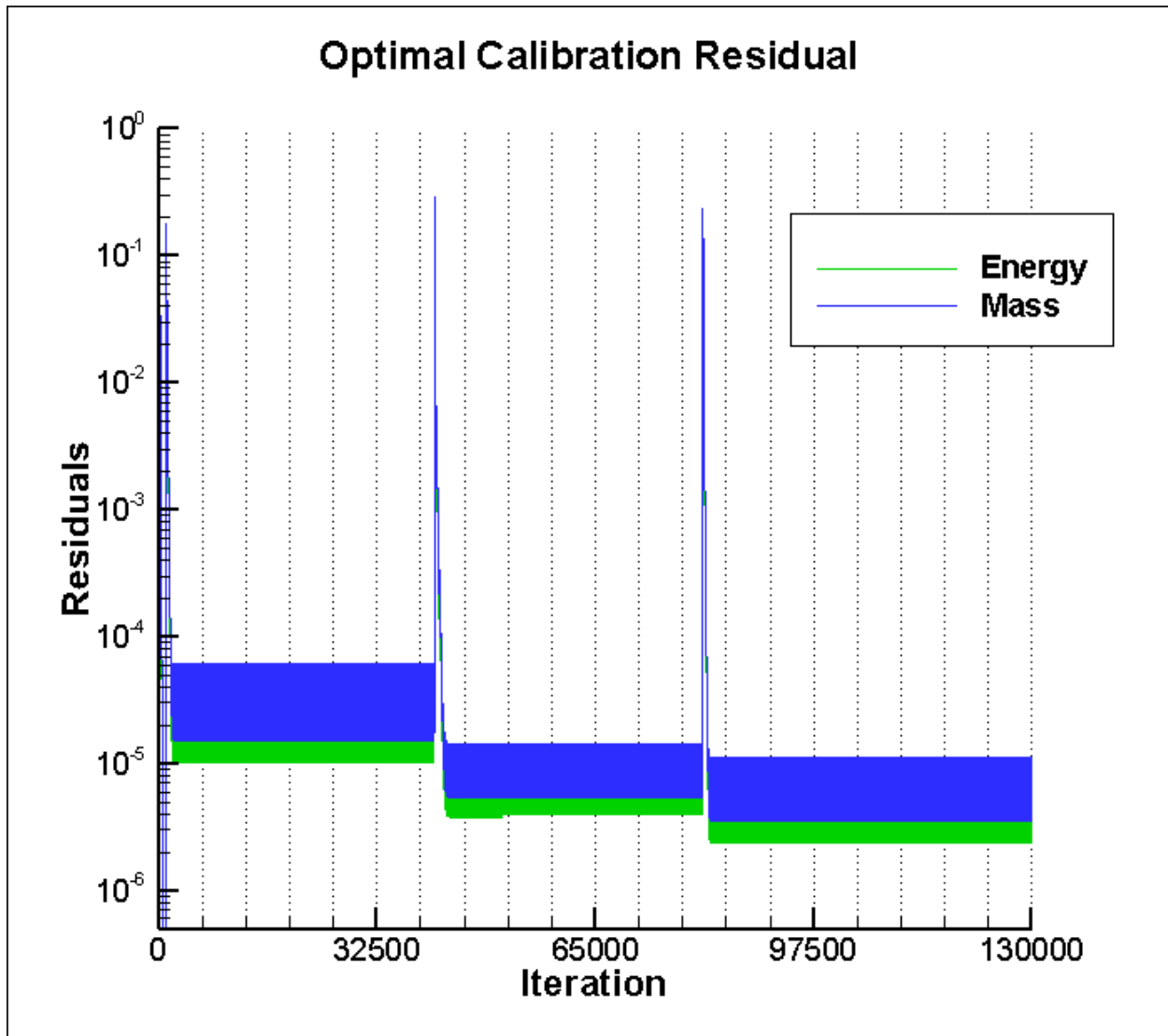


Figure 32: Additional Convergence Check