DESIGN OPTIMIZATION METHODS FOR ADDITIVELY MANUFACTURED NATURAL CONVECTION HEATSINKS

By

DANIEL CARNE

Bachelor of Science in Mechanical Engineering Technology

Oklahoma State University

Stillwater, Oklahoma

2019

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2021

DESIGN OPTIMIZATION METHODS FOR

ADDITIVELY MANUFACTURED NATURAL

CONVECTION HEATSINKS

Thesis Approved:

Aaron Alexander

Thesis Adviser

Omer San

Craig Bradshaw

ACKNOWLEDGEMENTS

I would like to thank my advisors and committee members Dr. Aaron Alexander, Dr. Ilchung Park, Dr. Omer San, and Dr. Craig Bradshaw for all their time, guidance, and encouragement along the way. This would not have been possible without them.

I am deeply grateful for Professor Laura Emerson for helping me and guiding me into the person I am today. I am also deeply grateful for my family and dog for their encouragement and support.

And finally, and big thank you to everyone on the heatsink team here at OSU for their support including Dr Hitesh Vora, Turner McCoy, and Stephen Higginbothom.

Name: DANIEL CARNE

Date of Degree: DECEMBER, 2021

Title of Study: DESIGN OPTIMIZATION METHODS FOR ADDITIVELY MANUFACTURED NATURAL CONVECTION HEATSINKS

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: With 3-D printing technology becoming more widely available, designs with complex geometries are now feasible to manufacture. The ability to create complex geometries opens many new possibilities in the optimization field. Without these manufacturing constraints, unique heatsink designs can be made. Efficient methods of creating heatsinks Designed for Additive Manufacturing (DfAM) are still lacking and not readily available to industry. This research investigates two different optimization methods for heatsinks. The first method investigates applying an evolutionary algorithm to optimize vertically mounted heatsinks. A custom code was used to simulate both the conduction within the heatsink and the temperature of the fluid field around the heatsink. The optimization of the heatsink design is achieved by growing from the base and slowly adding material. In order to choose where to add material, the code trains an evolutionary algorithm based on the geometry and the temperature distribution in the metal and fluid. The second optimization method investigates the use of an Artificial Neural Network (ANN) to assist in the optimization of natural convection heatsinks. A custom Computational Fluid Dynamics (CFD) code was used to simulate the fluid and temperature fields inside and around the heatsink. The optimization of the heatsink design is achieved by choosing a sensitivity equation to decide where to add material based on factors such as velocity and convection rate. A neural network then learns the system and optimizes the variables of the sensitivity equation to increase the heat transfer of the heatsink. These optimization methods can be constrained by several options such as heat transfer characteristics, weight, and maximum envelope size. Experimental analysis is performed to show the validation of the second optimization method and benefits of DfAM.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 Problem Statement

The goal of a heatsink is to enhance heat transfer for electronic devices. In order to enhance heat transfer, heatsinks are often heavy and take up a significant amount of space. This requires extra volume and weight in electronic devices. With the miniaturization of electronics, thermal power output per unit area increases requiring improved cooling. To meet these needs, heatsink designs with better cooling to volume ratios are necessary to keep from having bulkier and heavier electronics. Optimized geometry heatsinks can provide these designs helping to minimize weight for aircraft, and reduce volume necessary for phones, laptops, and many other electronic devices.

There are many different optimization methods that have been applied to heatsinks such as parametric optimization and topology optimization. Each of these methods have significant advantages and disadvantages. Parametric optimization can be quick and require little computational power but also provides a very simple solution. Topology optimization creates a unique design that is not constrained to any specific geometry but can require significant computational resources. New optimization methods can help us to better understand optimal heatsink designs as well as to help mitigate some of the disadvantages of current methods.

1.2 Objectives

The objective of this research is to investigate the application of new optimization methods to heatsink design. By studying different optimization methods, whether they increase heat transfer or not, will lead to a better understanding of heatsink design and what the ideal optimization method for each situation is.

Objective 1: The goal of the first study is to utilize an Evolutionary Algorithm (EA) to optimize the design of a heatsink in a two-dimensional simplified model. This will show whether it is possible for an evolutionary design method to optimize heat transfer within a simulation, and the benefits of applying this method to two-dimensional heatsink designs.

Objective 2: The goal of the second study is to apply an Artificial Neural Network (ANN) to optimize a sensitivity equation. The sensitivity equation, which chooses where material will be added, will have coefficients which the ANN can alter to optimize the heatsink design. This will be integrated into a three-dimensional Computational Fluid Dynamics (CFD) simulation. These heatsinks will then be Additively Manufactured (AM) and experimentally validated. From this, we will compare the benefits of this optimization method with a parallel fin heatsink.

CHAPTER II

REVIEW OF LITERATURE

2.1 Evolutionary Algorithms

Applying a gradient descent method to optimize complex problems or functions with multiple extrema, or multiple optimization objectives, does not often provide an ideal solution. With a gradient descent method, a local maxima or minima can be reached without testing if other possibly better solutions exist. An Evolutionary Algorithm (EA) is one of the early optimization methods used to overcome this issue. This method has many different types and can mean several things, but generally an EA is a type of algorithm that has a set population with varying traits. A fitness function is used to calculate the fitness of each individual based off these traits. The top individuals with the highest fitness ratings are selected to pass down their traits to the next generation. The traits are mutated so the new generation contains unique individuals, but still contain traits similar to the top individuals from the previous generation. This process is repeated until a set goal has been reached or the fitness of the top individuals from each generation are no longer increasing [1, 2].

Evolutionary algorithms have been applied to heatsink design optimization by choosing set dimensions to vary. Past studies have used an EA to vary dimensions such as fin width or channel height to reduce the thermal resistance of a heatsink while maintaining other constraints such as pressure drop [3-6].

2.2 Neural Networks

An Artificial Neural Network (ANN) is a form of Artificial Intelligence (AI) that assists in modeling and learning systems. An ANN consists of an input layer, an output layer, often including hidden layers in between, each consisting of a set number of nodes. Each node is connected to the nodes in the next layer through weights which are trained through gradient descent to minimize the error between the data provided for training and the final output of the ANN. This is an efficient method for learning systems and making complex predictions based on previous data. There are many different types of neural networks and a very extensive history on their application. For our purposes we will specifically be considering a Convolutional Neural Network (CNN) for its simplicity and well known capabilities [7-9].

Neural networks have been applied to many different applications within engineering. By training with experimental or simulated data, neural networks have been used to make predictions to reduce the overall time and cost of experiments or simulations, and improve our understanding of these systems [10]. This approach has been used for everything from evaluating concrete designs to modeling turbulent flow [11, 12]. Several previous studies have used ANNs to evaluate heatsink or heat exchanger performance based on the heatsink parameters or fluid properties [13-16].

2.3 Heatsink Optimization Methods

One of the most common and effective heatsink designs used is the parallel fin heatsink. This design works well with both forced and natural convection in multiple different mounting configurations. Using correlations, simulations, or experiments, the optimal number of fins can easily be determined. An early approach to optimizing heatsink geometries was to parametrically optimize by changing design features such as fin thickness and spacing [17-19]. Evolutionary and

genetic algorithms can optimize heatsinks in a similar way by varying chosen design features [20]. This type of optimization is widely used and can be applied to any design feature including lattice designs [21]. While it is an effective and popular optimization method, it is highly constrained and limiting. Optimization methods that can generate more complex geometries are needed to fully take advantage of current additive manufacturing capabilities.

Topology optimization (TO) can generate unique structures without preset designs. This was first applied to structural optimization problems [22] and has since been applied to fluid and heat transfer problems including heatsinks [23-28]. Topology optimization works by calculating a sensitivity analysis that determines where to add or remove material in a system. By not having preset design features such as a fin or pin, TO can create unique and complex structures that make the most of additive manufacturing capabilities.

2.4 Additive Manufacturing

Additive Manufacturing (AM), also known as 3-D printing, has become increasingly available to industry over the past decade. Standard heatsink manufacturing methods often limited designs to simple geometries, but with metal AM almost any design heatsink is possible to manufacture. This development has led to increased research and testing in thermal properties of AM metals and heatsinks Designed for Additive Manufacturing (DfAM). Additively manufactured metals used for heatsinks such as aluminum and copper can be porous and less dense than after other manufacturing methods which can lead to a drop in thermal conductivity [29]. Annealing the AM part can correct this issue and retain similar thermal properties to that of standard heatsink manufacturing methods and materials [30]. With excellent thermal properties and the capabilities to produce a wide range of geometries, AM has become one of the top choices for prototyping, testing, and manufacturing small orders of high performance heatsinks [31-33].

CHAPTER III

2-D EVOLUTIONARY ALGORITHM HEATSINK OPTIMIZATION

3.1 Introduction

Heatsink design optimization generally will use parametric optimization or topology optimization to increase the heat transfer properties. These methods both work well but also have their downsides. Applying new optimization methods, even if they may not be the preferred method, is important to learn how new designs can be generated, and how downsides of other optimization methods may be overcome. One optimization method not commonly applied to heatsink design is evolutionary algorithms and genetic algorithms. These algorithms mimic nature by creating a population of individuals, each with different variables. The individuals with the best solutions are chosen to pass down variables to the next population. These individual's values are also mutated in order to find better solutions over time. Evolutionary algorithms are especially preferred for systems that have multiple different local maxima where a gradient descent method might not find as favorable of a solution. Evolutionary algorithms and genetic algorithms have been applied in many different areas in engineering because of its ability to find optimal solutions in complex systems [34] and specifically in heatsink optimization [35].

Many of the generative design works function by allowing the algorithm to progressively add material to create the heatsink design. They start with only the base and adds a designated amount

of material during each iteration until the design criteria are met. Similar to the method done by Bornoff [36], this offers several benefits such as providing heatsink designs optimized for constrained spaces. However, it is difficult to determine where best to add material. One method used to decide where to add material determines the hottest area of the heatsink, adds the material there, then checks if the additional material lowered the base temperature. If it did not lower the base temperature, the material is removed, and the location marked to leave that space open. While this method does grow a heatsink, it is more of a guess and check method rather than an optimized solution. One way to reduce the arbitrariness of this method is to train an evolutionary algorithm to decide where to add material.

3.2 Simulation

To test the capabilities of this method, a simplified two-dimensional simulation is implemented. The X-Y plane is simulated considering conduction and convection using a specified heat transfer coefficient along material boundaries. Although the Z direction is not simulated, the fluid cells are cooled as if there was movement in the Z direction by a constant heat transfer to represent forced convection, or as a function of the local temperature to represent natural convection if gravity acted in the same Z direction. While this does not fully simulate the physics, it successfully facilitates the purpose of this study showing the benefits of an evolutionary algorithm for heatsink optimization.

Considering a steady state finite volume element, conservation of energy can be applied to show that

$$\sum_{i=1}^{n} \dot{Q}_i = 0 \tag{1}$$

where

$\dot{Q}$ heat transfer of a face on the volume element

n number of faces on the volume element.

Applying Fourier's law and Newton's law of cooling to calculate the steady state heat transfer at each face, this equation then becomes

$$\sum kA\frac{\Delta T}{\Delta x} + \sum hA\Delta T = 0 \tag{2}$$

where

A area of a cell face

h convection coefficient

k thermal conductivity

ΔT change in temperature

Δx change in distance.

The discretized equation rearranged to solve for the future pseudo time step temperature of a square volume element is

$$\boldsymbol{T}_{\boldsymbol{i,j}}^{\boldsymbol{n+1}} = \frac{\sum(\boldsymbol{d}_{\boldsymbol{i\pm0.5,j\pm0.5}}^{\boldsymbol{n}} * \boldsymbol{T}_{\boldsymbol{i\pm1,j\pm1}}^{\boldsymbol{n}}) + \boldsymbol{f}_{\boldsymbol{i,j}}}{\sum(\boldsymbol{d}_{\boldsymbol{i\pm0.5,j\pm0.5}}^{\boldsymbol{n}})} \tag{3}$$

where

d diffusion coefficient array

f heat transfer in the Z direction

T temperature array

n pseudo timestep

i,j X-Y location.

The diffusion coefficient array values have three different possibilities, dependent on if the interface is fluid-fluid, solid-solid, or fluid-solid. These values are

$$d_{f-f} = \frac{k_f A}{\Delta x} \tag{4}$$

$$d_{s-s} = \frac{k_s A}{\Delta x} \tag{5}$$

$$d_{f-s} = hA \tag{6}$$

where

k_f fluid thermal conductivity

k_s solid thermal conductivity.

The heat transfer in the Z-direction can be either a uniform heat transfer, or a function of the local temperature. This is only applied to fluid cells to represent cooling due to either forced convection or natural convection where

$$f_{fc} = a \tag{7}$$

$$f_{nc} = bT \tag{8}$$

where

a constant heat transfer

b constant diffusion coefficient

f_{fc} heat transfer for forced convection

f_{nc} heat transfer for natural convection.

Utilizing a 100 x 100 cell cartesian mesh gives a system of 10,000 equations and unknowns. A Gauss-Seidel iterative solver is applied to solve Eq. 3 at each cell. Successive over relaxation is combined with the Gauss-Seidel solver to accelerate the solution speed. $\boldsymbol{T^*_{i,j}}$, the cell temperature at the next iteration, is solved by

$$T^{*}_{i,j} = T^{n}_{i,j} + w(T^{n+1}_{i,j} - T^{n}_{i,j}) \tag{9}$$

where

$T^{n}_{i,j}$ previous iterations temperature array

$T^{n+1}_{i,j}$ future pseudo time step temperature array solved from Eq. 3

w successive over relaxation factor.

$T^{*}_{i,j}$ then replaces $T^{n}_{i,j}$ in Eq. 3 and this process is repeated until stopping criteria are met. The stopping criteria is based on the convergence of the simulation where

$$convergence = \sum \left| T^{n+1}_{i,j} - T^{n}_{i,j} \right|. \tag{10}$$

For this study an over relaxation factor of 0.95 is used to maximize solution speed while maintaining stability as shown in Fig. 3.1. The iterative solver considers the simulation solved when the convergence reaches below 10^{-8} as changes in the heat transfer are negligible past this point.

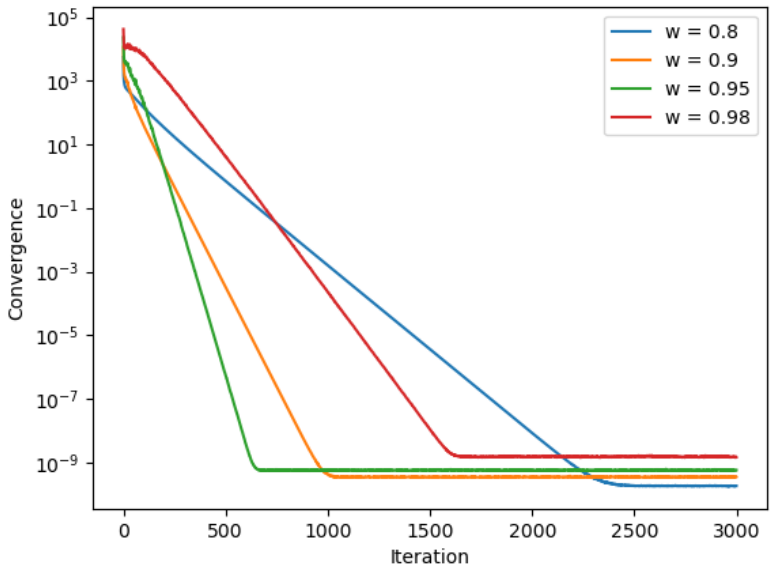

Figure 3.1: Convergence speed of varying over relaxation factors.

The boundary conditions include three walls with a constant temperature representing ambient air, T_c, one adiabatic wall with no heat transfer, and the base of the heatsink which can either be a set temperature, T_h, or a set heat transfer, $\dot{Q}$. In Fig. 3.2 the solid line represents the outer walls of the simulation and the dotted lines represent the area containing the heatsink.

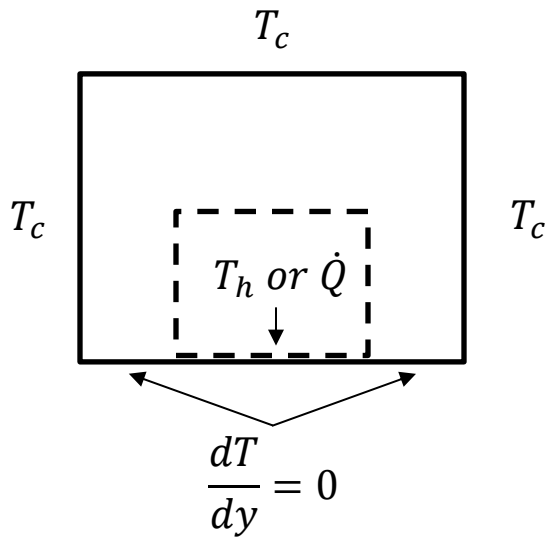

Figure 3.2: Boundary conditions of an example heatsink.

Figure 3.3 shows this simulation method applied to a forced convection parallel fin heatsink with a constant base temperature.

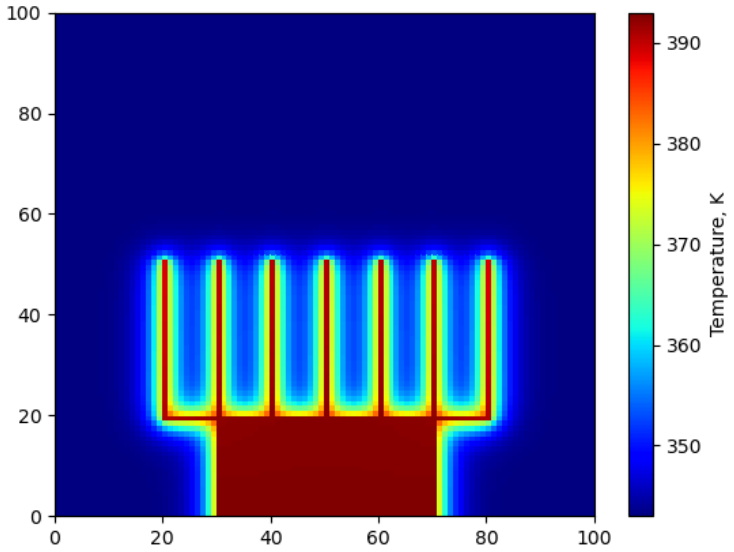

Figure 3.3: Temperature scene of example heatsink simulation.

3.3 Optimization

The method used by Bornoff [36] allows a heatsink to grow by slowly adding material until the desired output is met. This allows for “organic” heatsink designs similar to methods used in nature such as plants growing.

An evolutionary algorithm can assist in selecting where to add material during the heatsink growth process. Evolutionary algorithms are modeled after the way species adapt and change in nature [13]. As species reproduce random mutations can occur over time. After enough time mutations with better results survive and breed at a higher rate. To optimize a system with an evolutionary algorithm, variables to be optimized must be chosen. The first generation’s variables are normally randomized and only a set number of individuals are allowed to pass on their variables to the second generation based on a defined fitness function. A fitness function judges how well a set of variables performed. There are various methods for defining a fitness function

and is highly dependent on the optimization problem. The individuals with the highest fitness functions are selected to pass down their variables. These variables are then altered slightly, or mutated, and the second generation is created. This process is repeated until there is no improvement between generations.

Many different functions and sets of variables were evaluated, and the best ones found to optimize a heatsink under the conditions used are shown in Eqs. 11-12. Equation 11 calculates $f(A, \Delta T)$ based on the area that adding material at a location would add to the newly generated heatsink and the change in temperature between the metal and the fluid. This equation is solved for at the hottest location on the heatsink surface. By doing this, it forces the heatsink to grow outwards from the hotter areas to ensure evenly distributed growth. Equation 12 gives the criteria to determine whether the material should be added. If this is equation is valid then it adds the material and repeats the process. If Eq. 12 is not valid, then it does not add material at that location and moves on to check the next highest temperature cell in the heatsink. This process is repeated until it finds a location to add material. This process of creating the heatsink is described in Fig. 3.4. An evolutionary algorithm is used to optimize the x_1, x_2, and x_3 variables. By changing these variables, it can affect the heatsink growth by giving more or less influence to the area added and the temperature change.

$$f(A, \Delta T) = Ax_1 + \Delta T x_2 \tag{11}$$

$$f(A, \Delta T) > x_3 \tag{12}$$

where

A potential area added

x_i variable to optimize.

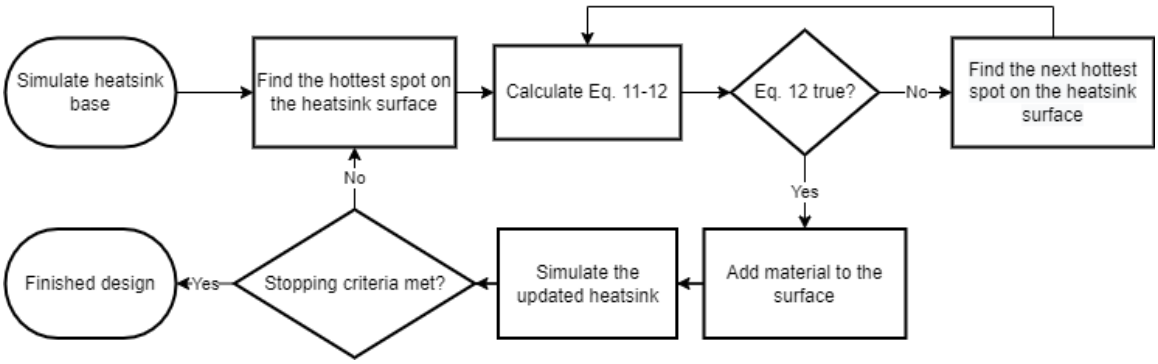

Figure 3.4: Heatsink growth flowchart.

An initial population size of sets of randomly selected variables is created. The fitness function, used to score how well each heatsink performs, is different depending on the boundary conditions. For a constant temperature at the base of the heatsink, the fitness function is the heat transfer by the heatsink. For a heat transfer input at the base of the heatsink, the fitness function is the maximum base temperature. The top three individuals from the initial population are then chosen to pass down their variables to the next generation of eight new individuals. These variables are randomly mutated by up to five percent in either direction allowing for a unique set of individuals in each generation.

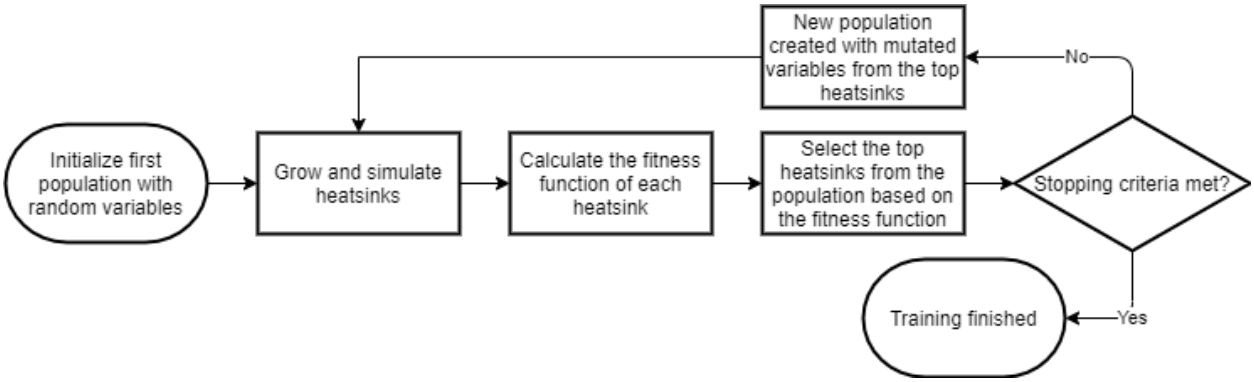

Figure 3.5: Evolutionary algorithm flowchart.

3.4 Results

To determine the performance of the heatsink designs provided by the evolutionary algorithm, results will be compared to parallel fin heatsinks. The spacing of the parallel fins will be optimized within the code by simulating different numbers of fins.

Figure 3.6 presents the results of the evolutionary algorithm. In this example, a maxima was achieved quickly because this example was optimized on a relatively course 100 x 100 cell mesh. On a more refined mesh, maxima would likely take longer to find because a small change in the variables is more likely to generate a unique heatsink design. As generations progressed, the average fitness of the 20 heatsinks improved. It is important to point out that the top individual in generation 2 and generation 3 had the same score on the fitness function, but the variables were passed down from different individuals. This shows there are two equally good solutions to this problem as shown in Table 3.1 and Fig. 3.7. For a more refined mesh, one of these would likely go on to give a slightly better solution. This highlights the importance of an evolutionary algorithm being used in this study because it was able to preserve multiple potential solutions throughout the optimization. In contrast, a gradient descent method would have only optimized one of these potential solutions.

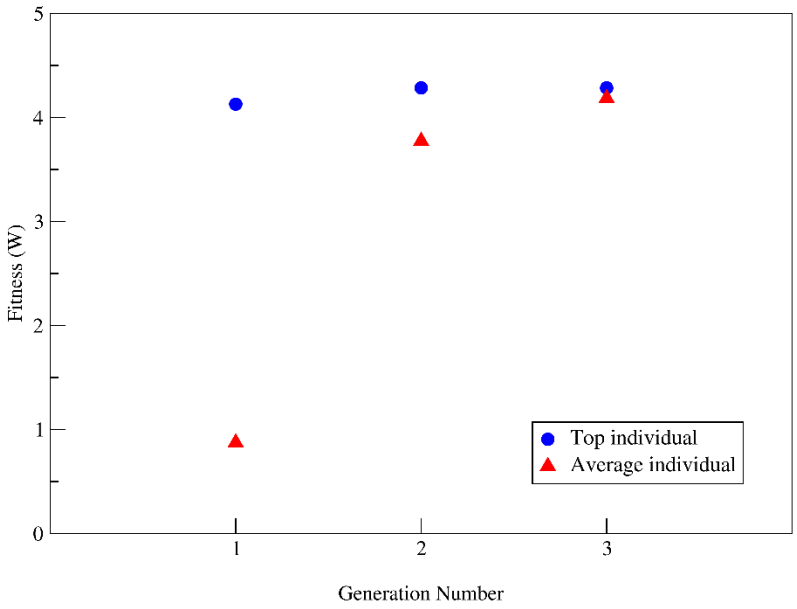

Figure 3.6: Evolutionary algorithm results.

Table 3.1: Top individuals between generations.

Generation Number	X1	X2	X3	Fitness (W)
1	374.0	169.0	5765.0	4.1
2	97.1	80.4	2398.3	4.3
3	359.8	176.0	5693.1	4.3

(a) (b) (c)

Figure 3.7: Top individual heatsink from each generation: (a) Generation 1; (b) Generation 2; (c) Generation 3.

Figure 3.8 shows a comparison between the optimized heatsink design and a parallel fin heatsink with optimized spacing. Both heatsinks are constrained to a set domain size. The optimized heatsink dissipated 2.7 Watts while the parallel fin heatsink dissipated 2.4 Watts representing a 12.5% increase in power dissipation.

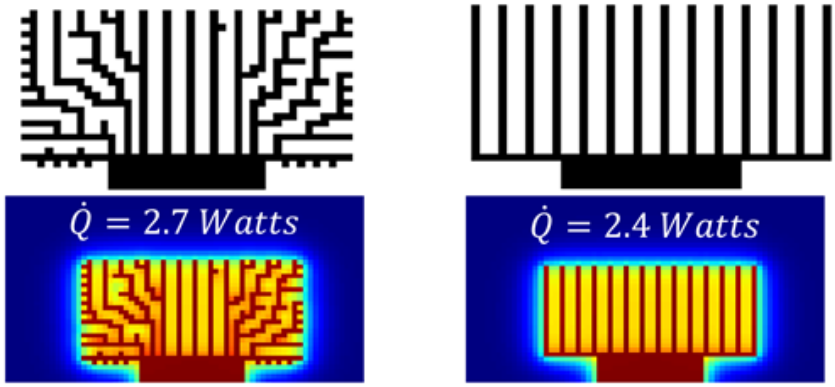

Figure 3.8: Heatsink comparison.

This growth method can also develop unique heatsink designs for geometric constraints. Figure 3.9 shows a comparison for heatsinks with a wall on one side, and a geometric constraint where another part or tool access may limit that area. The optimized heatsink dissipated 0.153 Watts while the parallel fin heatsink dissipated 0.146 Watts representing a 4.8% increase in power dissipation.

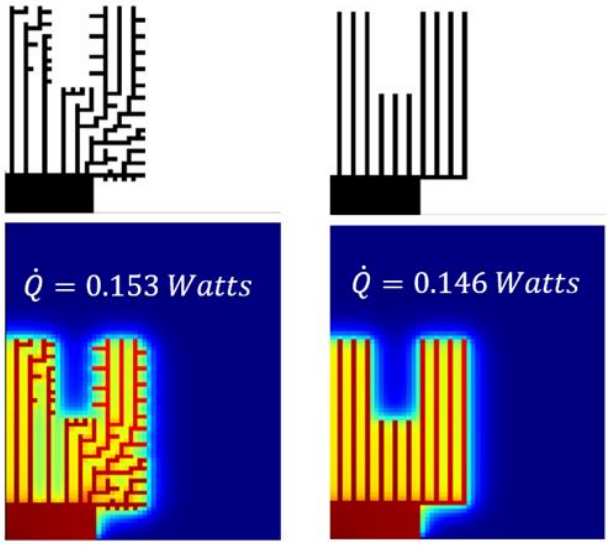

Figure 3.9: Heatsink comparison.

For the mesh size utilized in Figs. 3.8-3.9 solutions were found quickly because small changes in the optimization variables did not lead to differences in geometry. For finer meshes smaller changes in optimization variables will lead to larger changes in geometry. This means solutions will take more generations to converge for finer meshes. Figure 3.10 shows a heatsink optimized on a finer mesh than the previous heatsinks. While a similar design appears, Fig. 3.11 shows the evolutionary algorithm took five iterations instead of three to reach a maximum and displays higher volatility in the average individual. Less volatility could be achieved by limiting the variable mutation to a smaller percentage between generations however this would also slow down the generations required for a maximum to be found.

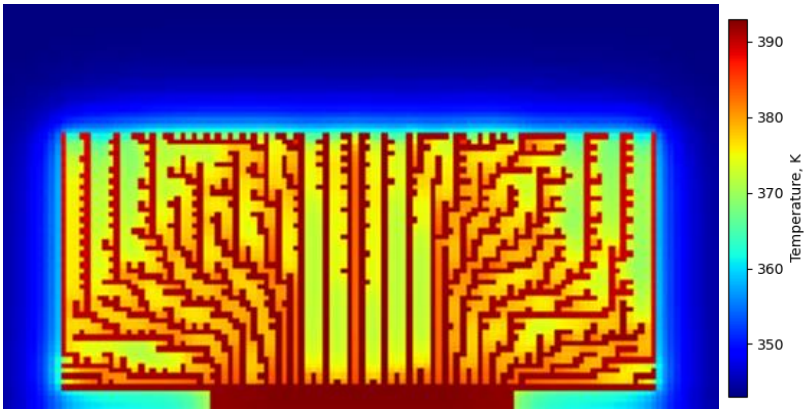

Figure 3.10: Heatsink optimized with refined mesh.

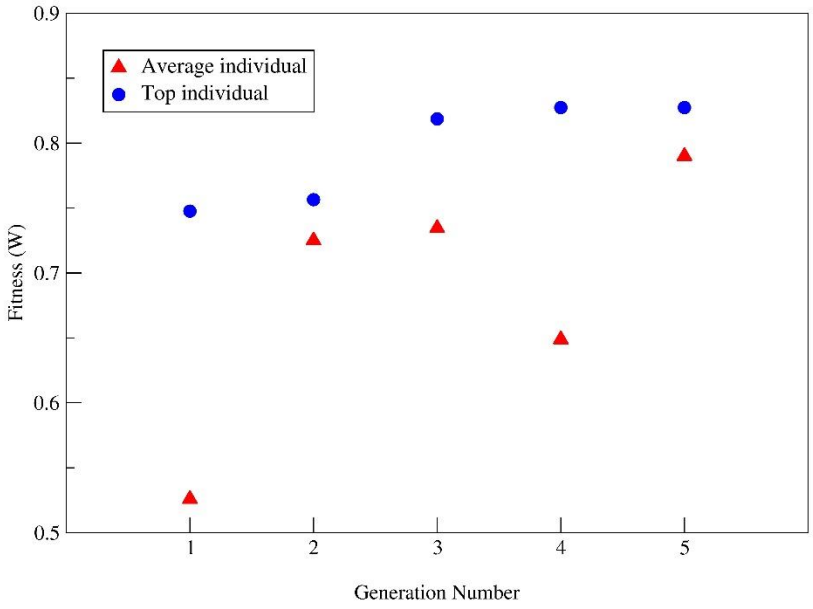

Figure 3.11: Evolutionary algorithm results.

One limitation to this method is in defining the equations in which cells are added. The current equations as defined in Eqs. 11-12 work well for optimizing the heatsinks shown where the fin thickness is not as important. In situations with higher convection or lower thermal conductivity changes to these equations may help to produce better solutions. For example, Eqs. 13-14 have been slightly altered by taking into account the fluid temperature.

$$f(A, \Delta T) = Ax_1 + \Delta T x_2 + T_{fluid} x_3 \quad (13)$$

$$f(A, \Delta T) > x_4 \quad (14)$$

where

T_{fluid} fluid temperature at adjacent cell.

This change in the equations allows the heatsink to build a thickness around the fins in certain areas to lower the thermal resistance along the fins. The heatsink in Fig. 3.12 displays a heatsink designed using this method. Other changes to these equations could better assist in improving heatsink design and performance.

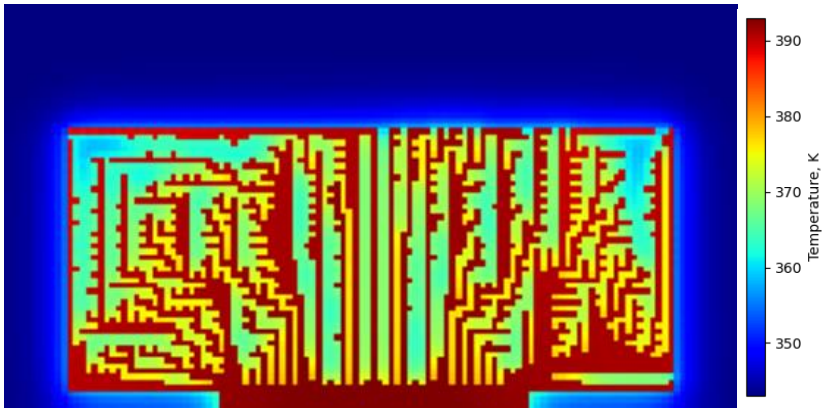

Figure 3.12. Heatsink optimized using different equations.

3.5 Discussion

In this study generatively designed heatsinks optimized by an evolutionary algorithm are compared with standard heatsink designs. As electronic devices continue to advance and become smaller, there is need for more efficient heatsinks that can fit into constrained spaces. Results from this study and previous studies show that generative methods are effective at creating unique heatsinks for constrained geometries. This allows for heatsinks to fit into places which standard heatsinks would not, allowing for smaller electronic devices.

An evolutionary algorithm has shown to be an effective method of optimization for generative design. By training these set equations to place material as the heatsink is generated creates efficient spacing between fins as well as minimizes the distance from the fin to the heat source, allowing for improved heat transfer.

CHAPTER IV

3-D NEURAL NETWORK HEATSINK OPTIMIZATION

4.1 Introduction

The overall goal of this research is to develop a heatsink design optimization method that can increase heat transfer performance while also being efficient and accessible. In Chapter III heatsink optimization was performed using an EA which shows promising results; however, this approach takes a considerable number of simulations to find a solution. If this method were to be combined with 3-D CFD simulations to better model the fluid flow and heat transfer, the computational power necessary would become excessive making this neither efficient nor accessible. A gradient ascent method could be used to optimize the variables instead of the evolutionary algorithm, but with multiple maxima existing within the variable field, a better optimization method is needed. To overcome these issues, a Neural Network (NN) is employed to optimize these variables. This method will allow for 3-D CFD simulation to be incorporated into the optimization while keeping the computational requirements low due to the reduced overall number of simulations needed. Due to the sensitivity analysis optimization not coming from governing equations, this method can be used in the optimization of any problem such as multiphase flow or phase change material heatsink design.

4.2 Numerical Method

Dozens of unique heatsink simulations are needed for a neural network to have enough data to properly begin learning the system. To achieve this a custom CFD code is used focusing on the speed of the solver. While a better solver would provide more accurate results, this is not needed as the optimization will not greatly change due to small differences in accuracy.

The governing partial differential equations used to model the flow field are the incompressible Navier Stokes equations. Respectively, the momentum and continuity equations are

$$\frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} = \frac{-\nabla P}{\rho} + \nu(\nabla^2 \bar{u}) + f \tag{15}$$

$$\nabla \cdot \bar{u} = 0 \tag{16}$$

where

$\bar{u}$ velocity vector

P pressure

ρ density

t time

f external force term.

The incompressible Navier Stokes equations are used due to the low Mach numbers generally seen in heatsinks. This means the change in density of the fluid due to high velocities is negligible. However, changes in density due to temperature will create buoyancy which needs to be accounted for to model natural convection. The Boussinesq approximation, used as an external force term to account for these density differences, is given by

$$f = g\beta(T - T_0) \tag{17}$$

where

g gravity

β volumetric expansion coefficient

T temperature

T_0 reference temperature.

The energy equation used to model the temperature convection and diffusion in both the fluid and solid regions is given by

$$\frac{\partial T}{\partial t} + (\bar{u} \cdot \nabla)T = \frac{\nabla \cdot (k\nabla T)}{\rho C_p} \tag{18}$$

where

k thermal conductivity

C_p specific heat.

To accelerate simulation speed during optimization a Cartesian mesh grid is used so that remeshing is not needed between geometry changes. A finite difference scheme is utilized to model the above equations with a second order central difference discretization in space. The first and second derivative approximations are

$$\frac{\partial u}{\partial x} = \frac{u^n_{i+1,j,k} - u^n_{i-1,j,k}}{2\Delta x} \tag{19}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u^n_{i+1,j,k} - 2u^n_{i,j,k} + u^n_{i-1,j,k}}{\Delta x^2} \tag{20}$$

where

i spatial index

n time index

Δx change in distance between mesh points.

With this discretization a staggered grid is necessary for stability. Figure 4.1 shows a two-dimensional example of the grid where pressure is solved at the black dots, temperature is solved at the red dots, and the arrows represent their respective velocity component. The shaded regions represent the material where it can either be solid or fluid. If a region is solid, then the velocities at the edge are set to zero and the pressure at each corner become Neumann boundaries.

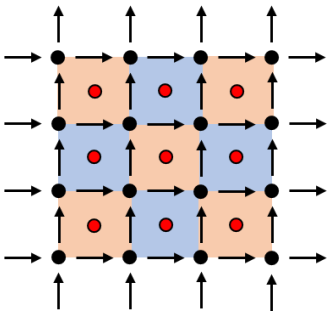

Figure 4.1: Staggered grid with pressure (black dots), temperature (red dots), velocity components (arrows), and material designation.

The momentum equations are discretized according to this grid where the perpendicular velocity components are averaged to determine the velocity at index points they are not directly solved at. The z-direction momentum equation becomes

$$w_{i,j,k}^{n+1} = w_{i,j,k}^{n} + \Delta t\left(-u\frac{\partial w}{\partial x} - v\frac{\partial w}{\partial y} - w\frac{\partial w}{\partial z} - \frac{1}{\rho}\frac{\partial P}{\partial z} + \upsilon\left(\frac{\partial^2 w}{\partial z^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial x^2}\right)\right) \quad (21)$$

where

$$u\frac{\partial w}{\partial x} = \frac{u_{i+0.5,j,k+0.5} + u_{i+0.5,j,k-0.5} + u_{i-0.5,j,k+0.5} + u_{i-0.5,j,k-0.5}}{4}\left(\frac{w_{i,j,k+1} - w_{i,j,k-1}}{2\Delta z}\right) \quad (22)$$

$$\frac{\partial P}{\partial z} = \left(\frac{P_{i,j,k+0.5} - P_{i,j,k-0.5}}{\Delta z}\right). \quad (23)$$

A Poisson equation is then left for solving pressure. The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) [37] is used by correcting the residuals in the continuity equation with the pressure gradient. The residuals are calculated at the same indexes as pressure where

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = \frac{u_{i+0.5,j,k} - u_{i-0.5,j,k}}{\Delta x} + \frac{v_{i,j+0.5,k} - v_{i,j-0.5,k}}{\Delta y} + \frac{w_{i,j,k+0.5} - w_{i,j,k-0.5}}{\Delta z} = residual_{i,j,k}. \quad (24)$$

To fully satisfy continuity for an incompressible flow the residual must be zero at all points to conserve mass. For a numerical solution this needs to be a very small number but not necessarily exactly zero. The discretized pressure Poisson equation to solve for the pressure corrector then becomes

$$P^* = \frac{1}{\alpha}\left(-\frac{residual}{\Delta t} + \frac{P^*_{i+1,j,k} + P^*_{i-1,j,k}}{\Delta x^2} + \frac{P^*_{i,j+1,k} + P^*_{i,j-1,k}}{\Delta y^2} + \frac{P^*_{ijk+1} + P^*_{i,j,k-1}}{\Delta z^2}\right) \quad (25)$$

where

$$\alpha = \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2}. \quad (26)$$

While this pressure solving method works well it does not consider if an index location is fluid or solid. To account for solid regions, boundary conditions would have to be updated at every surface point after each iteration. This can be computationally expensive so an equation that takes into account these boundaries as well as solving for the pressure corrector in one step is ideal. The material array, M, is introduced where a one or zero represent a fluid or solid region respectively. This array is used in the Poisson equation to multiply gradients by one if there is fluid and by zero if the region is solid. By doing this although there is a pressure gradient in the solid regions it does not interfere with the pressure field in the fluid. Using the staggered grid as shown in fig. 1 the discretization leads to

$$P^* = \frac{\left(-\frac{residual}{\Delta t} + \alpha_1 P^*_{i+1,j,k} + \alpha_2 P^*_{i-1,j,k} + \alpha_3 P^*_{i,j+1,k} + \alpha_4 P^*_{i,j-1,k} + \alpha_5 P^*_{i,j,k+1} + \alpha_6 P^*_{i,j,k-1}\right)}{\left(\sum_{m=1}^{6} \alpha_m\right)} \quad (27)$$

where the alpha coefficients are based on if the line between pressure indices is on the edge of a material region which gives

$$\alpha_1 = \frac{M_{i+0.5,j+0.5,k+0.5} * M_{i+0.5,j+0.5,k-0.5} * M_{i+0.5,j-0.5,k+0.5} * M_{i+0.5,j-0.5,k-0.5}}{\Delta x^2} \quad (28)$$

$$\alpha_2 = \frac{M_{i-0.5,j+0.5,k+0.5} * M_{i-0.5,j+0.5,k-0.5} * M_{i-0.5,j-0.5,k+0.5} * M_{i-0.5,j-0.5,k-0.5}}{\Delta x^2} \quad (29)$$

$$\alpha_3 = \frac{M_{i+0.5,j+0.5,k+0.5} * M_{i+0.5,j+0.5,k-0.5} * M_{i-0.5,j+0.5,k+0.5} * M_{i-0.5,j+0.5,k-0.5}}{\Delta y^2} \quad (30)$$

$$\alpha_4 = \frac{M_{i+0.5,j-0.5,k+0.5} * M_{i+0.5,j-0.5,k-0.5} * M_{i-0.5,j-0.5,k+0.5} * M_{i-0.5,j-0.5,k-0.5}}{\Delta y^2} \quad (31)$$

$$\alpha_5 = \frac{M_{i+0.5,j+0.5,k+0.5} * M_{i+0.5,j-0.5,k+0.5} * M_{i-0.5,j+0.5,k+0.5} * M_{i-0.5,j-0.5,k+0.5}}{\Delta z^2} \quad (32)$$

$$\alpha_6 = \frac{M_{i+0.5,j+0.5,k-0.5} * M_{i+0.5,j-0.5,k-0.5} * M_{i-0.5,j+0.5,k-0.5} * M_{i-0.5,j-0.5,k-0.5}}{\Delta z^2} \quad (33)$$

An additional term as shown is Eq. 34 allows for successive over relaxation (SOR). The relaxation factor can range between zero to two to either accelerate or deaccelerate a solution. For the majority of heatsink simulations done for this study a relaxation factor of 1.5 was used to reduce the number of iterations needed between time steps.

$$P^* = P^* + \frac{\omega}{\beta}\left(-\frac{residual}{\Delta t} + \alpha_1 P^*_{i+1,j,k} + \alpha_2 P^*_{i-1,j,k} + \alpha_3 P^*_{i,j+1,k} + \alpha_4 P^*_{i,j-1,k} + \alpha_5 P^*_{i,j,k+1} + \alpha_6 P^*_{i,j,k-1} - \beta P^*\right) \quad (34)$$

where

ω relaxation factor

$$\beta = \sum_{k=1}^{6} \alpha_k. \quad (35)$$

After each time step the velocities of the points on the edge of a solid cell are set to zero to satisfy the no-slip condition. The heat diffusion in the energy equation would generally be discretized using the central difference scheme shown in Eq. 20. However, if two cells have two different thermal conductivities, this equation is invalid. To solve for the heat diffusion a central difference scheme is still used except it is changed to calculate the second derivative only within the cell as

shown in Eq. 36. This prevents changes in thermal conductivity within the second derivative calculation.

$$\frac{\partial^2 T}{\partial x^2} = \frac{T^n_{i+\frac{1}{2},j,k} - 2T^n_{i,j,k} + T^n_{i-\frac{1}{2},j,k}}{(\Delta x/2)^2} \tag{36}$$

A thermal resistance network is used to calculate the temperatures in between index points where

$$T_{i+\frac{1}{2},j,k} = \frac{k_{i+1,j,k}T_{i+1,j,k} + k_{i,j,k}T_{i,j,k}}{k_{i+1,j,k} + k_{i,j,k}}. \tag{37}$$

Each time the material in the system is changed the thermal conductivities array is updated. For the convection term of the energy equation the velocities of the nearest velocity components are averaged to determine the velocity at the point the temperature is solved at. This gives the discretized energy equation as

$$T^{n+1}_{i,j,k} = T^n_{i,j,k} + \Delta t\left(-u\frac{\partial T}{\partial x} - v\frac{\partial T}{\partial y} - w\frac{\partial T}{\partial z} + \frac{1}{\rho C_p}\left(\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)\right)\right) \tag{38}$$

where

$$u\frac{\partial T}{\partial x} = \frac{u_{i,j+1/2,k+1/2} + u_{i,j+1/2,k-1/2} + u_{i,j-1/2,k+1/2} + u_{i,j-1/2,k-1/2}}{4}\left(\frac{T_{i+1,j,k} - T_{i-1,j,k}}{2\Delta x}\right) \tag{39}$$

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) = k_{i,j,k}\left(\frac{\frac{k_{i+1,j,k}T_{i+1,j,k} + k_{i,j,k}T_{i,j,k}}{k_{i+1,j,k} + k_{i,j,k}} - 2T_{i,j,k} + \frac{k_{i+1,j,k}T_{i+1,j,k} + k_{i,j,k}T_{i,j,k}}{k_{i+1,j,k} + k_{i,j,k}}}{(\Delta x/2)^2}\right). \tag{40}$$

There are many different boundary conditions which could be used for this model dependent on the heatsink setup. For our purposes, only half of the heatsink is modeled to reduce computational time required. The total domain, including the heatsink and the fluid domain, is a rectangular prism with XYZ dimensions of 7.2 x 15 x 10 cm. The half of the heatsink modeled takes up a maximum of 3 x 6 x 3 cm and is placed against the Z=0 and X=0 planes. The boundary conditions on each plane are as follows where Y=0, Y=15 and X=7.2 are inlets, Z=10 is an outlet, Z=0 and X=0 are walls:

$$\left.\frac{\partial T}{\partial z}\right|_{Z=0} = 0 \tag{41}$$

$$\left.\frac{\partial T}{\partial z}\right|_{Z=10} = 0 \tag{42}$$

$$\left.\frac{\partial T}{\partial x}\right|_{X=0} = 0 \tag{43}$$

$$T|_{X=7.2} = 295\ K \tag{44}$$

$$T|_{Y=0} = 295\ K \tag{45}$$

$$T|_{Y=15} = 295\ K \tag{46}$$

$$\left.\frac{\partial P}{\partial z}\right|_{Z=0} = 0 \tag{47}$$

$$P|_{Z=10} = 0 \tag{48}$$

$$\left.\frac{\partial P}{\partial x}\right|_{X=0} = 0 \tag{49}$$

$$P|_{X=7.2} = 0 \tag{50}$$

$$P|_{Y=0} = 0 \tag{51}$$

$$P|_{Y=15} = 0 \tag{52}$$

$$\bar{u}|_{Z=0} = 0 \tag{53}$$

$$\nabla\bar{u}|_{Z=10} = 0 \tag{54}$$

$$\bar{u}|_{X=0} = 0 \tag{55}$$

$$\nabla\bar{u}|_{X=7.2} = 0 \tag{56}$$

$$\nabla\bar{u}|_{Y=0} = 0 \tag{57}$$

$$\nabla \bar{u}|_{Y=15} = 0 \tag{58}$$

Star-CCM+ [38], a commercial CFD solver, is also utilized to determine the optimization effectiveness and compare with experimental data. Once validated against experimental data, heatsink designs can be tested using a high order commercial CFD solver on a fine mesh before any manufacturing is necessary. A polyhedral mesher is used to mesh both the heatsink and the fluid domain. To model the flow, a 3-D, steady state, K-Omega Reynolds-Averaged Navier-Stokes turbulent solver is used with the Ideal Gas equation to close out the model.

4.3 Stability Analysis

Von Neumann stability analysis [39] is applied to the discretized equations to calculate the maximum time step per iteration. This is done by expanding the finite difference equation in a Fourier series to determine the amplification. If the amplification between timesteps is greater than one, then the scheme is unstable. Variables such as the timestep must be adjusted so that the amplification at any phase angle is less than one. This process is applied to Eq. 38 by first expanding the components into a finite Fourier series where

$$T_i^n = \sigma^n e^{I(\theta i + \phi j + \gamma k)} \tag{59}$$

where

I imaginary unit

σ amplitude

θ, ϕ, γ phase angles.

Equation 38 combined with Eq. 59 becomes

$$\sigma^{n+1}e^{I(\theta i+\phi j+\gamma k)} \tag{60}$$

$$= \sigma^n e^{I(\theta i+\phi j+\gamma k)} + a\left(\sigma^n e^{I(\theta(i+1)+\phi j+\gamma k)} - \sigma^n e^{I(\theta(i-1)+\phi j+\gamma k)}\right)$$

$$+ b\left(\sigma^n e^{I(\theta i+\phi(j+1)+\gamma k)} - \sigma^n e^{I(\theta i+\phi(j-1)+\gamma k)}\right)$$

$$+ c\left(\sigma^n e^{I(\theta i+\phi j+\gamma(k+1))} - \sigma^n e^{I(\theta i+\phi j+\gamma(k-1))}\right)$$

$$+ d\left(\sigma^n e^{I(\theta(i+1)+\phi j+\gamma k)} - 2\sigma^n e^{I(\theta i+\phi j+\gamma k)}\right.$$

$$\left.+ \sigma^n e^{I(\theta(i-1)+\phi j+\gamma k)}\right)$$

$$+ e\left(\sigma^n e^{I(\theta i+\phi(j+1)+\gamma k)} - 2\sigma^n e^{I(\theta i+\phi j+\gamma k)}\right.$$

$$\left.+ \sigma^n e^{I(\theta i+\phi(j-1)+\gamma k)}\right)$$

$$+ f\left(\sigma^n e^{I(\theta i+\phi j+\gamma(k+1))} - 2\sigma^n e^{I(\theta i+\phi j+\gamma k)}\right.$$

$$\left.+ \sigma^n e^{I(\theta i+\phi j+\gamma(k-1))}\right)$$

which after eliminating $e^{I(\theta i+\phi j+\gamma k)}$ and applying trigonometric relations simplifies to

$$G = \frac{\sigma^{n+1}}{\sigma^n} = [1 + 2d(\cos(\theta) - 1) + 2e(\cos(\phi) - 1) + 2f(\cos(\gamma) - 1)] \tag{61}$$

$$+ I[a(2\sin(\theta)) + b(2\sin(\phi)) + c(2\sin(\gamma))]$$

where

G amplification factor

$$a = \frac{-u\Delta t}{2\Delta x} \tag{62}$$

$$b = \frac{-v\Delta t}{2\Delta y} \tag{63}$$

$$c = \frac{-w\Delta t}{2\Delta z} \tag{64}$$

$$d = \frac{k_x \Delta t}{\rho C_p \Delta x^2} \tag{65}$$

$$e = \frac{k_y \Delta t}{\rho C_p \Delta y^2} \tag{66}$$

$$f = \frac{k_z \Delta t}{\rho C_p \Delta z^2}. \tag{67}$$

For any possible phase angle, G must be less than one to provide a stable solution. In the real-complex plane, the hypotenuse length then becomes

$$|G| = [[1 + 2d(\cos(\theta) - 1) + 2e(\cos(\phi) - 1) + 2f(\cos(\gamma) - 1)]^2 + [a(2\sin(\theta)) + b(2\sin(\phi)) + c(2\sin(\gamma))]^2]^{\frac{1}{2}}. \tag{68}$$

Equation 38 is the limiting equation requiring the smallest timestep when solving for the temperature field of the air.

To satisfy Eq. 68 the timestep must be at or less than 0.14 milliseconds. With this small of a timestep, other equations are not as efficient since their solution speed will be limited to the

smallest timestep. Equation 38 is also used to solve for the temperature field in the solid portions. Here the velocity components go to zero leaving you with

$$T_{i,j,k}^{n+1} = T_{i,j,k}^{n} + \frac{\Delta t}{\rho C_p}\left(\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)\right). \tag{69}$$

Since these simulations run until a steady state solution is reached where $T_{i,j,k}^{n+1} \approx T_{i,j,k}^{n}$ the ρC_p term does not affect the final solution, only the thermal inertia during the transient period. These values can be altered for the solid portion to accelerate the speed of the solution. For our aluminum properties used and the timestep limitation the speed of the solution can be represented by

$$\frac{\Delta t}{\rho C_p} = 5.68 * 10^{-11} \tag{70}$$

However, with the stability requirement of

$$\frac{k_x \Delta t}{\Delta x^2 \rho C_p} + \frac{k_y \Delta t}{\Delta y^2 \rho C_p} + \frac{k_z \Delta t}{\Delta z^2 \rho C_p} \leq 0.5 \tag{71}$$

the value of ρC_p can be brought down from 2,466,100 $\frac{J}{m^3 K}$ to 168,350 $\frac{J}{m^3 K}$ giving the solution speed of:

$$\frac{\Delta t}{\rho C_p} = 8.32 * 10^{-10} \tag{72}$$

This allows the solid portion to be solved approximately 14.7 times faster than if the original material properties had been applied. Other methods could be utilized where a separate steady state equation is solved for the solid portion. This approach may be quicker in many circumstances, however the approach used for this study vectorizes these equations to solve on a GPU which in this case allows the method detailed above to be the preferred method.

4.4 Optimization

The standard approach in topology optimization defines a sensitivity analysis based on the governing equations. Often, assumptions are made to simplify these equations for specific scenarios, or due to the difficulty of solving them. For this research a sensitivity analysis equation is created by including information that may help find better solutions. For these heatsinks, the sensitivity analysis used as shown in Eq. 72 considers the fluid velocity, convection, conduction.

$$S = x_1\bar{u} + x_2(\bar{u} \cdot \nabla)T + x_3\nabla \cdot (k\nabla T) + x_4 T \tag{73}$$

Generally, a gradient descent method would be used to optimize the variables in this sensitivity analysis. This is good approach and would work, however this method can be slow due to the small change in the variables between each iteration and the number of simulations required to

calculate the gradient. Training a neural network on these variables to output the heat transfer as shown in Fig. 4.2 can map out the estimated heat transfer field.

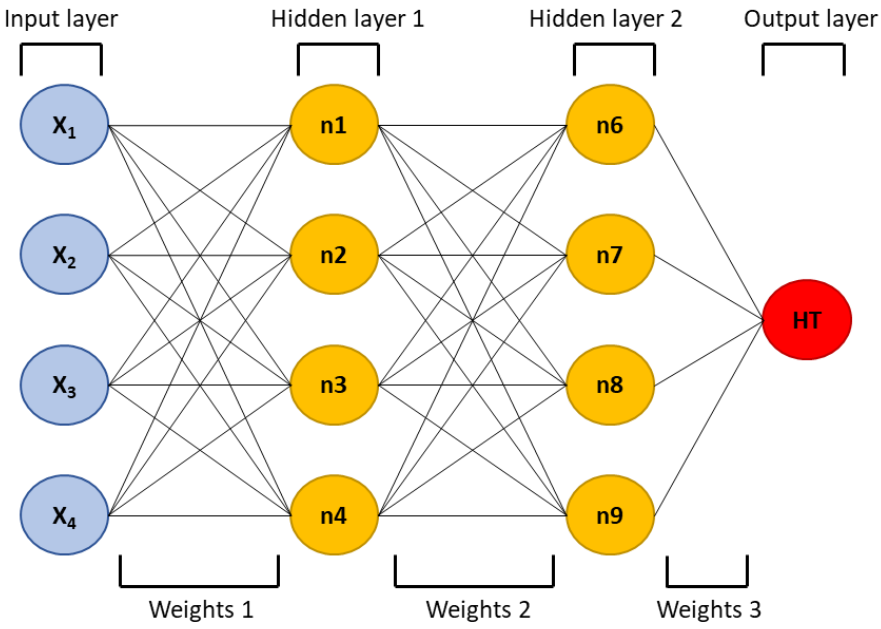

Figure 4.2: Neural network used in this study consisting of an input layer, two hidden layers, and an output layer with sigmoidal activation function.

A neural network works through two main steps, forward and backward propagation. The neural network is initialized with a random value for each weight and bias. To begin forward propagation at the initial layer's matrix is multiplied by the weight matrix. The input layer is given as

$$\bar{X} = \begin{bmatrix} X_1^1 & \cdots & X_n^1 \\ \vdots & \ddots & \vdots \\ X_1^i & \cdots & X_n^i \end{bmatrix} \tag{74}$$

where

i number of input sets

n number of nodes in input layer

$\bar{X}$ input array.

The initial layer of weights, weights one, is given as

$$\bar{W}_1 = \begin{bmatrix} W_1^1 & \cdots & W_1^m \\ \vdots & \ddots & \vdots \\ W_n^1 & \cdots & W_n^m \end{bmatrix} \tag{75}$$

where

m number of nodes in hidden layer one

$\bar{W}_1$ weights in the weight 1 layer.

Additionally, each node has an associated bias given as

$$\bar{B}_1 = \begin{bmatrix} B_1^1 & \cdots & B_1^m \\ \vdots & \ddots & \vdots \\ B_n^1 & \cdots & B_n^m \end{bmatrix} \tag{76}$$

where

$\bar{B}_1$ weights in the weight 1 layer.

A sigmoidal function is then applied to the hidden layer matrix, $\bar{N}_1$, to normalize the matrix to a number between zero and one. This matrix is calculated as

$$\bar{N}_1 = \sigma(\bar{X} \cdot \bar{W}_1 + \bar{B}_1) \tag{77}$$

where

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{78}$$

These steps are repeated for each layer until the final output layer is given as

$$\overline{HT} = \sigma(\bar{N}_2 \bar{W}_3 + \bar{B}_3) \tag{79}$$

where

$\overline{HT}$ output estimated heat transfer for each input set.

This is the end of the forward propagation step where the final heat transfer estimates are output for each input set. To train the weights to minimize the difference between the heat transfer estimate and the simulation data backward propagation is used. Backward propagation applies a gradient descent to each weight and bias in the network to minimize the cost function where

$$\bar{C} = \frac{1}{2}(\bar{T} - \overline{HT})^2 \tag{80}$$

where

$\bar{C}$ cost function array

$\bar{T}$ target array of heat transfer values from simulation.

The derivative of the cost function in terms of each weight and bias is then calculated using the chain rule. For example, the derivative of the cost function in respect to one of the weights directly before the output layer would be

$$\frac{\partial \bar{C}}{\partial w_3} = \frac{\partial \bar{C}}{\partial \overline{HT}} * \frac{\partial \overline{HT}}{\partial Out_{HL2}} * \frac{\partial Out_{HL2}}{\partial w_3}. \tag{81}$$

Since the cost function is the difference between the neural network output and the desired input, we want to minimize this. To do this we subtract the gradient from the weight or bias to lower the cost. The gradient is multiplying by alpha, a learning rate to smooth the gradient descent and increase stability. After each gradient is calculated, this is applied to every weight and bias before beginning the process again with forward propagation. For our above example, w_3 would become

$$w_3 = w_3 - \alpha \frac{\partial \bar{C}}{\partial w_3} \tag{82}$$

where

α learning rate.

By mapping out the variable field with a neural network, the values of the different variables that create the highest heat transfer value can easily be calculated. These new values can create a new heatsink and the simulation data fed back into the neural network to continue training after each iteration.

To increase the speed of this method a lattice design is used for the initial heatsink geometry. A sensitivity equation is calculated at every fluid cell touching the heatsink and material is added where the sensitivity is the lowest by

$$s = x_1\left(\frac{U}{\max(U)}\right) + x_2\left(\frac{(\bar{u}\cdot\nabla)T}{\max((\bar{u}\cdot\nabla)T)}\right) + x_3\left(\frac{cond}{\max(cond)}\right) + x_4\left(\frac{T}{\max(T)}\right) \quad (83)$$

where

cond conduction, magnitude of $k\nabla T$

conv convection within the cell

U velocity magnitude

x_k variables to optimize.

Between each iteration of adding a set amount of material, the simulation is run until a steady state solution is reached until final criteria are met. Dependent on the desired output, final criteria could be weight, heat transfer, or cost.

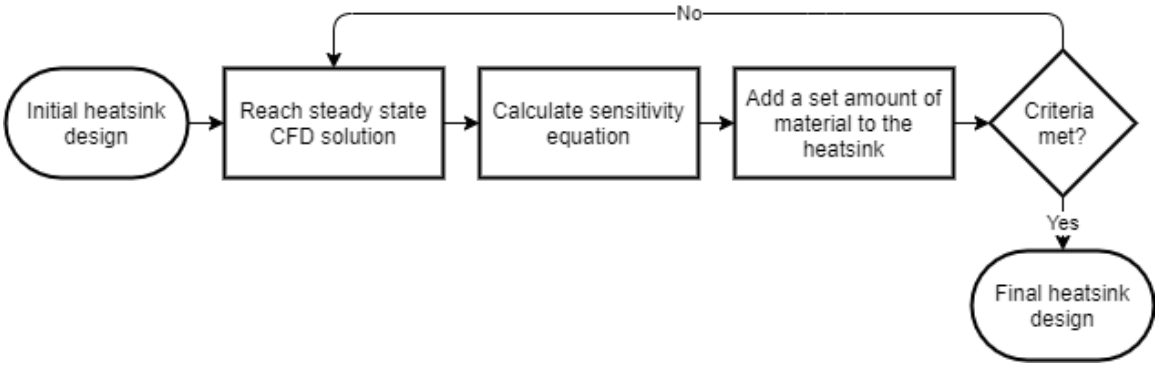

Figure 4.3: Design optimization process flowchart.

Initially all the function variables, x_1, x_2, x_3 and x_4, are set to a random number between zero and one. The heatsink goes through the design optimization process of adding material and a final heat transfer output is recorded. This process is repeated five times with random numbers to initialize the neural network. The neural network is trained based on the function variables as inputs of each heatsink and the heat transfer as the desired output. The neural network can then quickly test each different possible set of function variable values in small steps and output the

variables that give the highest predicted heat transfer. Figures 4.4-4.6 shows cut scenes of the neural network heat transfer output while varying different function variables.

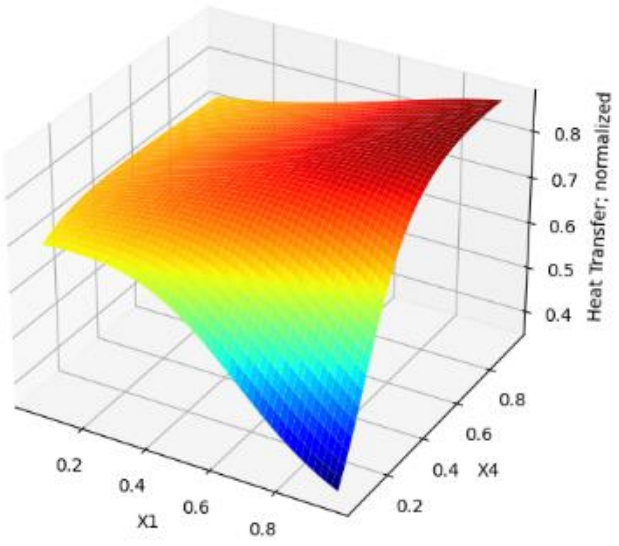

Figure 4.4: Cut scene of the neural network output varying x_1 and x_4.

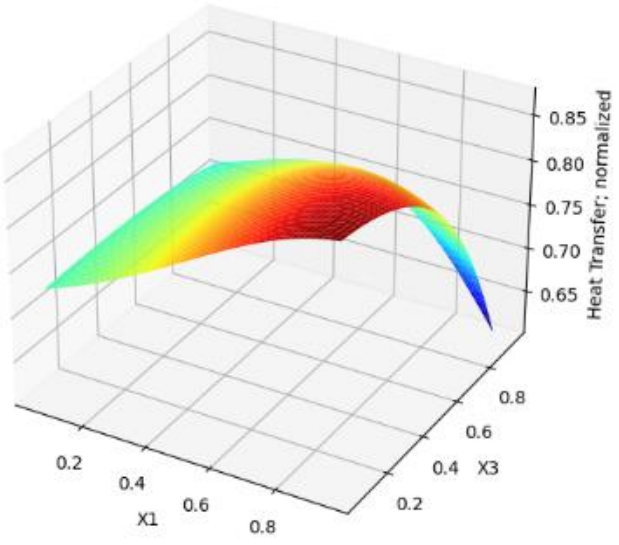

Figure 4.5: Cut scene of the neural network output varying x_1 and x_3.

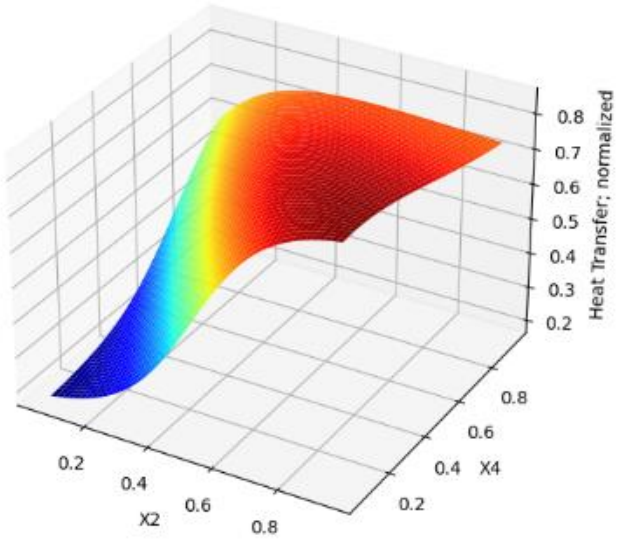

Figure 4.6: Cut scene of the neural network output varying x_2 and x_4.

After the neural network outputs the function variables that will give the highest heat transfer value, these variables are put through the design optimization process to determine the actual heat transfer value. This value is then used to retrain the neural network and the process is repeated. Initially, the neural networks outputs are inaccurate, and do not provide function variables which improve the heat transfer. As the training progresses, the accuracy improves and a solution that increases the heat transfer is found. Figure 4.8 shows an example of this process. The first five iterations only have values of the simulated data to train the neural network with. As the process moves on, by the twelfth iteration the neural network has outputted variables which matched the simulation data and improved the heat transfer above any other variable set tested.

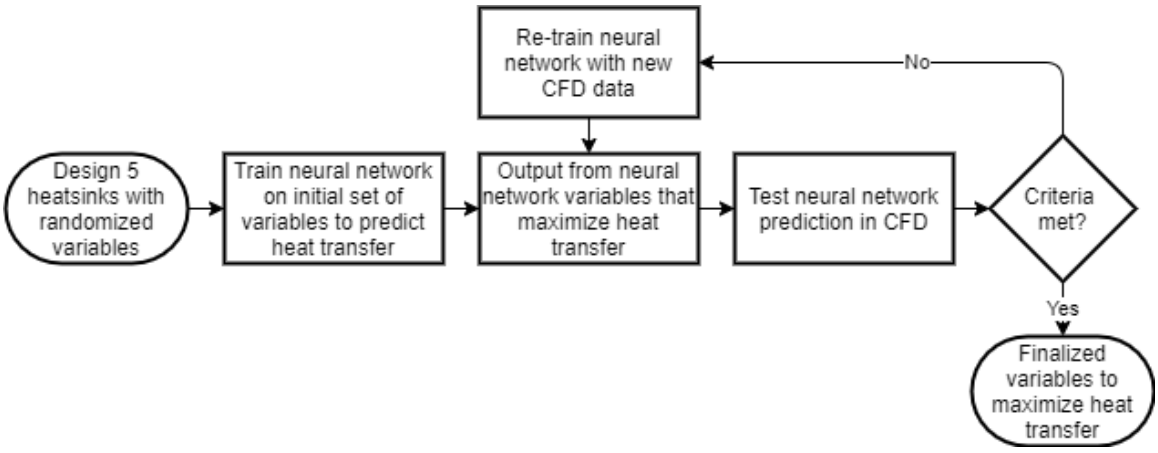

Figure 4.7: Neural network training flowchart.

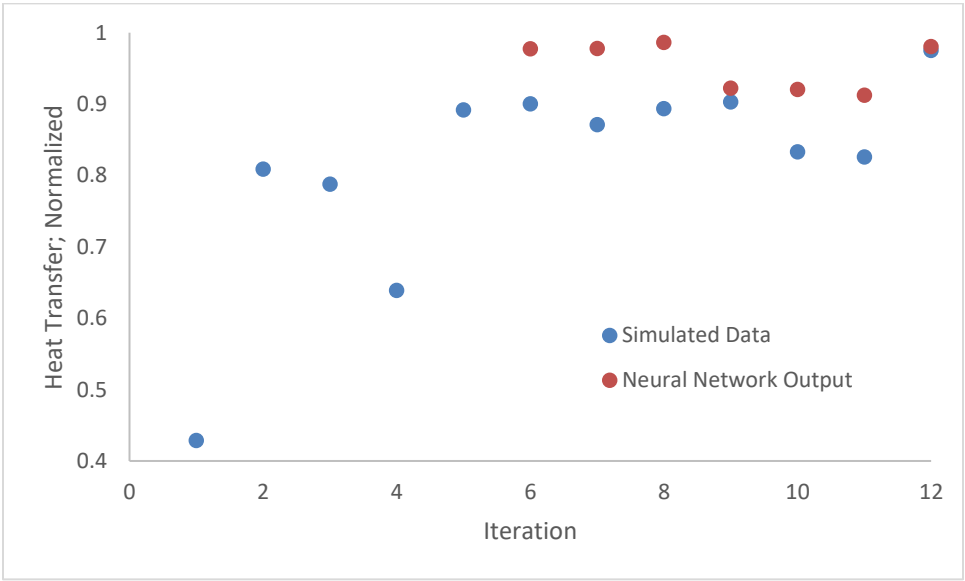

Figure 4.8: Neural network output compared to simulation data.

The heat transfer performance and heatsink design is dependent on the initial material layout. An initial material layout such as a lattice design, or pin fins are placed around the edges of the heatsink, provide quicker optimization times. This is not a necessary step, however if this is not done the material added per iteration needs to be a small amount which increases the total iterations and simulations needed. With pins added around the edges of the heatsink only a few iterations of adding material are needed. The process of how to initially setup the heatsink and

how much material to add between iterations is different for every heatsink and needs to be adjusted accordingly.

4.5 Experimental Analysis

Experimental validation is necessary to determine if the simulation and optimization methods used were successful in providing a design with increased heat transfer over conventional designs. The custom simulation data is not expected to be very accurate due to the low mesh resolution and assumptions used to have fast enough simulations to optimize the heatsink. This may lead to an inaccurate optimization if the fluid and temperature field are incorrect. Experimental results as well as commercial simulation software will validate the ability of this heatsink design optimization method to produce increased performance heatsink designs.

An experimental setup to measure the base temperature and power transfer of these heatsinks is used consisting of an aluminum block, four thermocouples, a cartridge heater, a voltage regulator, thermal mats, insulation, a data acquisition system, and various structural components. The aluminum block has multiple thermocouple probe locations to measure the heatsink base temperature and the temperature difference within the block to calculate the thermal power transfer. Insulation is added to completely cover the outer walls so that 1-D heat transfer can be assumed when calculating the power.

Accurate temperature measurements are needed to calculate the thermal power transfer through the aluminum block shown in Fig 4.9 Two different thermocouples are used to measure the temperatures at two different heights.

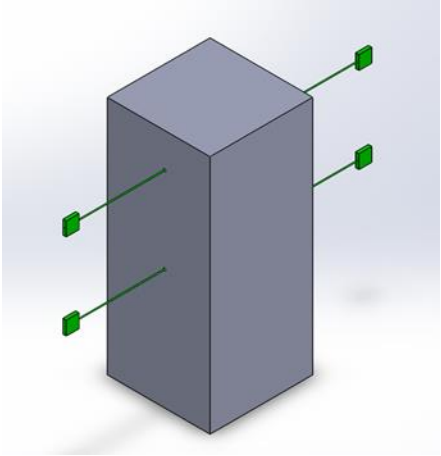

Figure 4.9: Aluminum block with thermocouples.

The accuracy of temperature measurements can be significantly off based on the thermocouples, connectors, wiring, and data acquisition unit. For this setup using Eq. 84 the expected temperature drop is about two degrees C.

$$\dot{Q} = kA\frac{\Delta T}{\Delta x} \tag{84}$$

where

A cross sectional area

$\dot{Q}$ heat transfer

ΔT change in temperature

Δx distance between thermocouples.

Due to the small expected temperature difference if both the thermocouples were off by one degree C this could create an error up to 50% in the heat transfer calculation. To increase the accuracy of this calculation the thermocouples and data acquisition system bias will need to be determined. The actual temperature reading is not important as all that is needed is the change in temperature. This makes the process easier as the thermocouple readings can be normalized to

one set thermocouple so that accurate differences can be measured instead of calibrating to a specific temperature.

The thermocouple bias determination was performed by placing all the thermocouples in a pot of distilled water at the same height. Insulation is used to cover the top of the pot to increase the temperature uniformity within the water. The water is then heated to about 80° C and allowed to cool down from there. Thermocouple measurements are recorded every five seconds until the water has cooled to 30° C. Figure 6 shows recorded data of five thermocouples over a portion of this process.

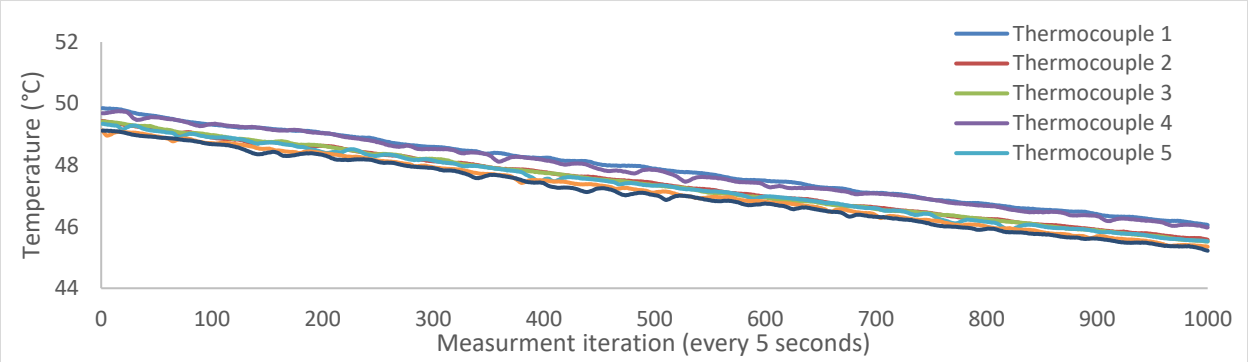

Figure 4.10: Thermocouple measurements in heated water.

While there are consistent trends the data has inconsistencies due to temperature fluctuations within the water. A moving average is applied to smooth the data and remove these fluctuations. Each thermocouple measurement is then subtracted from the measurement of thermocouple one to see the difference in temperature reading. Figure 4.11 shows the smoothed differences in temperature of the thermocouples as compared to the reading of thermocouple one.

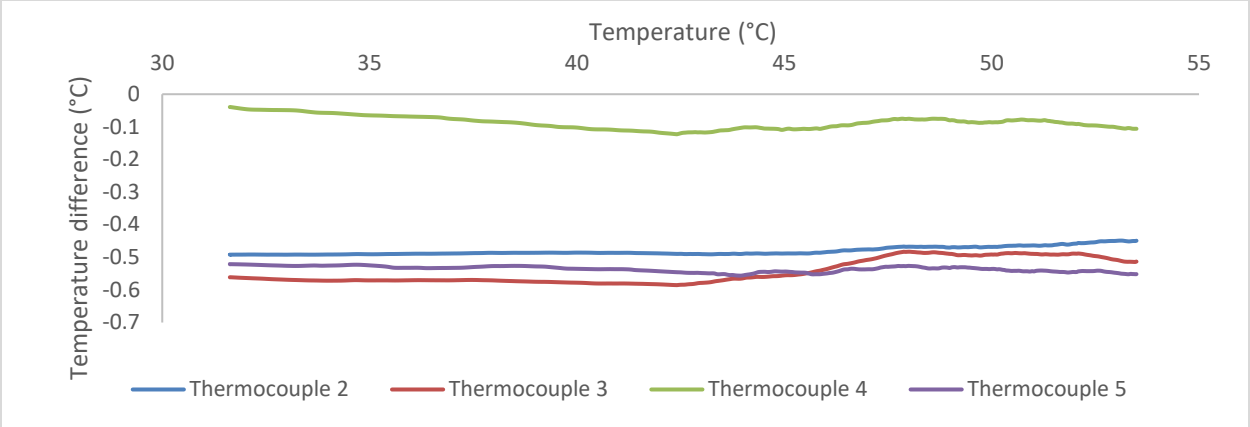

Figure 4.11: Thermocouple measurement differences over a range of temperatures.

The overall difference is consistent giving a constant number to use as a correction term. This experiment is then repeated to ensure similar results. Figure 4.12 highlights the results from the two different trials showing the temperature reading difference between thermocouples is constant and a correction can be applied to correct this.

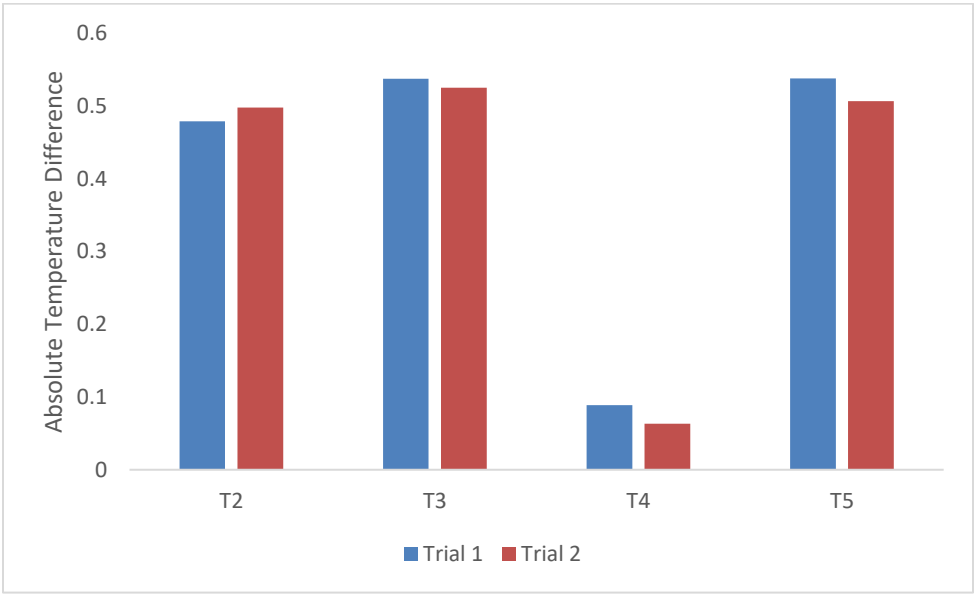

Figure 4.12: Thermocouple measurement differences from two different trials.

A voltage regulator is used to control the power going to a cartridge heater within the aluminum block. The voltage can be held constant for each heatsink to keep the power level the same or

adjusted to match until a desired base temperature is reached. Because the base temperature cannot be exactly matched for each heatsink, the thermal resistance per weight will be compared between heatsinks.

$$C = \frac{T_{base} - T_{amb}}{Wg} \tag{85}$$

where

C Thermal resistance per gram

g weight of the heatsink

T_{base} Heatsink base temperature

T_{amb} Ambient air temperature

W Power dissipated by the heatsink.

Figure 4.13. shows the experimental setup including the voltage regulator, insulated heatsink test box, and the data acquisition unit. The heatsink test box is placed inside of a closed off environment to prevent any air currents or other factors from influencing the heat transfer.

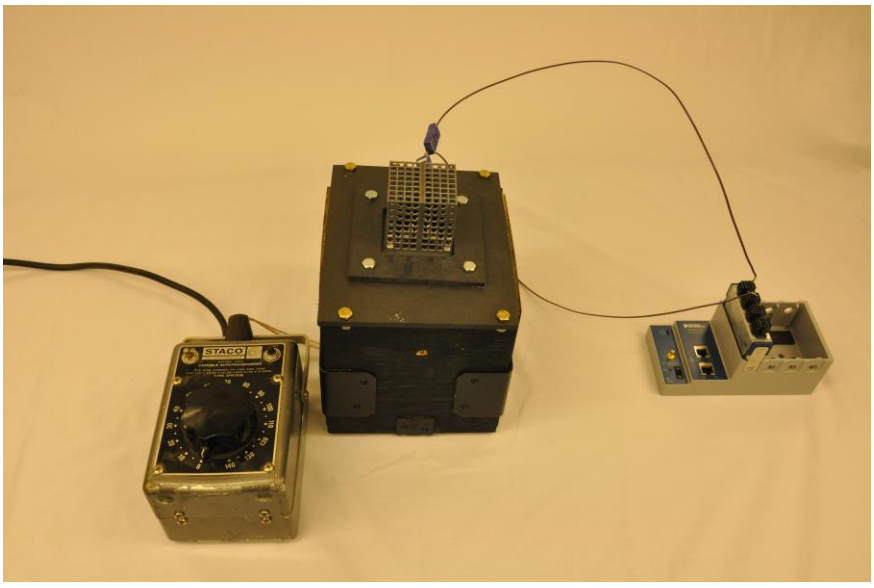

Figure 4.13: Heatsink experimental setup.

Originally, tests were performed with thermal paste between the interface of the aluminum block and the heatsink. Three trials were performed using the same heatsink and input power. Between trials the heatsink was removed from the test box, cleaned, and new thermal paste applied. With additively manufactured heatsinks, the rough surfaces mean the contact resistance is highly dependent on the smoothness and amount of thermal paste. To keep these process consistent, thermal mats were used in place of thermal paste which showed considerably better test consistency.

Table 4.1: Heat transfer consistency of thermal paste vs. thermal mats.

	Trial 1 (W)	Trial 2 (W)	Trial 3 (W)
Thermal paste	22.1	26.9	22.8
Thermal mat	24.65	23.81	25.13

4.6 Results

Initially, a heatsink is optimized to determine if the custom CFD code and mesh resolution is adequately refined, as well as to test the additive manufacturing process and experimental setup. Experimental data is then used to validate simulations from a commercial CFD software. The design is initialized as a lattice structure where the neural network will decide where to add material to maximize heat transfer. A custom code is written to convert the voxel array into a point cloud and to automatically fillet certain edges to smooth the heatsink. A separate software is then used to smooth the edges and create an STL from the point cloud.

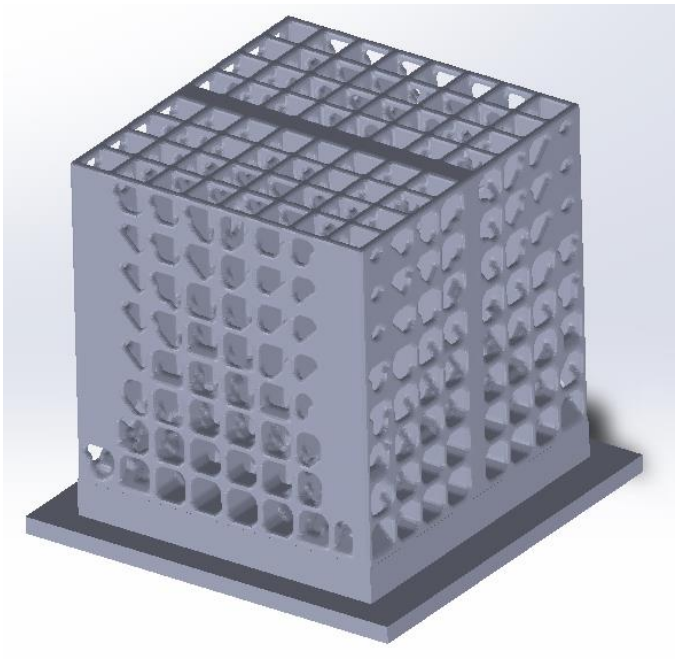

Figure 4.14: Optimized lattice heatsink design.

This optimized lattice heatsink as well as a parallel fin heatsink with optimized spacing are additively manufactured using Laser Powder Bed Fusion (LPBF). They are printed out of an aluminum alloy, AlSi10Mg. This alloy has excellent thermal properties and is a relatively well priced option compared to copper and other AM metals with similar thermal properties.

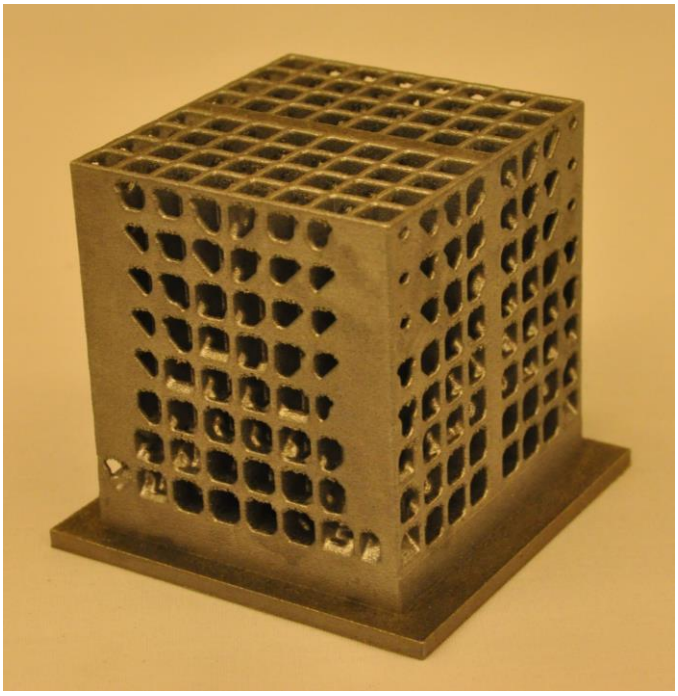

Figure 4.15: Additively manufactured optimized lattice heatsink.

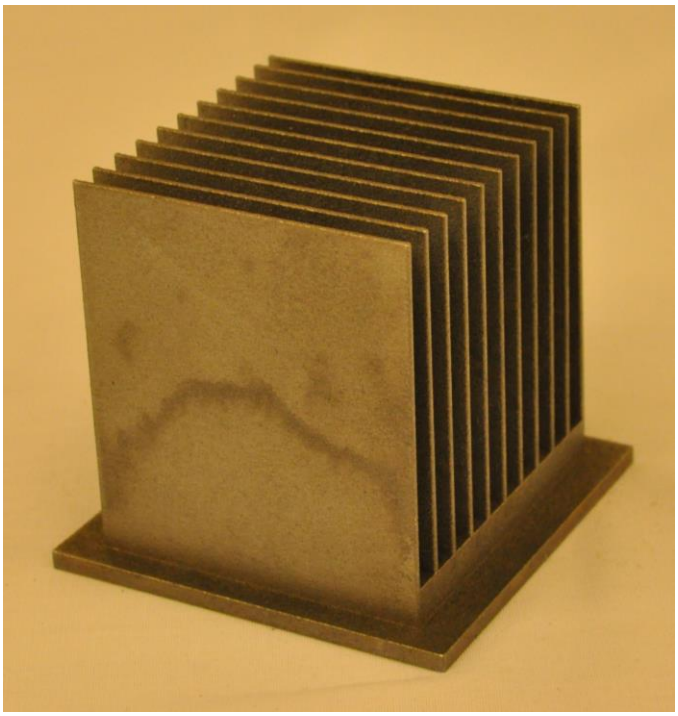

Figure 4.16: Additively manufactured parallel fin heatsink.

Experimental testing was performed at 50 V with a calculated power output of 26.04 W from the cartridge heater. Two thermocouple locations within the aluminum block are used to measure the power going to the heatsink, and one of the thermocouples is measuring the base temperature. The commercial CFD is given the input power as measured from the experiment and outputs the base temperature. Experimental testing of these heatsinks showed the custom CFD code mesh resolution was not refined enough to provide an effective optimization. This testing did however validate the commercial CFD software used.

Table 4.2: Experimental vs. CFD comparison of the parallel fin heatsink.

	Experimental data	Commercial CFD	Percent error
Input power (W)	25.13	25.13	
Base temperature (K)	366.74	368.26	0.41%
ΔT (K)	71.74	73.26	2.12%

To provide an effective optimization process the custom CFD code mesh resolution was refined by decreasing the cell size 40% from the original mesh used in the lattice heatsink optimization. Due to this, increased computational time would be necessary to optimize a lattice heatsink adding material in 3 dimensions. Instead, a 2-D design is optimized, still utilizing 3-D CFD, to keep the computational requirements low. Two new heatsinks are optimized with pin fins around the edge to accelerate the optimization and training process. The first heatsink, heatsink A, is optimized to maximize heat transfer while holding the same weight as the parallel fin heatsink. The second heatsink, heatsink B, is optimized to minimize weight while holding about the same heat transfer as the parallel fin heatsink.

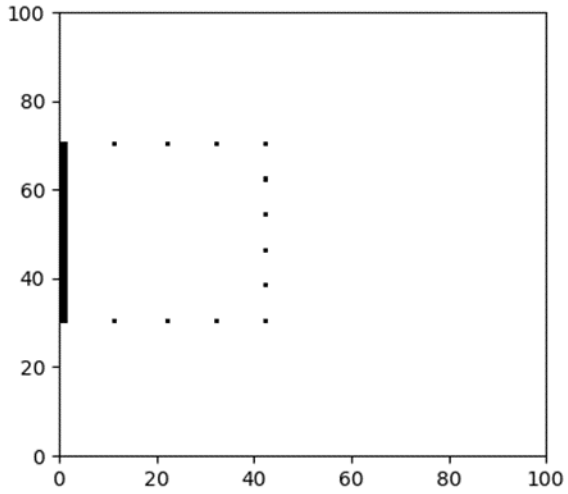

Figure 4.17: Initial pin fin material layout of heatsink A.

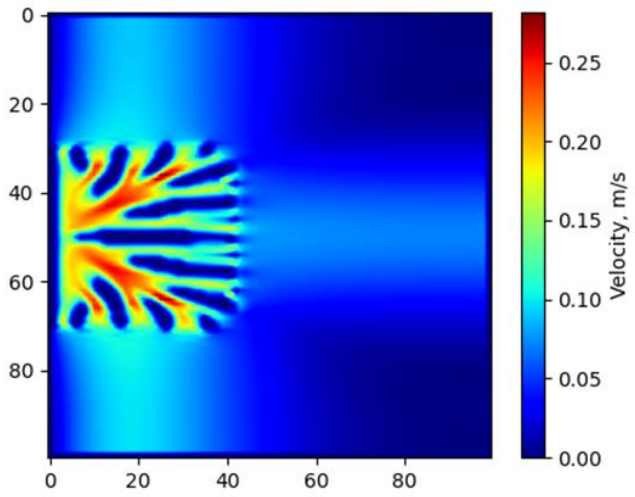

Figure 4.18: Post-optimization velocity cut scene of heatsink A.

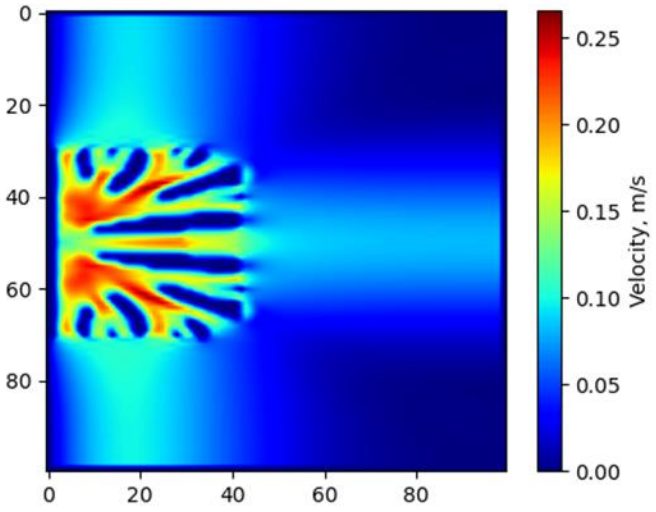

Figure 4.19: Post-optimization velocity cut scene of heatsink B.

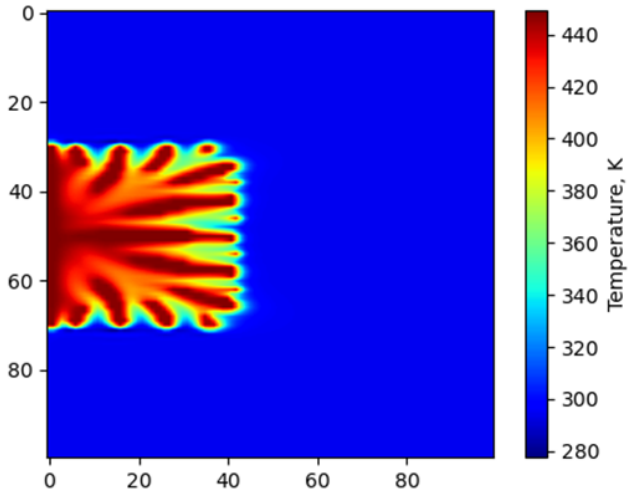

Figure 4.20: Post-optimization temperature cut scene of heatsink A.

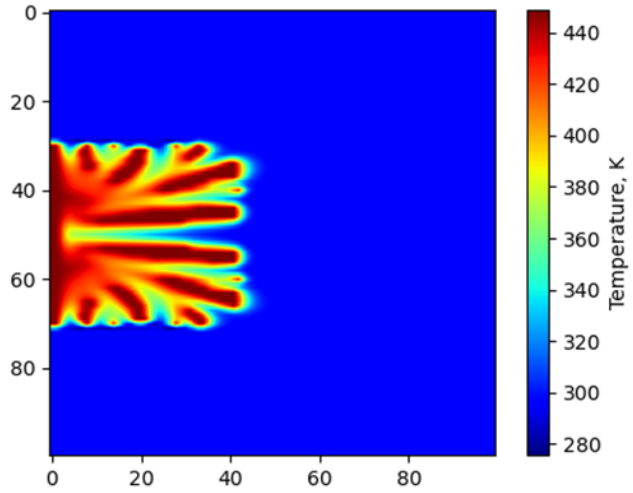

Figure 4.21: Post-optimization temperature cut scene of heatsink B.

Each of these heatsinks were tested in the commercial CFD software with the exact same physics and mesh settings as used when validating the previous results with experimental data. The heatsinks were optimized for and set to a constant base temperature of 450 K. As shown in Table 4.3, heatsink A was able to increase the heat transfer by 10% while maintaining the same weight as the parallel fin heatsink. Since heatsink B and the parallel fin heatsink do not have the same heat transfer, the specific power used for comparison. Heatsink B increased the specific power by 37% over the parallel fin heatsink.

Table 4.3: Commercial CFD results of optimized heatsinks.

	Parallel fin	Heatsink A	Heatsink B
Weight (g)	59.75	59.75	41.90
Heat transfer (W)	35.92	39.67	34.61
Specific power (W/g)	0.601	0.664	0.826

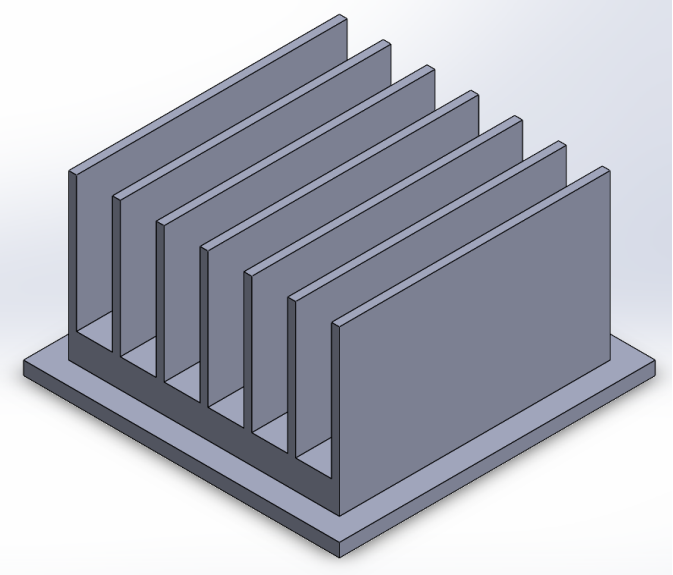

Figure 4.22: Parallel fin heatsink final design.

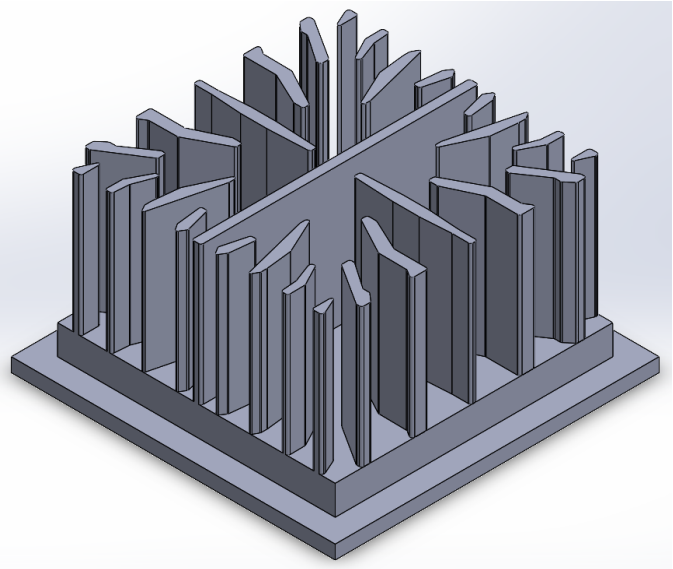

Figure 4.23: Heatsink A final design after smoothing.

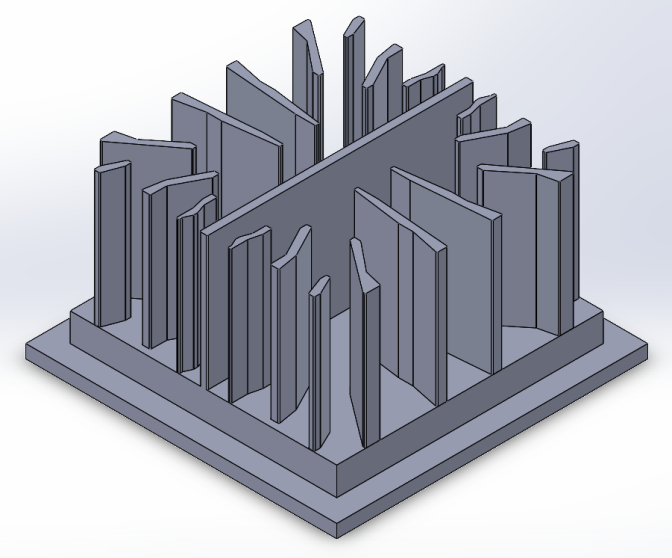

Figure 4.24: Heatsink B final design after smoothing.

4.7 Discussion

The objective of this study was to utilize a NN to optimize a sensitivity equation to maximize real world heatsink performance. Initial results of additively manufactured optimized heatsinks showed an inaccurate optimization, however results from a commercial CFD solver were validated against experimental data. With mesh refinement to improve the optimization accuracy, results in the commercial CFD software showed successfully optimized heatsinks for either heat transfer improvement or weight reduction.

These substantial increases in performance shown over standard parallel fin heatsinks could have significant impacts in industry. Weight reduction is critical in aerospace applications and smaller heatsinks could help lead to higher efficiencies in compact electronics such as phones and laptops.

CHAPTER V

CONCLUSION

5.1 Summary

The goal of this study was to test and create novel heatsink design optimization methods that can increase heat transfer performance while being efficient and accessible. Many of the current methods that exist in industry constrain the heatsink geometry to parallel fins or pin fins, which work well but do not fully take advantage of additive manufacturing capabilities. Methods which do not constrain the geometry such as topology optimization are computationally expensive and not readily available to industry. By testing novel heatsink design optimization methods, the objective is to find a middle ground that has the benefits of topology optimization while keeping the time and computational costs low.

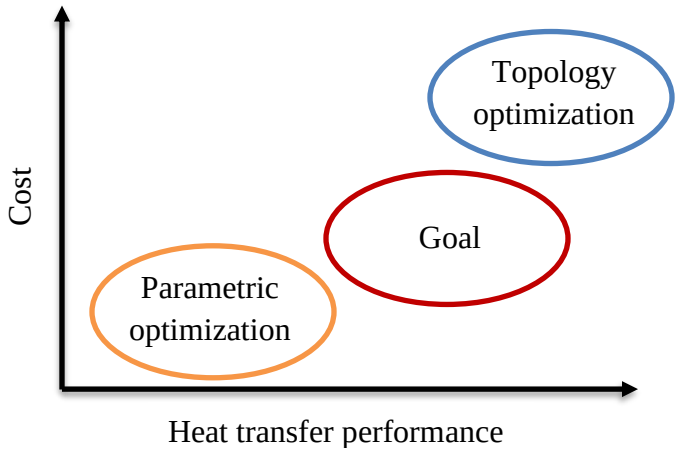

Figure 5.1: Cost to heat transfer performance of optimization methods.

Chapter III applied an evolutionary algorithm to optimize coefficients in a sensitivity equation. Material was then added in a generative design method based on this sensitivity equation. Analysis was then performed in a simplified simulation designed to test the effectiveness of this method versus an optimized parallel fin heatsink. Results showed after only a few generations, the evolutionary algorithm found a solution that increased heat transfer compared to the parallel fin heatsink. While this method works well in a simplified simulation, the number of simulations required for the evolutionary algorithm to train would make this difficult when applying CFD.

Chapter IV applied a neural network to optimize coefficients in a sensitivity equation. A low resolution 3-D CFD simulation was used to simulate the heatsinks. Initially, the mesh resolution tested was too low to effectively optimize a heatsink. After experimental testing, adjustments were made, and the resolution corrected. The mesh resolution was still low enough to simulate heatsinks quickly for the NN, while being refined enough to capture the relevant physics. Experimental and computational testing showed this process, in combination with additive manufacturing, is effective at both increasing heat transfer performance and minimizing weight.

5.2 Overall Conclusions

The results of this study show two novel optimization methods that can increase heat transfer performance over standard heatsink designs. While these methods may not be ideal for every application, they have certainly shown new techniques which can be applied to all types of problems. The evolutionary algorithm showed promising results for a simplified simulation but is too computationally intensive to be interfaced with CFD. When switching to a neural network, the optimization process converged considerably quicker than with the evolutionary algorithm making it a good choice for computationally intensive projects. Training the neural network on CFD simulations to optimize the material layout of a heatsink showed a capable method of

quickly increasing heat transfer performance or decreasing weight. Since this method does not derive the sensitivity equation from governing equations like many topology optimization problems, it can be applied to any problem that can be simulated. This could be very beneficial to multi-phase flow or heat transfer optimization problems where a sensitivity equation could be difficult to derive but the physics can be simulated.

5.3 Future Work

The upcoming work performed directly on this project will be experimentally validating the final heatsinks optimized in Chapter IV. These heatsinks are currently in the manufacturing process and will be tested afterwards. While experimental data verified our simulation results, it is still important to show the optimization process from start to finish, from initial simulation and NN training to real world results.

There are many possible directions to take this research next. As mentioned above, these methods could be very useful when applied to multi-phase heatsink optimization. Another path from this research would be to train a sensitivity equation over a wide range of heatsinks. For this study, the sensitivity equation coefficients were optimized directly for each heatsink. This means for each new heatsink optimization the neural network would need to be retrained. If the sensitivity equation coefficients were trained on a large number of different heatsink setups in different physics settings, a universal sensitivity analysis could be developed. This would allow for heatsink optimization to be considerably quicker.

REFERENCES

[1] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* (Natural Computing Series). Springer, Berlin, Heidelberg, 2015.

[2] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computing,* vol. 3, no. 1, pp. 1-16, 1995, doi: 10.1162/evco.1995.3.1.1.

[3] K. Foli, T. Okabe, M. Olhofer, Y. Jin, and B. Sendhoff, "Optimization of micro heat exchanger: CFD, analytical approach and multi-objective evolutionary algorithms," *International Journal of Heat and Mass Transfer,* vol. 49, no. 5-6, pp. 1090-1099, 2006, doi: 10.1016/j.ijheatmasstransfer.2005.08.032.

[4] A. Husain and K.-Y. Kim, "Multiobjective Optimization of a Microchannel Heat Sink Using Evolutionary Algorithm," *Journal of Heat Transfer,* vol. 130, no. 11, 2008, doi: 10.1115/1.2969261.

[5] S. Bureerat and S. Srisomporn, "Optimum plate-fin heat sinks by using a multiobjective evolutionary algorithm," *Engineering Optimization,* vol. 42, no. 4, pp. 305-323, 2010, doi: 10.1080/03052150903143935.

[6] S. Kanyakam and S. Bureerat, "Multiobjective Evolutionary Optimization of Splayed Pin-Fin Heat Sink," *Engineering Applications of Computational Fluid Mechanics,* vol. 5, no. 4, pp. 553-565, 2011, doi: 10.1080/19942060.2011.11015394.

[7] S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks,* vol. 1, no. 1, pp. 17-61, 1988, doi: 10.1016/0893-6080(88)90021-4.

[8] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology,* 2017, doi: 10.1109/ICEngTechnol.2017.8308186.

[9] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *2017 International Conference on Communication and Signal Processing*, 2017, doi: 10.1109/ICCSP.2017.8286426.

[10] M. Y. Rafiq, G. Bugmann, and D. J. Easterbrook, "Neural Network Design for Engineering Applications," *Computers & Structures*, vol. 79, no. 17, pp. 1541-1552, 2001, doi: 10.1016/S0045-7949(01)00039-6.

[11] J. Kutz, "Deep Learning in FLuid Dynamics," *Journal of Fluid Mechanics*, vol. 814, pp. 1-4, 2017, doi: 10.1017/jfm.2016.803.

[12] O. San and R. Maulik, "Extreme learning machine for reduced order modeling of turbulent geophysical flows," *Physical Review*, vol. 97, no. 4, 2018, doi: 10.1103/PhysRevE.97.042322.

[13] Y. Kuan and H. Lien, " The Integration of the Neural Network and Computational Fluid Dynamics for the Heatsink Design," in *Advances in Neural Networks. Lecture Notes in Computer Science.*, vol. 3498, W. J., L. XF., and Y. Z. Eds.: Springer, Berlin, Heidelberg, 2005.

[14] Y. Hsueh, H. Lien, and M. Hsueh, "An Intelligent System for the Heatsink Design," in *Advances in Neural Networks. Lecture Notes in Computer Science.*, vol. 3973, W. J., Y. Z., Z. J.M., L. BL., and Y. H. Eds.: Springer, Berlin, Heidelberg, 2006.

[15] Y. Islamoglu, "A new approach for the prediction of the heat transfer rate of the wire-on-tube type heat exchanger—use of an artificial neural network model," *Applied Thermal Engineering*, vol. 23, no. 2, pp. 243-249, 2003, doi: 10.1016/S1359-4311(02)00155-2.

[16] G. N. Xie, Q. W. Wang, M. Zeng, and L. Q. Luo, "Heat transfer analysis for shell-and-tube heat exchangers with experimental data by artificial neural networks approach," *Applied Thermal Engineering*, vol. 27, no. 5-6, pp. 1096-1104, 2007, doi: 10.1016/j.applthermaleng.2006.07.036.

[17] D. Copeland, "Optimization of parallel plate heatsinks for forced convection," *IEEE*, vol. Sixteenth Annual IEEE Semiconductor Thermal Measurement and Management Symposium, pp. 266-272, 22-23 March 2000 2000, doi: 10.1109/STHERM.2000.837093.

[18] W. B. Krueger and A. Bar-Cohen, "Optimal numerical design of forced convection heat sinks," *IEEE*, vol. 27, no. 2, pp. 417 - 425, 28 June 2004 2004, doi: 10.1109/TCAPT.2004.830969.

[19] J. Lee, K. Rew, C. Lee, D. Kim, and M. Kim, "A study on heatsink fin wave optimization for climate control seat," *Journal of Mechanical Science and Technology,* vol. 31, pp. 1489-1495, 23 March 2017 2016, doi: 10.1007/s12206-017-0248-4.

[20] W. Tong, Z. Wang, B. Ozpineci, M. Chinthavali, and S. Campbell, "Automated Heatsink Optimization for Air-Cooled Power Semiconductor Modules," *IEEE Transactions on Power Electronics,* vol. 34, no. 6, pp. 5027-5031, June 2019 2019, doi: 10.1109/TPEL.2018.2881454.

[21] B. Vaissier, J.-P. Pernot, L. Chougrani, and P. Véron, "Parametric Design of Graded Truss Lattice Structures for Enhanced Thermal Dissipation," *Computer-Aided Design,* vol. 115, pp. 1-12, 2019, doi: 10.1016/j.cad.2019.05.022.

[22] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Applied Mechanics and Engineering,* vol. 71, pp. 197-224, 1988.

[23] A. A. Koga, E. C. C. Lopes, H. F. Villa Nova, C. R. de Lima, and E. C. Nelli Silva, "Development of heat sink device by using topology optimization," *International Journal of Heat and Mass Transfer,* vol. 64, pp. 759-772, 6/4/2013 2013.

[24] K. Yaji, T. Yamada, S. Kubo, K. Izui, and S. Nishiwaki, "A topology optimization method for a coupled thermal–fluid problemusing level set boundary expressions," *International Journal of Heat and Mass Transfer,* vol. 81, pp. 878-888, 2015, doi: 10.1016/j.ijheatmasstransfer.2014.11.005.

[25] C. Saglietti, P. Schlatter, E. Wadbro, M. Berggren, and D. S. Henningson, "Topology optimization of heat sinks in a square differentially heated cavity," *International Journal of Heat and Fluid Flow,* vol. 74, pp. 36-52, 2018, doi: 10.1016/j.ijheatfluidflow.2018.08.004.

[26] T. Lei *et al.*, "Investment casting and experimental testing of heat sinks designed by topology optimization," *International Journal of Heat and Mass Transfer,* vol. 127, B, pp. 396-412, 2018, doi: 10.1016/j.ijheatmasstransfer.2018.07.060.

[27] E. M. Dede, S. N. Joshi, and F. Zhou, "Topology Optimization, Additive Layer Manufacturing, and Experimental Testing of an Air-Cooled Heat Sink," *Journal of Mechanical Design,* vol. 137, no. 11, October 12, 2015 2015, doi: 10.1115/1.4030989.

[28] A.-C. Iradukunda, A. Vargas, D. Huitink, and D. Lohan, "Transient thermal performance using phase change material integrated topology optimized heat sinks," *Applied Thermal Engineering,* vol. 179, July 2020 2020, doi: 10.1016/j.applthermaleng.2020.115723.

[29] L. C. Wei, L. E. Ehrlich, M. J. Powell-Palm, C. Montgomery, J. Beuth, and J. A. Malen, "Thermal conductivity of metal powders for powder bed additive manufacturing," *Additive Manufacturing*, vol. 21, pp. 201-208, 2018, doi: 10.1016/j.addma.2018.02.002.

[30] T. Wu, A. Wereszczak, H. Wang, B. Ozpineci, and C. Ayers, "Thermal response of additive manufactured Aluminum," *2016 International Symposium on 3D Power Electronics Integration and Manufacturing (3D-PEIM)*, pp. 1-15, 2016, doi: 10.1109/3DPEIM.2016.7570537.

[31] R. Tucker, M. Khatamifar, W. Lin, and K. McDonald, "Experimental investigation of orientation and geometry effect on additive manufactured aluminium LED heat sinks under natural convection," *Thermal Science and Engineering Progress*, vol. 23, 2021, doi: 10.1016/j.tsep.2021.100918.

[32] R. Kempers, J. Colenbrander, W. Tan, R. Chen, and A. J. Robinson, "Experimental characterization of a hybrid impinging microjet-microchannel heat sink fabricated using high-volume metal additive manufacturing," *International Journal of Thermofluids*, vol. 5-6, 2020, doi: 10.1016/j.ijft.2020.100029.

[33] M. Chinthavali, C. Ayers, S. Campbell, R. Wiles, and B. Ozpineci, "A 10-kW SiC inverter with a novel printed metal power module with integrated cooling using additive manufacturing," *2014 IEEE Workshop on Wide Bandgap Power Devices and Applications*, pp. 48-54, 2014, doi: 10.1109/WiPDA.2014.6964622.

[34] R. Steinbuch, "Successful Application of Evolutionary Algorithms in Engineering Design," *Journal of Bionic Engineering*, vol. 7, pp. S199-S211, 10/13/2010 2010, doi: S1672-6529(09)60236-5.

[35] A. Husain and K. Kim, "Design optimization of manifold microchannel heat sink through evolutionary algorithm coupled with surrogate model," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 3, no. 4, pp. 617-624, 2013, doi: 10.1109/TCPMT.2013.2245943.

[36] R. Bornoff and J. Parry, "An additive design heatsink geometry topology identification and optimization algorithm," *IEEE*, pp. 303-308, 4 May 2015 2015, doi: 10.1109/SEMI-THERM.2015.7100177.

[37] H. Khawaja and M. Moatamedi, "Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) – solution in MATLAB®," *The International Journal of Multiphysics*, vol. 12, 4, pp. 313-326, 2018.

[38] *Simcenter STAR-CCM+*. (2021).

[39] K. A. Hoffman and S. T. Chiang, "Computational Fluid Dynamics Volume 1," Engineering Education System, 2000, ch. 3, pp. 124-132.

[40] J. D. Hunter, "Matplotlib: a 2D graphics environment", *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007, DOI: 10.1109/MCSE.2007.55.

VITA

Daniel Carne

Candidate for the Degree of

Master of Science

Thesis: DESIGN OPTIMIZATION METHODS FOR ADDITIVELY MANUFACTURED NATURAL CONVECTION HEATSINKS

Major Field: Mechanical and Aerospace Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in your Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2021.

Completed the requirements for the Bachelor of Science in your Mechanical Engineering Technology at Oklahoma State University, Stillwater, Oklahoma in December, 2019.

Experience:
Graduate Research Assistant / Teaching Assistant, Oklahoma State University, Stillwater, OK, Jan. 2018 – Dec. 2021.

Engineering Intern, Peterbilt Motors, Denton TX, May – Aug. 2019.