

HEAT TRANSFER MODELING AND OPTIMIZATION OF A CARBONIZED  
MICROVASCULAR SOLAR RECEIVER

by

Taylor Brown



A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

Boise State University

December 2020

© 2020

Taylor Brown

ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

**DEFENSE COMMITTEE AND FINAL READING APPROVALS**

of the thesis submitted by

Taylor Brown

Thesis Title: Heat Transfer Modeling and Optimization of a Carbonized Microvascular Solar Receiver

Date of Final Oral Examination: 01 December 2020

The following individuals read and discussed the thesis submitted by student Taylor Brown, and they evaluated their presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Todd Otanicar, Ph.D. Chair, Supervisory Committee

Eric Jankowski, Ph.D. Member, Supervisory Committee

Krishna Pakala, Ph.D. Member, Supervisory Committee

The final reading approval of the thesis was granted by Todd Otanicar, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

## ACKNOWLEDGMENTS

I would like to thank Dr. Todd Otanicar for advising and mentoring me for the past two and half years. His guidance has inspired my interest in renewable energies and his encouragement has allowed me to apply my technical interests to this field. I would also like to thank my fellow lab members, in particular Matt Zuzelski for his CFD modeling work and Chase Christen for his help in the initial development of the optimization work.

This work would not be possible without the other members of the research project at The University of Tulsa, and the direct help received from Dr. Michael Keller and Dr. Daniel Crunkleton. Also, special thanks to fellow graduate student Jonathan Argot for his work on stress calculations.

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number EE0008736.

## ABSTRACT

Concentrating solar power is an emerging renewable energy source. The technology can collect and store thermal energy from the sun over long durations, generating electricity as needed at a later time. Current CSP systems are limited to a maximum operational temperature due to constraints of the working fluid, which limits the maximum possible efficiency of the system. One proposed pathway forward is to utilize a gas phase for the working fluid in the system such as supercritical carbon dioxide.

A composite gas phase modular receiver is being developed by researchers at Boise State University and the University of Tulsa. The receiver uses supercritical carbon dioxide as the working fluid, which can operate at temperatures greater than 1000 °C. The unique carbon-carbon composite material has high thermal conductivity and is structurally durable at extreme temperatures.

A model has been developed in this work to simulate the thermal and hydraulic performance of a composite receiver unit cell. The model is built as a thermal resistance network that solves more quickly than traditional computational fluid dynamics simulations. The thermal and hydraulic models are compared with CFD simulations and show close agreement over a wide range of inlet velocities and path architectures.

A genetic algorithm has been developed to optimize the design of the receiver. The algorithm optimizes the fluid channel diameter, inlet velocity, and the path architecture design of a unit cell. The optimization scheme weighs the thermal

performance of the receiver with the hydraulic performance, maximizing the thermal efficiency and minimizing the pressure drop. The nominal strain is also calculated and constrained. The algorithm produces an optimal design from a constrained set of architectures. The optimal design is a simple three-channel parallel path with an acceptable pressure drop, less than 17 kPa. The thermal efficiency of the design is 75.6% with a  $1,000,000 \text{ W/m}^2$  solar flux and the nominal strain is an allowable 0.03%. Future work will be done to expand the path design space and remove arbitrary constraints from the optimization process.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS .....	iv
ABSTRACT .....	v
LIST OF TABLES .....	x
LIST OF FIGURES .....	xi
LIST OF ABBREVIATIONS .....	xiii
CHAPTER ONE: INTRODUCTION .....	1
1.1 Background.....	1
1.2 Modular sCO <sub>2</sub> Receiver Modeling.....	6
1.3 Receiver Design Optimization.....	9
1.4 Research Goal and Objective.....	11
CHAPTER TWO: THERMAL MODEL .....	13
2.1 Thermal Model Methodology.....	13
2.1.1 Resistance Network .....	13
2.1.2 Surface Boundary .....	14
2.1.3 Fluid Boundary .....	16
2.1.4 System of Equations .....	20
3.1.5 Solving the System of Equations.....	20
2.2 Thermal Model Comparison with CFD.....	22
2.2.1 CFD Model.....	22

2.2.2 CFD Results Comparison.....	25
CHAPTER THREE: PRESSURE MODEL .....	34
3.1 Pressure Model Methodology .....	34
3.1.1 Major Losses .....	34
3.1.2 Minor Losses .....	35
3.1.3 Mass Flow Distribution.....	37
3.2 Pressure Model Comparison with CFD.....	39
3.2.1 CFD Results Comparison.....	39
3.2.2 Cross Analysis .....	46
CHAPTER FOUR: OPTIMIZATION.....	48
4.1 Genetic Algorithm Methodology .....	49
4.1.1 Design Variables.....	49
4.1.2 Seeding.....	52
4.1.3 Fitness Function.....	53
4.1.4 Activation Functions .....	53
4.1.5 Strain Constraint .....	55
4.1.6 Parent Selection .....	56
4.1.7 Crossover.....	57
4.1.8 Mutation .....	58
4.1.9 Termination .....	59
4.2 Optimization Results .....	60
4.2.1 Design Space .....	60
4.2.2 Design Variable Bounds .....	61

4.2.3 Algorithm Parameters .....	62
4.2.4 Seeding Analysis .....	63
4.2.5 Activation Function Analysis .....	66
4.2.6 Optimal Design.....	69
4.2.7 Strain and Pressure Constraints .....	78
4.2.8 Optimization Results Discussion.....	79
CHAPTER FIVE: CONCLUSION .....	82
5.1 Conclusion .....	82
5.2 Future Work and Recommendations.....	83
REFERENCES.....	85
APPENDIX A .....	89

## LIST OF TABLES

Table 2.1	Mesh Sizing.....	23
Table 2.2	Model Parameters.....	26
Table 4.1	Optimization Model Parameters.....	61
Table 4.2	Design Variable Bounds .....	62
Table 4.3	Optimal Case Design Variables .....	75
Table 4.4	Optimal Case Results.....	75
Table 4.5	Optimal Thermal Efficiency Case Design Variables .....	76
Table 4.6	Optimal Thermal Efficiency Case Results.....	76
Table 4.7	Custom High Efficiency Case Results.....	77
Table A.1	sCO <sub>2</sub> Properties at 873 K and 20 MPa .....	91

## LIST OF FIGURES

Figure 1.1	Power Tower from [4] .....	2
Figure 1.2	Schematic of Tubular External Receiver from [7] .....	4
Figure 1.3	Brayton Energy Plate-Fin sCO <sub>2</sub> Receiver Design from [16] .....	5
Figure 1.4	Numbering Up Central Receiver from [10] .....	6
Figure 1.5	Two-Dimensional Thermal Resistance Network from [10] .....	7
Figure 1.6	Three-Dimensional Resistance Network from [12] .....	8
Figure 1.7	Branching Flow Design Optimization from [14] a) Reference Design b) Optimal Design c) Initial Temperature Contour d) Optimal Temperature Contour .....	10
Figure 2.1	Top Down View of the Resistance Network .....	14
Figure 2.2	Receiver Surface Resistance Diagram.....	15
Figure 2.3	Fluid Boundary Resistance Diagram.....	16
Figure 2.4	Simulation Method Solve Time Comparison .....	21
Figure 2.5	Path Types a) Straight b) Serial and c) Parallel .....	23
Figure 2.6	Mesh Section View a) Straight b) Serial and c) Parallel .....	24
Figure 2.7	Outlet Temperature CFD Comparison for a) Straight b) Serial and c) Parallel Designs.....	28
Figure 2.8	Outlet Temperature CFD Error for a) Straight b) Serial and c) Parallel Designs .....	30
Figure 2.9	Outlet Temperature CFD Percent Error for a) Straight b) Serial and c) Parallel Designs.....	32
Figure 3.1	Path Architecture Loop Example .....	38

Figure 3.2	Pressure Drop CFD Comparison for a) Straight b) Serial and c) Parallel Designs .....	41
Figure 3.3	Pressure Drop CFD Error for a) Straight b) Serial and c) Parallel Designs .....	43
Figure 3.4	Pressure Drop CFD Percent Error for a) Straight b) Serial and c) Parallel Designs .....	45
Figure 3.5	Cross Pressure Drop Comparison .....	46
Figure 4.1	Path Types a) Serial b) Parallel and c) Bifurcating .....	50
Figure 4.2	Path Length Similarity for a) Serial b) Parallel and c) Bifurcating .....	51
Figure 4.3	Sigmoid Activation Function .....	54
Figure 4.4	Fitness and Design Variable Evolution for Various Seeding Types .....	65
Figure 4.5	Fitness and Design Variable Evolution for Various Activation Functions .....	68
Figure 4.6	Optimal Case Design Variable Evolution of a) Velocity b) Path Type c) Path Value and d) Diameter .....	71
Figure 4.7	Model Performance Evolution for the Optimal Case .....	72
Figure 4.8	Optimal Case Pathway Design .....	74
Figure 4.9	Custom High Efficiency Path Design .....	77
Figure 4.10	Surface Temperatures for a) Optimal Design b) Optimal Thermal Efficiency Design and c) Custom High Efficiency Design .....	78
Figure 4.11	Branching Flow Design Optimization from [14] a) Reference Design b) Optimal Design c) Initial Temperature Contour d) Optimal Temperature Contour .....	80

## LIST OF ABBREVIATIONS

BiCGStab	Biconjugate Gradient Stabilized Method
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CSP	Concentrating Solar Power
GPU	Graphics Processing Unit
sCO <sub>2</sub>	Supercritical Carbon Dioxide
UDF	User Defined Function

## CHAPTER ONE: INTRODUCTION

### 1.1 Background

The Earth's climate crisis is driving researchers to develop and optimize alternative forms of energy. Renewable energy sources, which do not include nuclear, account for only 17.6% of the United States' electricity supply. Fossil fuels, in contrast, directly contribute to climate change and account for 62.6% of the electricity generated [1]. The National Renewable Energy Laboratory (NREL) predicted that 80% of the United States' electricity demand could be supplied by renewable sources by 2050 with current technologies [2].

Concentrating solar power (CSP) is a form of renewable energy that harnesses thermal energy from the sun and converts it into electricity. The main benefit of CSP, as compared to photovoltaic cells, is its ability to cheaply and efficiently store energy collected from the sun thermally. Thermal energy can be stored in a working fluid more cost effectively than electrical energy and does not need to be immediately converted to electricity. This is especially beneficial for solar energy because energy can be collected during the day and utilized at night.

CSP is an emerging technology with expected increased use in the US and around the world. In 2019, CSP accounted for only 0.1% of the electricity generation in the US [1]. The Department of Energy's Solar Energy Technologies Office aims to make CSP cost competitive with other dispatchable systems by 2030, with a target cost of \$0.05-\$0.10 per kWh [3].

There are multiple forms of concentrating solar power. Parabolic Trough and Linear Fresnel collectors are low power (100 suns) collectors that focus solar energy on a “line” receiver. The main benefit of these designs is the relative low cost and maintainability. Stirling dish collectors focus energy on a single point with high efficiency but are not integrated with thermal energy storage and have low relative power output [4].

Solar power towers are the focus of this work. Figure 1.1 shows an image of a solar power tower. A central receiver is surrounded by heliostats - mirrors that direct the sun’s energy to a single point. Power towers operate at high temperatures that can be achieved by the working fluid given the high solar flux created by the heliostats. The higher temperatures result in a higher Carnot efficiency, increasing the maximum possible efficiency of the system.



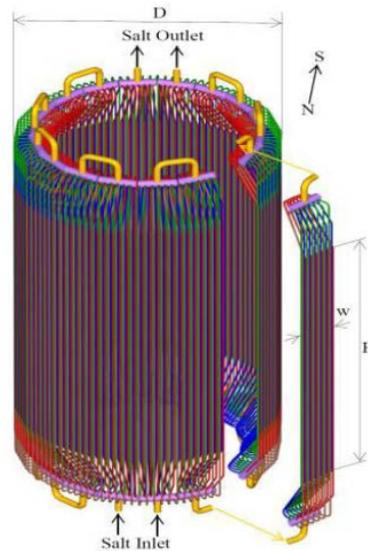
**Figure 1.1 Power Tower from [4]**

In a 2018 review by Islam et al. [5], the total gross capacity of installed solar power tower systems was over 618 MW worldwide, with 6 plants under construction.

The focus of future research for solar power towers is making the systems more efficient and therefore, more cost effective. This work focuses on improving the design of the central receiver unit.

The central receiver in a concentrating solar power system is a source of improvement for efficiency and cost. Receivers transfer the solar energy directed by heliostats to a working fluid as thermal energy. The efficiency of the receiver describes the amount of energy the working fluid absorbs, relative to the amount of energy directed to the receiver surface. Receivers with higher efficiency, therefore, will result in more energy absorbed by the fluid and eventually converted to electricity. Along with the thermal efficiency, receivers can be improved by decreasing the material cost and cost of operation. Operation costs include the cost of pumping the working fluid, which is affected by the fluid pressure drop.

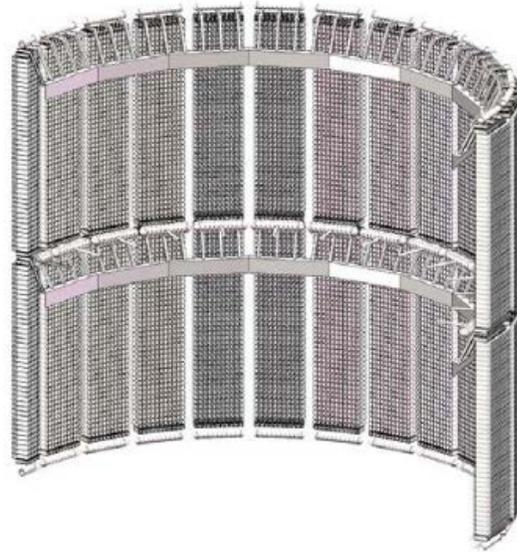
The current state-of-the-art power towers use molten salt external tubular receivers [6]. The fluid flows through tubes on the exterior of the receiver and is indirectly heated by the heliostat field. A schematic of a molten salt external tubular receiver design from [7] is shown in Figure 1.2. Molten salts are beneficial because they can be used as the thermal storage medium as well as the working fluid but are currently limited to a maximum operational temperature near 600 °C [8]. In addition, molten salts are corrosive, requiring special receiver material selection.



**Figure 1.2 Schematic of Tubular External Receiver from [7]**

Supercritical carbon dioxide ( $s\text{CO}_2$ ) is a promising working fluid alternative to molten salts. The fluid can operate at temperatures above  $1000\text{ }^\circ\text{C}$ , increasing the maximum possible efficiency of the system. The system components for  $s\text{CO}_2$  can also be more compact and less costly, given the relatively low density of the fluid [9]. High pressures are necessary, however, to maintain the supercritical phase, which requires special mechanical considerations for the receiver. Tubular external receivers are possible with  $s\text{CO}_2$  but would require small diameters to maintain structural integrity, which would increase the overall pressure drop [8].

Several receiver concepts have been developed to utilize  $s\text{CO}_2$ 's unique properties and strengths as a working fluid [6]. Brayton Energy used a plate-fin heat exchanger style to form a receiver capable of withstanding the high operating pressure and temperatures of  $s\text{CO}_2$ . The plate-fin indirectly heats the flowing  $s\text{CO}_2$ , which absorbs sufficient energy to limit the maximum temperature on the receiver surface. The total receiver is a collection of modular units connected in parallel, as shown in Figure 1.3.

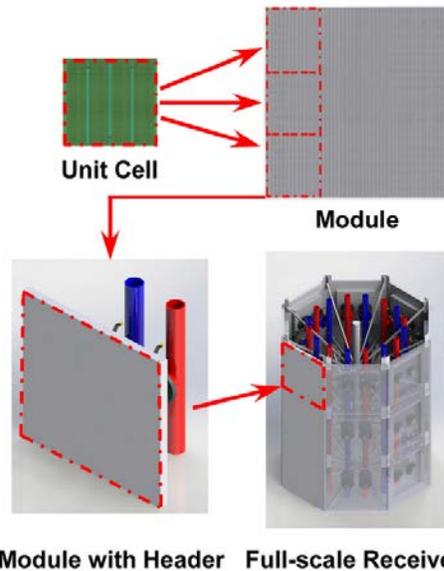


**Figure 1.3 Brayton Energy Plate-Fin  $s\text{CO}_2$  Receiver Design from [16]**

NREL developed a cellular cavity receiver that aimed to reduce reflective losses by directing solar energy into cavities [6]. Rays that are initially reflected are directed deeper into the cavity and eventually absorbed. This causes a decrease in the peak solar flux on any receiver surface location as the energy is spread through the receiver. Similar to Brayton Energy's design, the receiver is composed of unit cells connected in parallel. Certain mechanical issues arose from this design; headers in direct irradiation needed to be protected with an actively cooled reflector and certain tubes buckled under their own weight. Also, the ideal receiver material would have a unique balance of absorptivity and reflectivity properties which required a special coating.

Researchers at Oregon State University [10] developed a modular central receiver that uses supercritical carbon dioxide as the working fluid. The high pressure required to maintain the supercritical phase is possible given the unit cell-based design of the receiver. The receiver is a collection of small pin-fin metallic unit cells that are connected in parallel in a numbering up approach. A schematic of the design from [10] is shown in

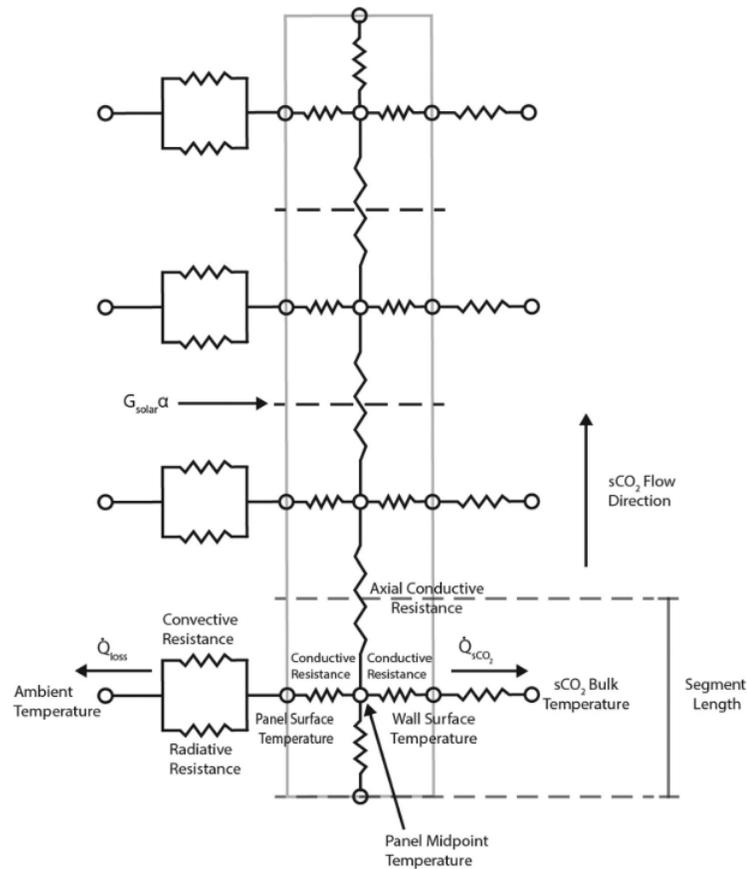
Figure 1.4. The receiver design supported sCO<sub>2</sub> as the working fluid, but the machinability of the material limited the design space of the working fluid path and limited the maximum possible temperature of the receiver.



**Figure 1.4** Numbering Up Central Receiver from [10]

## 1.2 Modular sCO<sub>2</sub> Receiver Modeling

Oregon State modeled the receiver design with a two-dimensional thermal resistance network and validated it with experimental testing. A thermal resistance network breaks down the receiver design into temperature nodes that are connected with resistors. The resistors control the flow of energy between nodes and vary based on the connection and material properties. The diagram in Figure 1.5 from [10] shows the two-dimensional resistance network.



**Figure 1.5 Two-Dimensional Thermal Resistance Network from [10]**

The diagram shows how the external heat flux, convection, and radiation (left) interact with the receiver surface and pass energy to the  $s\text{CO}_2$  working fluid (right). Thermal resistance networks are a simplified alternative to more complex computational fluid dynamics simulations, saving time and computational resources. Oregon State's thermal model was compared with experimental results and an uncertainty analysis predicted a maximum thermal efficiency uncertainty of 0.285% near the design conditions. The corresponding hydraulic model was not validated with experimental data and the average pressure drop uncertainty was 23.1% [11].

Besarati et al. [12] used a similar three-dimensional thermal resistance model of a metallic microchannel receiver. The resistance network from [12] is shown in Figure 1.6.

Besarati's model simulated sCO<sub>2</sub> heated with a solar flux of 500 kW/m<sup>2</sup> rising from 530 °C to 700 °C while limiting the surface temperature to below the allowed material constraint.

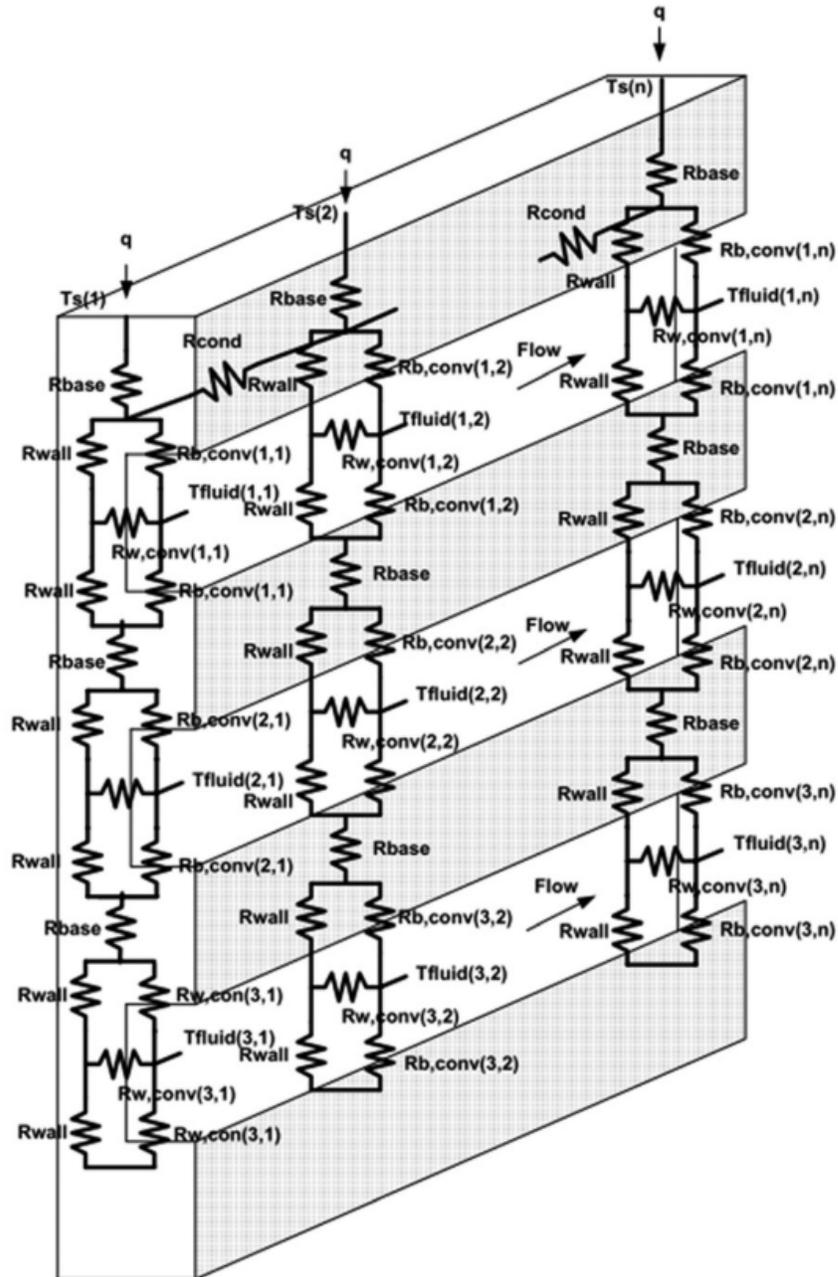
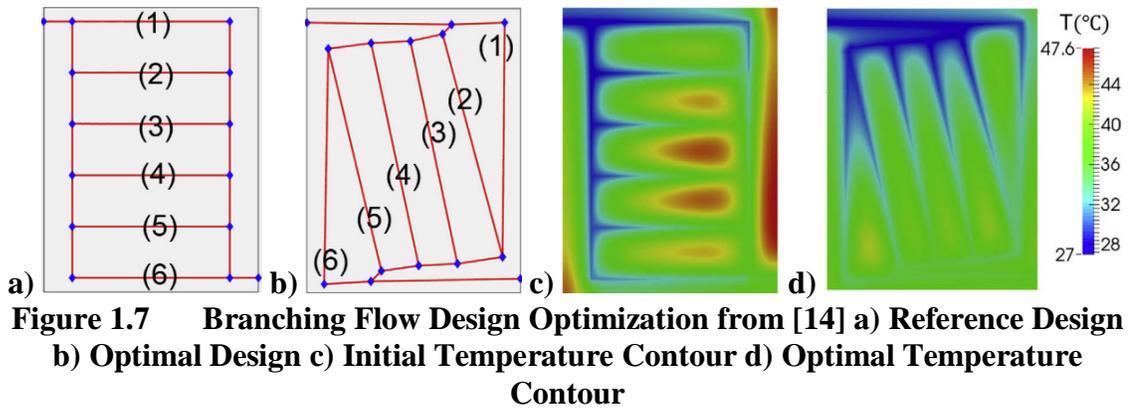


Figure 1.6 Three-Dimensional Resistance Network from [12]

The resistance network used by Besarati et al. was developed by Lei [13] for small channel multilayer heat sinks. The uncertainty of the thermal resistance model and pressure drop model were determined to be less than 8% and 6% respectively, when compared with CFD simulations. Experimental tests showed close agreement with the thermal and pressure models. The thermal model consistently underestimated the surface temperature but with a max error of only 5 °C. The pressure model was similarly accurate, with a maximum reported error of less than 12%, which correlates to a pressure difference of 157 Pa.

### **1.3 Receiver Design Optimization**

Past research has been done on optimizing the design of heat transfer devices, such as receivers and heat exchangers, and microvascular flow networks. Tan et al. [14] optimized the thermal and hydraulic performance of a microvascular composite battery packaging design. A combination of computational fluid dynamics (CFD) and finite element models (FEM) were used to analyze the temperatures on the surface of the composites and the pressure drop throughout the microchannels. A gradient-based optimization scheme was used to limit the surface temperatures and drive towards temperature uniformity on the surface. The optimization started with a set of simple reference designs that were significantly modified to form the optimal solutions. Figure 1.7 shows the initial and optimal designs of a six-channel parallel network from [14].



Micro heat exchangers were studied by Abdoli and Dulikravich [15] with the purpose of optimizing their design with a multi-objective genetic algorithm. The design was a branching four floor counterflow exchanger. The algorithm maximized the heat removal capacity and minimized the needed pumping power for the coolant and the temperature gradient on the surface. The optimal design resulted in a 13% better heat removal percentage than a non-optimal design.

Aragón et al. [16] studied self-healing microvascular flow networks. A multi-objective genetic algorithm was used to optimize the networks for flow efficiency and void volume fraction, and to ensure flow in every microchannel. In addition, the network's self-healing design required full coverage and channel redundancy. The results found optimal designs increase the channel diameter between the source and target locations, with smaller diameters elsewhere to maintain coverage.

A 3D printed liquid cooled heat sink was studied by Dokken and Fronk [17]. The heat sink was discretized into an array of bits that represented the additive manufacturing process. CFD was used to calculate the thermal and hydraulic performance of the designs and the two metrics were unified by calculating the entropy generation. A micro-genetic algorithm which reduced the number of necessary CFD simulations was used to minimize

the entropy generation of the heat sink. The algorithm reduced the entropy generation by up to 26.4% compared to non-optimized designs.

#### **1.4 Research Goal and Objective**

A novel carbon-carbon composite receiver with sCO<sub>2</sub> working fluid is being developed by researchers at the University of Tulsa and Boise State University that aims to increase the thermal efficiency and reduce the overall cost of current CSP receivers. The unique structure of the composite allows for complex architectures not machinable in traditional materials, i.e. metals. This expanded design space allows for microvascular receiver designs that maximize thermal efficiency and minimize pressure drop.

The full composite receiver will consist of an array of small unit cells, the specific size of which is still being determined, but will likely be in the range of square centimeters. Each unit cell will have a microvascular pathway for supercritical carbon dioxide to pass through and absorb thermal energy. These unit cells will be attached in parallel, in a numbering up approach, similar to the work of Oregon State [10].

Carbon-carbon composites have unique properties that make them promising materials for solar receivers. They are currently used in the aerospace and aeronautics industry as brakes, nose tips, and nozzles because of their low density and ability to operate in extreme temperatures [18]. The composites are thermally resistant, with low thermal expansion that maintains mechanical strength at high temperatures. The composites also conduct heat well, with high thermal conductivity that is ideally suited for heat exchangers and thermal receivers [19]. Additive manufacturing techniques, such as 3D printing, allow the material to be constructed with high precision, making complex

microvascular designs possible. This complex design space will allow for better receiver designs, and better overall performance.

A three-dimensional thermal resistance model has been developed to dynamically model the thermal and hydraulic performance of the composite receiver with any microvascular path architecture. Model parameters, such as solar flux and receiver dimensions, can also be altered and studied. The thermal resistance model is discussed in detail in Chapter 2 and the hydraulic model is detailed in Chapter 3.

The dynamic nature of the thermal and hydraulic models allows the receiver to be optimized with an automated process that designs the internal microvascular network. The purpose of the optimization is to maximize the efficiency of the receiver and limit the pressure drop. The nominal strain of the receiver is also calculated and constrained. A single objective genetic algorithm has been developed to optimize the design of the receiver. The genetic algorithm and resulting optimal designs are discussed in Chapter 4.

## CHAPTER TWO: THERMAL MODEL

The thermal model was developed with C# to simulate the heat transfer of a receiver with any path architecture. Also, a wide variety of model parameters are included as inputs, such as inlet velocity, inlet temperature, and solar flux. The purpose was to calculate the overall thermal efficiency more quickly than with CFD and to provide information about how the heat is distributed in the system.

The receiver has two main boundaries, the surface and the fluid. The surface interacts with the environment via radiation and convection and passes heat to the fluid. The fluid absorbs heat via convection and transfers it out of the receiver. The thermal model simulates this process at steady state.

Section 2.1 describes the thermal model, which is a resistance network. Section 2.2 discusses how this model has been validated.

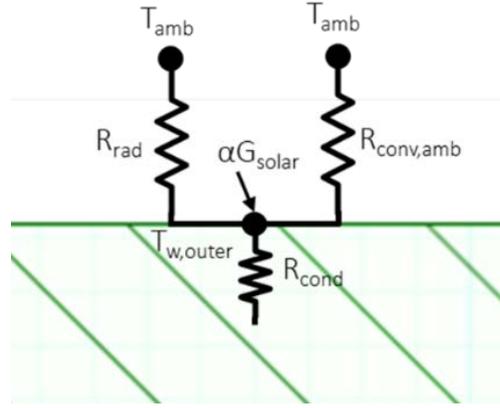
### **2.1 Thermal Model Methodology**

#### 2.1.1 Resistance Network

The receiver's thermal model is built as a resistance network, with the receiver divided into nodes that are connected by thermal resistance values. Each node represents the temperature at a certain point in the receiver and the resistance value connecting them dictates how energy passes between them. Figure 2.1 shows a simplified top down view of the receiver broken into temperature nodes.



heat flux from the heliostat field. The three heat transfer forms are shown in Figure 2.2 and Eq. (2.3) describes the energy balance at a receiver surface node.



**Figure 2.2 Receiver Surface Resistance Diagram**

$$0 = \sigma \varepsilon A_s T_s^4 - \sigma \alpha A_s T_{amb}^4 + h_{ext} A_s (T_s - T_{amb}) - \alpha G_{solar} + Q_{in} \quad (2.3)$$

where  $Q_{in}$  is the energy entering the surface,  $\sigma$  is the Stefan-Boltzmann constant,  $\varepsilon$  is emissivity,  $A_s$  is the surface area the node represents,  $T_s$  is the node temperature,  $T_{amb}$  is the ambient temperature,  $h_{ext}$  is the external convection coefficient,  $\alpha$  is the surface absorptivity, and  $G_{solar}$  is the solar flux on the receiver.

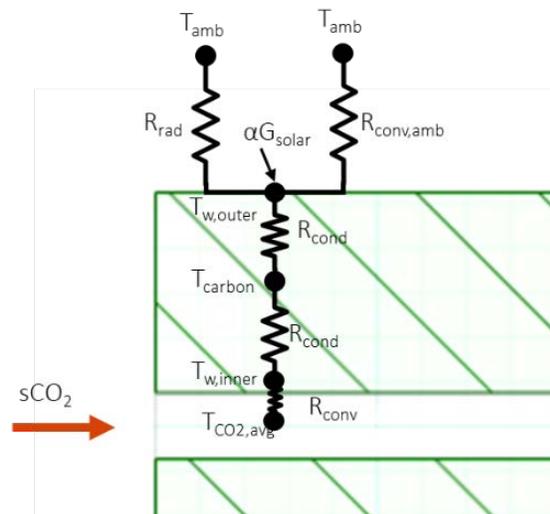
To solve the resistance network efficiently, each node was made to be linearly dependent on its surrounding nodes. This is the case for every resistance value except for the radiation exchanged with the surroundings, which involves exponential temperature terms. To account for this, the radiation exchange at the boundary is linearized with Eq. (2.4).

$$\frac{\sigma \varepsilon A_s T_s^4 - \sigma \alpha A_s T_{amb}^4}{(T_{s,assume} + T_{amb})(T_{s,assume}^2 + T_{amb}^2)} = (\sigma \varepsilon A_s T_s - \sigma \alpha A_s T_{amb}) \quad (2.4)$$

where  $T_{s,assume}$  is the assumed final temperature for each surface node. The model is solved iteratively until the assumed final temperature is within a certain tolerance of the outputted final temperature for every surface node.

### 2.1.3 Fluid Boundary

The fluid passing through the receiver is modeled as temperature nodes, connected to the receiver via convection resistances. This is shown in Figure 2.3.



**Figure 2.3 Fluid Boundary Resistance Diagram**

The convection resistances that connect the fluid nodes to the receiver are dependent on the convection heat transfer coefficient. This relationship is shown in Eq. (2.5).

$$R_{conv} = \frac{1}{h * A_f} \quad (2.5)$$

where  $h$  is the convection heat transfer coefficient and  $A_f$  is the surface area of the fluid channel represented by each node. The convection heat transfer coefficient is determined, in part, by the Nusselt number,  $Nu$ , as described in Eq. (2.6).

$$Nu = \frac{hD}{k} \quad (2.6)$$

where  $D$  is the diameter of the fluid pathway and  $k$  is the thermal conductivity.

The Nusselt number for internal flow is dependent on whether the flow is laminar or turbulent and whether the flow is developing or fully developed. Therefore, four correlations were used to calculate the Nusselt number for a fluid node depending on the flow state.

To determine if the flow is laminar or turbulent, the Reynolds number,  $Re$ , is calculated, as shown in Eq. (2.7).

$$Re = \frac{\rho V D}{\mu} \quad (2.7)$$

where  $\rho$  is the fluid density,  $V$  is the fluid velocity, and  $\mu$  is the fluid dynamic viscosity.

This work assumes that flow with a Reynolds number greater than 2,100 is turbulent, and all other flow is laminar.

A fluid node is developing or fully developed based on its distance from a disturbance (such as a bend or branch) and the fluid entry region length. The fluid entry length calculation is unique for laminar and turbulent flows. The laminar entry region length correlation is shown in Eq (2.8) and the turbulent entry region length correlation is shown in Eq (2.9) [20].

$$L_{h,laminar} = 0.06 * Re * D \quad (2.8)$$

$$L_{h,turbulent} = 404 * D * Re^{1/6} \quad (2.9)$$

To determine if a node is developing or fully developed, the distance from a disturbance is calculated and compared with the entry region length. If the distance is within the entry region length, the flow is developing, else it is fully developed.

Laminar, fully developed flow has a constant Nusselt number, shown in Eq. (2.10) [21]. The resistance network is constructed to assume a constant heat flux into

each fluid region that is represented by a node. Therefore, internal flow correlations were selected that assume a constant wall heat flux.

$$Nu_{developed,laminar} = 4.364 \quad (2.10)$$

Laminar, developing flow uses the Prandtl ( $Pr$ ) and Graetz ( $Gz$ ) numbers to calculate the Nusselt number. The Prandtl and Graetz equations are shown in Eq. (2.11) and (2.12), respectively.

$$Pr = \frac{c_p \mu}{k} \quad (2.11)$$

where  $c_p$  is the specific heat of the fluid.

$$Gz = \frac{D}{L_{dist}} Re Pr \quad (2.12)$$

where  $L_{dist}$  is the distance from a disturbance.

The Nusselt number correlation for laminar, developing flow by Jacimovic et al. [21] is shown in Eq. (2.13).

$$Nu_{developing,laminar} = 4.364 + \frac{0.055 * Gz^{1.709}}{1 + 0.046 * Pr^{0.078} * Gz^{1.277}} \quad (2.13)$$

Turbulent, fully developed flow uses a correlation by Gnielinski [22] to calculate the Nusselt number. This correlation uses the friction factor,  $f$ , that is discussed in more detail in Chapter 3. The correlation is shown in Eq. (2.14).

$$Nu_{developed,turbulent} = \frac{\left(\frac{f}{8}\right) (Re - 1000) Pr}{1 + 12.7 \left(\frac{f}{8}\right)^{\frac{1}{2}} \left(Pr^{\frac{2}{3}} - 1\right)} \quad (2.14)$$

Turbulent, developing flow uses a correlation by Kakaç et al. [23] shown in Eq. (2.15).

$$Nu_{developing,turbulent} = Nu_f \left( 1 + \frac{0.9756}{\left(\frac{L_{dist}}{D}\right)^{0.76}} \right) \quad (2.15)$$

where  $Nu_f$  is the fully developed Nusselt number, calculated with Eq. (2.14).

Each fluid node represents a volume of fluid that has an inlet and an outlet depending on the path configuration. Certain branch nodes have multiple inlets or outlets. The outlet temperature of a fluid volume is the inlet temperature of the next corresponding volume. The change in temperature across a fluid volume is dependent on the heat entering the node, as shown in Eq. (2.16).

$$q_{fluid} = \dot{m}c_p(T_{outlet} - T_{inlet}) \quad (2.16)$$

where  $q_{fluid}$  is the rate of energy entering the fluid node,  $\dot{m}$  is the mass flow rate of the fluid, and  $c_p$  is the specific heat of the fluid. Because the path architecture is broken down into many small zones, the temperature profile from a zone's inlet to outlet is assumed to be linear, and the heat flux is constant. Therefore, the fluid node's temperature value is set to be the average of the inlet and outlet, as shown in Eq. (2.17).

$$T_{fluid} = \frac{T_{outlet} + T_{inlet}}{2} \quad (2.17)$$

Combining Eq. (2.16) and (2.17) allows each fluid node's temperature to be a function of the energy entering the node and the temperature at the inlet of the node's zone. The boundary condition at the receiver inlet is given as an input to the model. By making each fluid node dependent on their surrounding nodes, the fluid nodes can be organized as a set of linear equations that can be solved at the same time as the other receiver nodes.

#### 2.1.4 System of Equations

The thermal resistance network is set up as a system of linear equations, shown in Eq. (2.18).  $N$  is the total number of nodes in the system.

$$\mathbf{Rt} = \mathbf{b} \quad (2.18)$$

where  $\mathbf{R}$  is the  $N \times N$  resistance matrix,  $\mathbf{t}$  is the temperature vector of size  $N$ , and  $\mathbf{b}$  is the energy vector of size  $N$ .

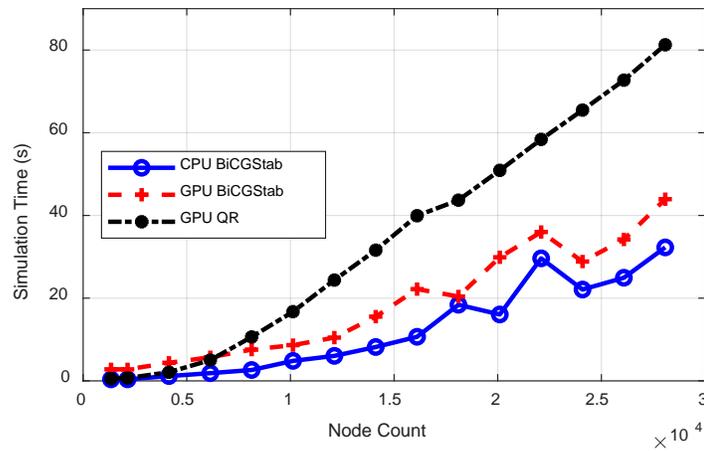
To form the system of equations, each node is given an index that represents their position in the resistance matrix as well as in the temperature and energy vectors. Next, each node fills out their resistance matrix row by assigning resistance values to the nodes they interact with, represented by their column position. Boundary conditions, such as external solar flux, are added to appropriate nodes via the energy vector,  $\mathbf{b}$ . This results in each nodes' energy balance being represented in matrix form in Eq. (2.18).

#### 3.1.5 Solving the System of Equations

The resistance network is organized as a system of linear equations in order to solve the system efficiently. Several methods were investigated to solve the system, including both CPU and GPU based approaches. A CPU QR decomposition function by Math.NET [24] was initially used by the model. This method, however, was not optimized for sparse matrices and was inefficient. A GPU QR decomposition function in the cuSOLVER [25] toolkit drastically improved performance, but QR decomposition is memory intensive and requires large GPU memory resources. Both the GPU and CPU QR decomposition methods were direct solvers, that solved the equations without error. For this model, which is based on experimental correlations and meant as a design tool, exact solutions are not necessary.

A biconjugate gradient stabilized (BiCGStab) method [26] was developed to solve the system of equations iteratively. The CPU and GPU methods were created in C# and CUDA, respectively. The BiCGStab solve method has several advantages compared to the QR decomposition methods. The solve method is iterative, meaning it does not solve the equations directly, and therefore can save time, depending on the convergence tolerance. The method also can be helped by supplying final temperature guesses. As the overall system is solved iteratively (for the surface temperature), the BiCGStab temperature guesses become increasingly accurate. Therefore, each iteration solves more quickly than the previous one. This greatly increases the performance compared to the QR decomposition solvers, which were not benefited by previous solutions. Finally, the BiCGStab method is not as memory intensive, and therefore can handle larger simulations than the QR decomposition method.

The solution times for an arbitrary receiver model with a varying number of total nodes is shown in Figure 2.4. The times for the CPU-based QR solver are not shown because the system could not be solved in a reasonable amount of time.



**Figure 2.4 Simulation Method Solve Time Comparison**

As is seen in Figure 2.4, the quickest solve method was the CPU-based BiCGStab method. The advantage of the CPU over the GPU method is most likely the reduction of costly, time intensive copies from the GPU to the CPU and vice versa. Very large models may benefit from the GPU method, but these would likely require more GPU memory than what is accessible on typical GPUs. Ultimately, the CPU-based BiCGStab method is used in the optimization routines because it utilizes a single CPU core, instead of an entire GPU. As CPU cores are cheaper and more easily accessible, more simulations could be run in parallel during the optimization, saving time.

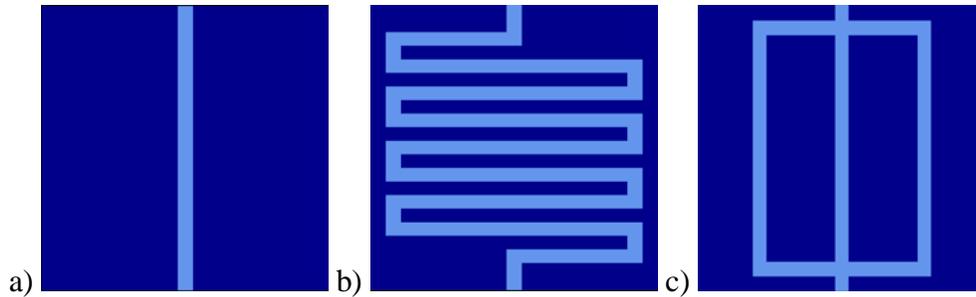
## **2.2 Thermal Model Comparison with CFD**

The thermal model was compared with computational fluid dynamics (CFD) simulations to ensure that the model is accurate. Experimental testing will validate the model in the future, but as the composite receiver is still in development, testing has not yet begun. CFD provides the most comprehensive analysis available at this time and its results are considered the “true” behavior of the receiver for the purposes of this work.

Section 2.2.1 discusses the CFD model and Section 2.2.2 compares the CFD results with the thermal model results. The CFD pressure results are discussed in Chapter 3.

### 2.2.1 CFD Model

ANSYS Fluent was used to simulate the composite receiver in CFD. Three path architectures were modeled: straight, serial, and parallel. These path designs were first created in the resistance model, as shown in Figure 2.5. The designs in the resistance model are set on a discretized grid and represent the elbows as sharp, 90 degree turns, while in reality they are modeled as curved bends.



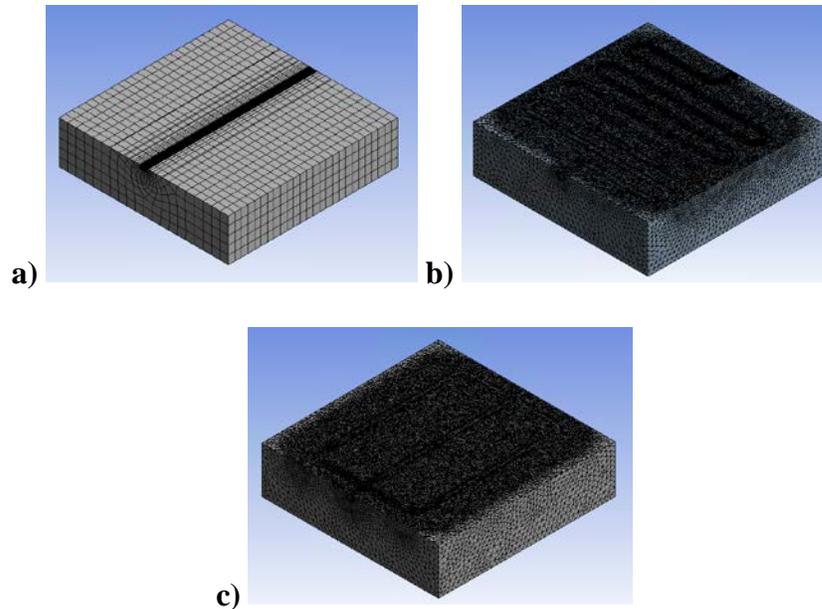
**Figure 2.5 Path Types a) Straight b) Serial and c) Parallel**

The thermal resistance model designs were replicated in CFD. The process of generating the geometry, meshing the bodies, and running the simulation in Fluent is time intensive and is why only three path architectures were considered. These three designs were chosen to represent simplified straight channels, complex winding channels, and splitting, parallel channels that can exist in the microvascular pathway.

The three-dimensional bodies were modeled and meshed in ANSYS. The meshing for each path architecture included an inflation layer to capture the near wall behavior of the fluid, and the fluid body was sized more finely than the receiver. The mesh sizing was set to ensure that the models converged sufficiently. The serial and parallel designs required finer meshes than the straight design, because the flow encountered more disturbances leading to higher turbulence. The mesh sizing for the fluid body and receiver body for each geometry are shown in Table 2.1. A section view of the mesh for each path design is shown in Figure 2.6.

**Table 2.1 Mesh Sizing**

<b>Path Type</b>	<b>Receiver Body Sizing (m)</b>	<b>Fluid Body Sizing (m)</b>
Straight	1e-3	5e-5
Serial	5e-4	7e-5
Parallel	5e-4	7e-5



**Figure 2.6 Mesh Section View a) Straight b) Serial and c) Parallel**

The sCO<sub>2</sub> material properties were considered constant to limit the complexity of the CFD model. The resistance network model supports temperature-dependent properties, but CFD would not converge in any reasonable amount of time with the added complexity of temperature-based properties. The CFD comparison remains valid because both models used the same constant properties, which assume a constant temperature and pressure of 873 K and 20 MPa, respectively. A full discussion of temperature-dependent sCO<sub>2</sub> properties is found in Appendix A.

The boundary conditions for the CFD model are the same as the resistance network model. The fluid enters the receiver at a uniform velocity and temperature. The outlet is considered a pressure outlet. The receiver's receiving surface interacts with the environment with the same heat transfer mechanisms: constant solar flux from the heliostats, radiation exchange with the environment, and convection with the environment. All other receiver walls are considered adiabatic.

The heat flux on the surface boundary is not possible in Fluent without a custom user defined function (UDF). A UDF was created that exchanged heat in the same manner discussed in Section 2.1.2, with solar flux, radiation, and convection. The inclusion of the UDF, and specifically the reliance on a surface temperature to the fourth power term, increased the complexity of the model and required a very fine mesh.

Fluent solved the fluid flow with a k-epsilon model that performed well over a wide variety of flow states (laminar, transitioning, and turbulent). The energy was also solved to model the temperatures throughout the receiver and fluid. The residual convergence criteria were set to  $1e-3$ , except for the energy residual, which was set to  $1e-6$ .

### 2.2.2 CFD Results Comparison

The three path architectures shown in Figure 2.5 were simulated in CFD. A wide variety of inlet velocities were modeled, from 1 m/s to 100 m/s. This correlates to a Reynolds number from 572 to 57,244, which represents flow states from laminar to turbulent.

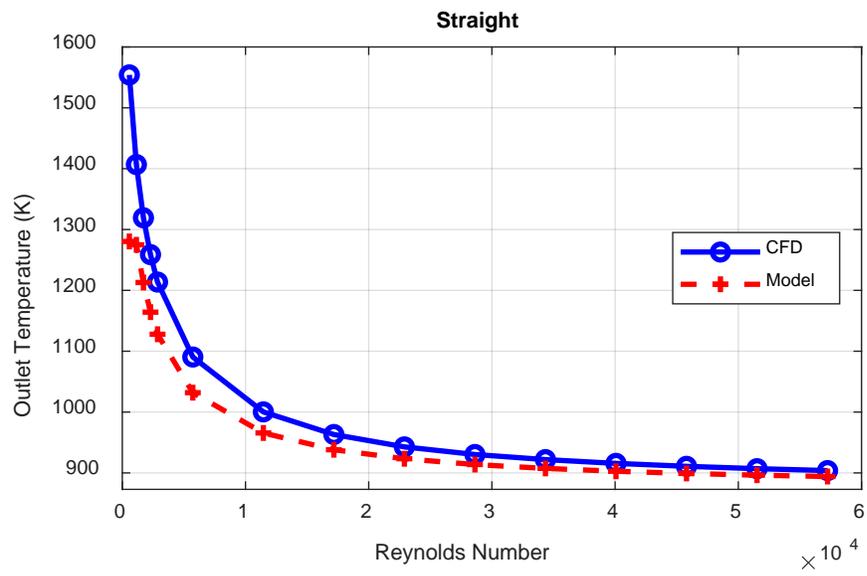
The receiver was modeled with estimated real-life parameters. These model parameters are shown in Table 2.2.

**Table 2.2 Model Parameters**

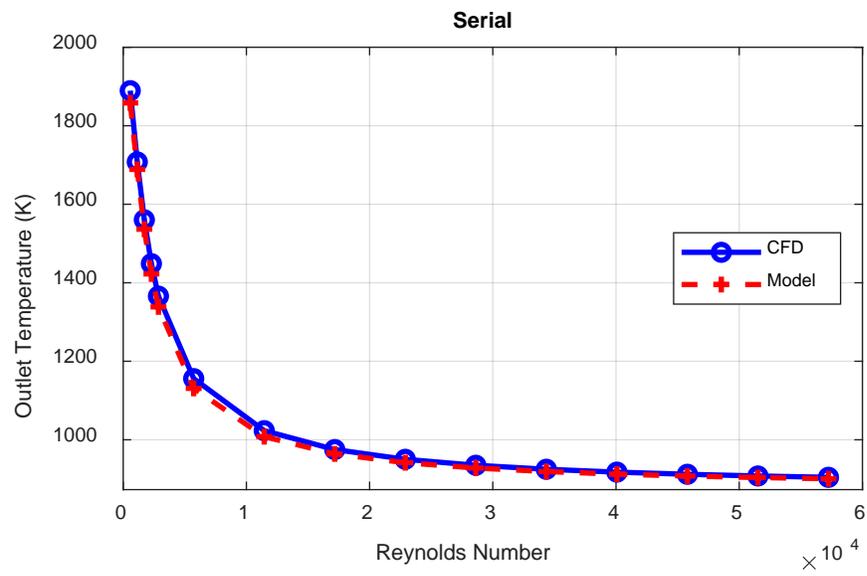
<b>Parameter</b>	<b>Value</b>
Solar Flux (W/m <sup>2</sup> )	1,000,000
Receiver Dimensions (L x W x H) (m)	0.02 x 0.02 x 0.01
Turn Radius (m)	0.001
Inlet Temperature (K)	873
Inlet Pressure (MPa)	20
Channel Diameter (m)	0.001
Emissivity	0.88
Absorptivity	0.94
Receiver Thermal Conductivity (W / m <sup>2</sup> K)	27.5
External Convective Heat Transfer Coefficient (W / m <sup>2</sup> K)	8

Certain model parameters, such as receiver thermal conductivity, are not known and are only estimates. The CFD model is still valid for the purposes of comparing results with the thermal resistance model because both models use the same values.

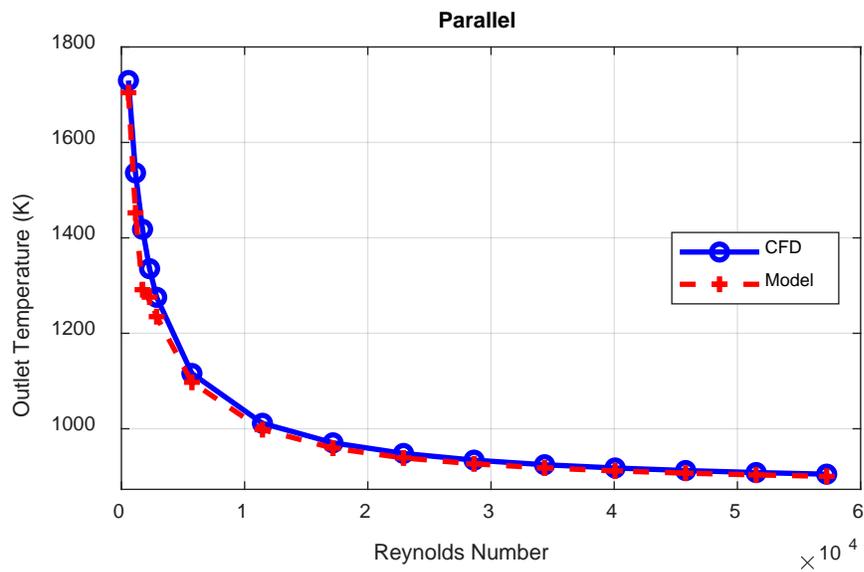
The models were simulated with the stated parameters and Figure 2.7 compares the fluid outlet temperature for each path design, showing the CFD and thermal model results as a function of the inlet Reynolds number.



a)

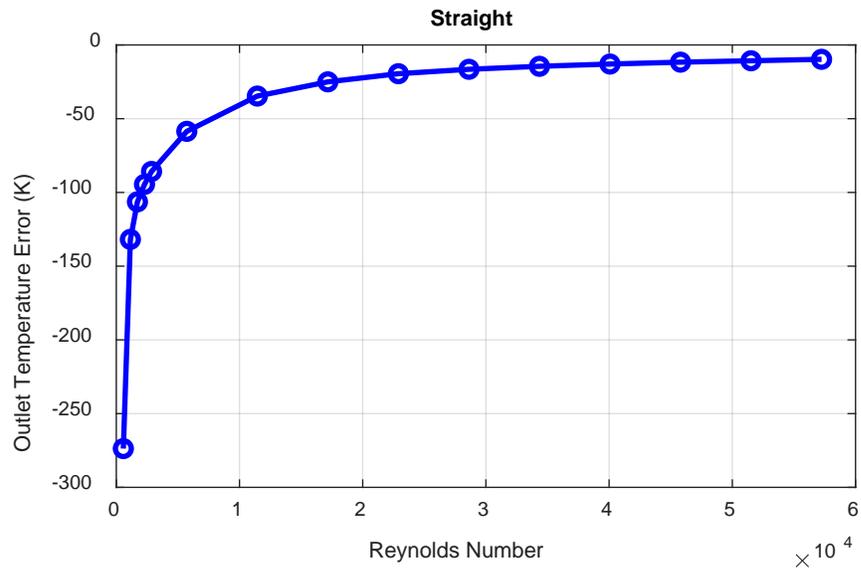
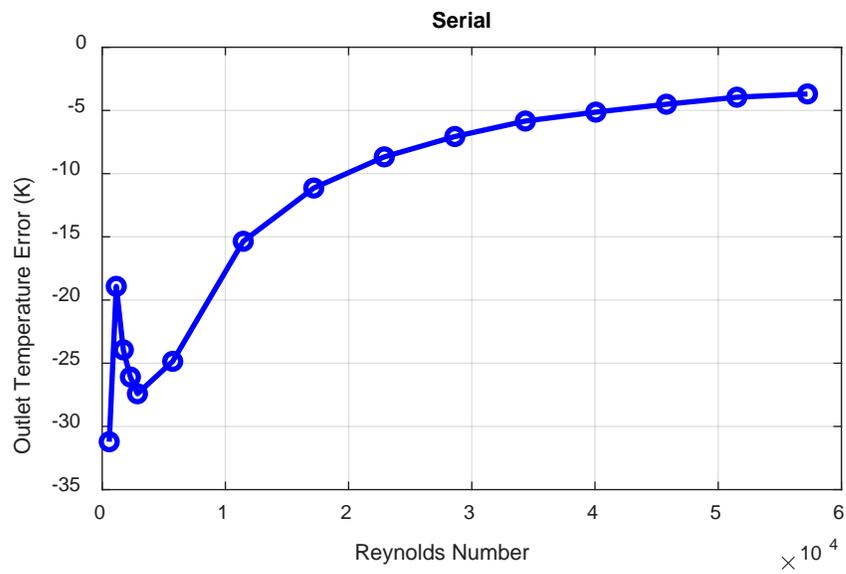


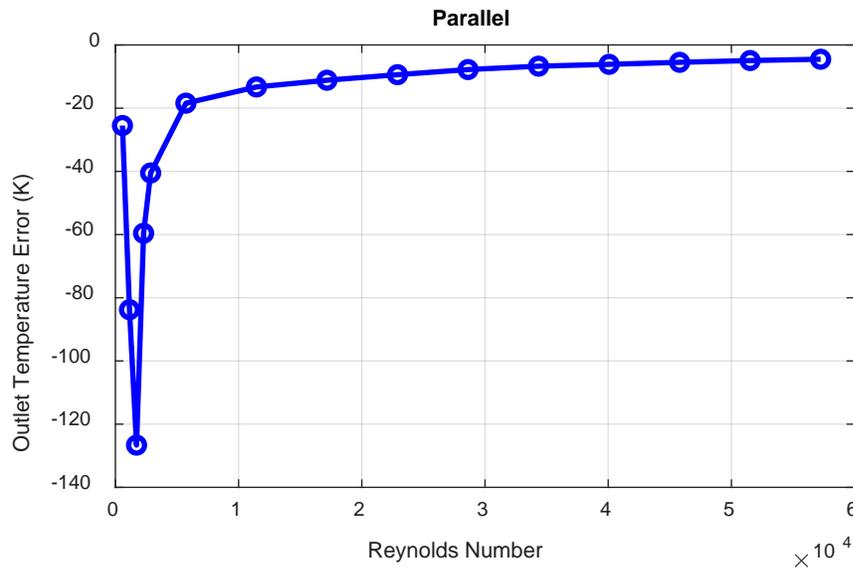
b)



c)  
**Figure 2.7 Outlet Temperature CFD Comparison for a) Straight b) Serial and c) Parallel Designs**

As is shown in Figure 2.7, there is a strong correlation between the CFD and resistance model outlet temperature curves for each path type. To further quantify the results, the difference between the CFD and resistance model outlet temperatures is plotted in Figure 2.8. The CFD temperatures are considered the “real” result, thus error is calculated as the resistance temperature minus the CFD temperature.

**a)****b)**

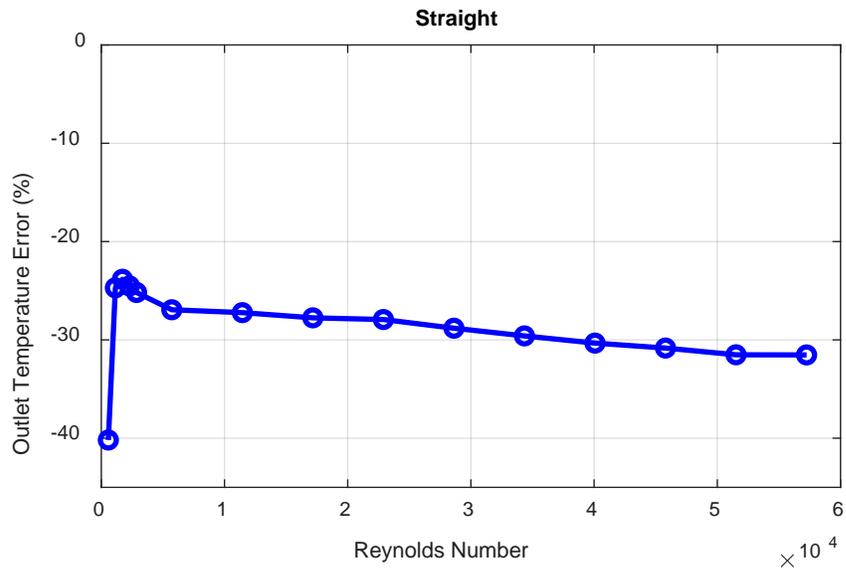
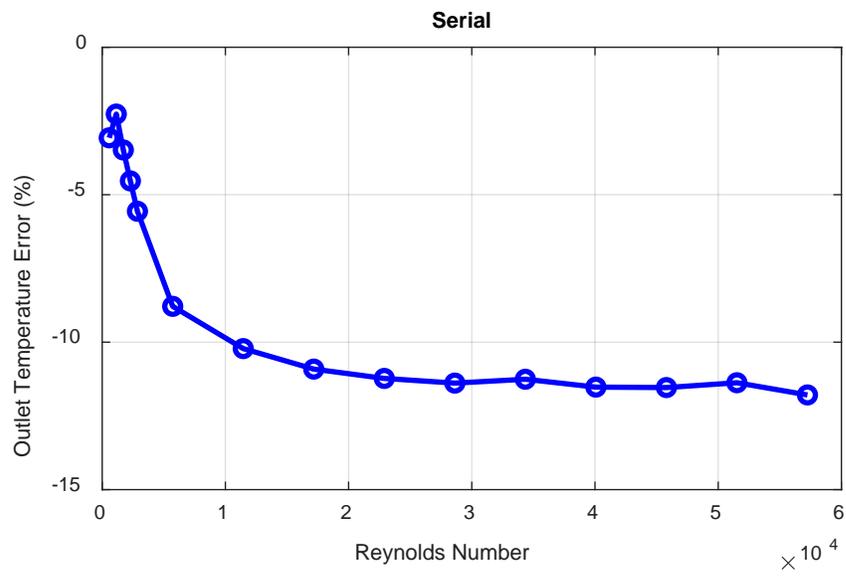


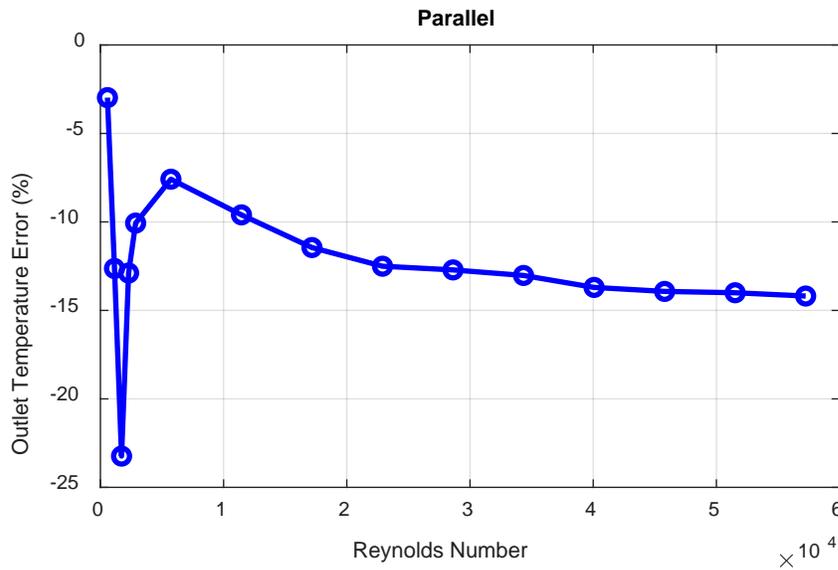
c)  
**Figure 2.8 Outlet Temperature CFD Error for a) Straight b) Serial and c) Parallel Designs**

For each path design, the error between the CFD and resistance model decreases as the velocity increases. This could be due to the flow becoming increasingly turbulent and less transitional. The thermal resistance model considers flow to either be laminar or turbulent and does not account for effects of transitional flow. Another reason the error decreases is that the rise in temperature from inlet to outlet decreases as the velocity increases. The percent error is calculated by comparing the rise in temperature for the CFD and resistance model as shown in Eq. (2.19).

$$\%error = \frac{\Delta T_{model} - \Delta T_{CFD}}{\Delta T_{CFD}} * 100 \quad (2.19)$$

where  $\Delta T_{model}$  is the rise in temperature from inlet to outlet from the resistance model, and  $\Delta T_{CFD}$  is the rise in temperature from the CFD model. Figure 2.9 shows the percent error for each path design.

**a)****b)**



c)  
**Figure 2.9 Outlet Temperature CFD Percent Error for a) Straight b) Serial and c) Parallel Designs**

The straight path has the largest error, with a maximum percent error value of -40%. The straight path is likely less accurate than the serial or parallel paths because of the short path length. Although the thermal model accounts for developing flow, the straight path is likely so short that larger developing effects increase the heat flow to the fluid that are not represented in the model.

The serial path has the lowest error of the three path designs, with a maximum percent error of -12%. This error is at the final data point, a velocity of 100 m/s, where the rise in temperature across the receiver is decreasing. This causes the percent error to increase even while the temperature difference between CFD and the resistance model is decreasing.

The parallel path performs in a similar manner to the serial path. The largest percent error is -23%. Again, the percent error magnitude increases as the velocity increases even though the temperature difference decreases.

For each design, the effect of the flow state is clear. The error has a local peak in the laminar region before becoming smoother as the flow becomes fully turbulent. As mentioned previously, the thermal correlations assume the flow is either laminar or turbulent. Because of this, it is expected that the highest error would be near this region. As is shown in Chapter 4, the optimal receiver design has turbulent flow, as will most designs considered in any real-world applications, which is the most accurate flow region.

## CHAPTER THREE: PRESSURE MODEL

The carbon composite receiver's pressure drop is modeled by calculating the major and minor pressure loss in each fluid segment and summing over the course of the fluid pathway. The purpose of modeling the pressure drop is to compare different path architectures, not necessarily to simulate the pressure drop with the precision of CFD. Therefore, certain values, such as minor loss coefficients, were chosen based on CFD simulations and are not based on experimental results. Also, unlike the thermal model, all flow is considered to be fully developed when calculating the pressure drop to simplify the model and reduce the number of unknown coefficients.

The pressure model is broken into two major pieces: major and minor losses. Major losses are the pressure drop in a straight path, and minor losses occur in bends or crosses. Major and minor losses are discussed in Section 3.1.1 and 3.1.2, respectively.

Paths with parallel channels do not have a constant mass flow rate. To calculate the mass flow rate in any channel, the Hardy Cross method is implemented. This method ensures that the mass flow is distributed according to pressure drop along each path. This is discussed in detail in Section 3.1.3.

### **3.1 Pressure Model Methodology**

#### 3.1.1 Major Losses

Major loss occurs in straight paths and is calculated with the Darcy-Weisbach equation, which is dependent on a friction factor, as shown in Eq. (3.1).

$$\Delta p_{major} = f \frac{L\rho V^2}{2D} \quad (3.1)$$

where  $L$  is the length of the segment,  $\rho$  is fluid density,  $V$  is fluid velocity, and  $D$  is the channel diameter.

The friction factor,  $f$ , is calculated with Eq. (3.2) by Cheng [27].

$$\frac{1}{f} = \left(\frac{Re}{64}\right)^\alpha \left(1.8 \log \frac{Re}{6.8}\right)^{2(1-\alpha)\beta} \left(2 \log \frac{3.7D}{k_s}\right)^{2(1-\alpha)(1-\beta)} \quad (3.2)$$

where  $k_s$  is the channel roughness.  $\alpha$  and  $\beta$  are defined in Eq. (3.3) and (3.4), respectively.

$$\alpha = \frac{1}{1 + \left(\frac{Re}{2720}\right)^9} \quad (3.3)$$

$$\beta = \frac{1}{1 + (Re/(320r/k_s))^2} \quad (3.4)$$

The Cheng correlation is valid over all Reynolds numbers, meaning the same equation is used to calculate major loss in laminar, transitioning, and turbulent flow. Multiple correlations exist that are intended to be valid over a wide flow range; Cheng was selected for its agreement with CFD modeling, as discussed in Section 3.2.

### 3.1.2 Minor Losses

Minor losses compose all other pressure losses in the path architecture. There are three forms of minor loss considered in the model: elbows, tees, and crosses. The loss for each case is dependent on a minor loss coefficient, as detailed in Eq. (3.5).

$$\Delta p_{minor} = k \frac{\rho V^2}{2} \quad (3.5)$$

where  $k$  is the minor loss coefficient. The forms of minor loss are differentiated by the  $k$  calculation.

### 3.1.2.1 Elbows

Elbows, or bends, are 90° turns that occur in the path. The minor loss coefficient related to elbows varies in literature [20] and has a reported range of 0.2 to 1.5, depending on the connection properties, which are largely unknown at this time. For this work, CFD was used to determine the appropriate minor loss coefficient value that most closely matched the results. This value is shown in Eq. (3.5).

$$k_{elbow} = 0.2 \quad (3.5)$$

### 3.1.2.2 Tees

Tees occur when one path splits into two, or two paths combine into a single path. For the purpose of this work, a tee was assumed to simply perform as an elbow at each outlet. Eq. (3.4) was used to calculate the pressure drop for each outlet, with an elbow's minor loss coefficient, at each outlet's velocity.

### 3.1.2.3 Crosses

Crosses occur at four-way intersections. These intersections can have 1-3 inlets and 1-3 outlets. Calculating the minor loss coefficient for these situations has not been extensively studied in literature, outside of the work by Sharp et al. [28]. Sharp classified crosses as having dividing flow, combining flow, perpendicular flow, or colliding flow depending on the location and number of inlets and outlets.

Sharp produced experimental data of minor loss coefficients for crosses of equal diameter for each type of intersection. The coefficients were reported as contour diagrams for each type of flow. In order to process the information, the diagrams were converted to data points, which were fitted with 2D splines. These splines returned an interpolated minor loss coefficient value for each cross configuration.

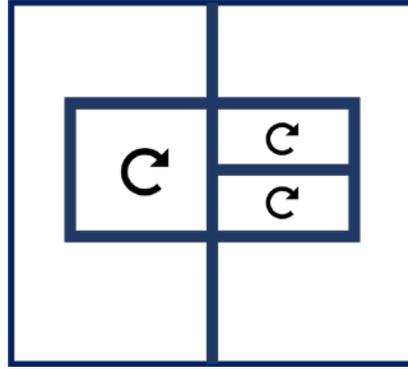
This method of calculating the minor loss coefficients for crosses was implemented, but significantly overestimated the pressure loss when compared with CFD results. These results are discussed in Section 3.2, along with alternative approaches, which are treating crosses in the same manner as tees, or assuming no pressure loss across the connection altogether.

Nikfetrat et al. [29] applied the work of Sharp in a similar manner by fitting bicubic splines to the experimental data and calculating the minor loss coefficient. It was discovered that the model could produce error up to 100% and even incorrectly estimate the flow direction. The researchers recommended applying the model if extensive data is available, which is not the case for the carbon composite receiver.

### 3.1.3 Mass Flow Distribution

The thermal and hydraulic model can support path architectures with branching and dividing paths. This is accomplished by calculating the mass flow rate in each path, which is not simply the mass flow rate divided by the number of parallel channels. The mass flow rate is determined by the pressure drop in each path; specifically, the head loss along each “loop” in a path design must sum to zero and the mass going in a cross or tee must equal the mass leaving the cross or tee.

To solve for the correct mass flow in every channel, the Hardy Cross method [30] was implemented. The method starts by assuming mass flow in each channel. This was done by assuming the mass flow was distributed equally at each branch outlet. Next “loops” were formed that travelled along pathways until reaching back to the starting node. An example of loops is shown in Figure 3.1.



**Figure 3.1 Path Architecture Loop Example**

The head loss for each loop is calculated and summed. The head loss is related to pressure drop, as shown in Eq. (3.6).

$$\Delta H = \frac{\Delta P}{\rho g} \quad (3.6)$$

where  $g$  is the gravitational constant. Only major losses were considered when calculating the head loss in a loop. This was done because the minor loss at a cross or a tee is not calculated specifically for a single outlet, but rather for the entire cross or tee. Therefore, only major losses are considered.

If the head loss about a loop is not equal to zero, a correction factor is added to the mass flow rate in that loop, as shown in Eq. (3.7). The correction factor formula is shown in Eq. (3.8).

$$\dot{m} = \dot{m}_0 + \Delta \dot{m} \quad (3.7)$$

where  $\dot{m}$  is the new mass flow rate,  $\dot{m}_0$  is the current mass flow rate, and  $\Delta \dot{m}$  is the correction factor.

$$\Delta \dot{m} = \frac{-1}{2} \frac{\sum \Delta H}{\sum \left( \frac{\Delta H}{\dot{m}} \right)} \quad (3.8)$$

where  $\sum \Delta H$  is the sum of the head loss around the loop.

This process of summing the head losses around each loop and applying the correction factor is repeated until  $\Delta\dot{m}$  is sufficiently converged to zero.

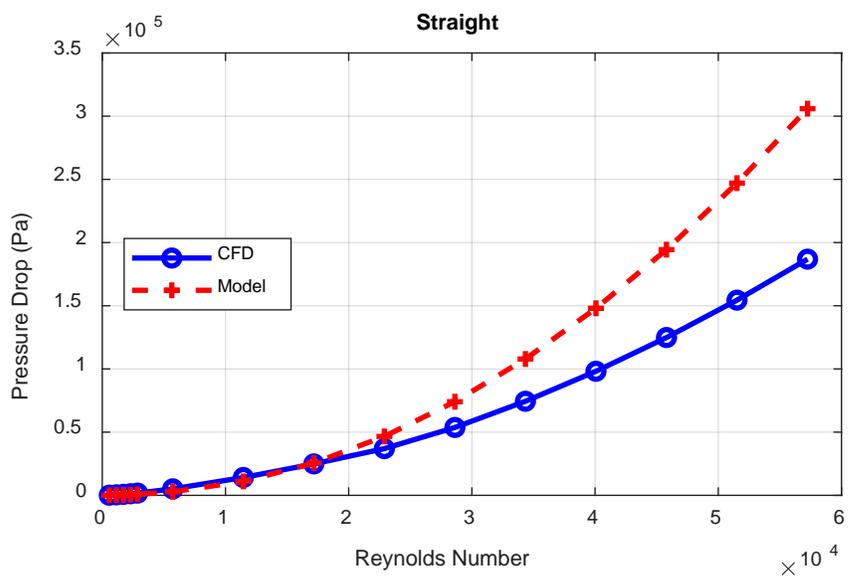
### **3.2 Pressure Model Comparison with CFD**

The pressure model was validated in the same manner as the thermal model – with CFD. The CFD model is described in detail in Section 2.2 and the same cases were used for the pressure model validation. These cases consist of three path architectures, straight, serial, and parallel, with a variety of inlet velocities from 1 m/s to 100 m/s.

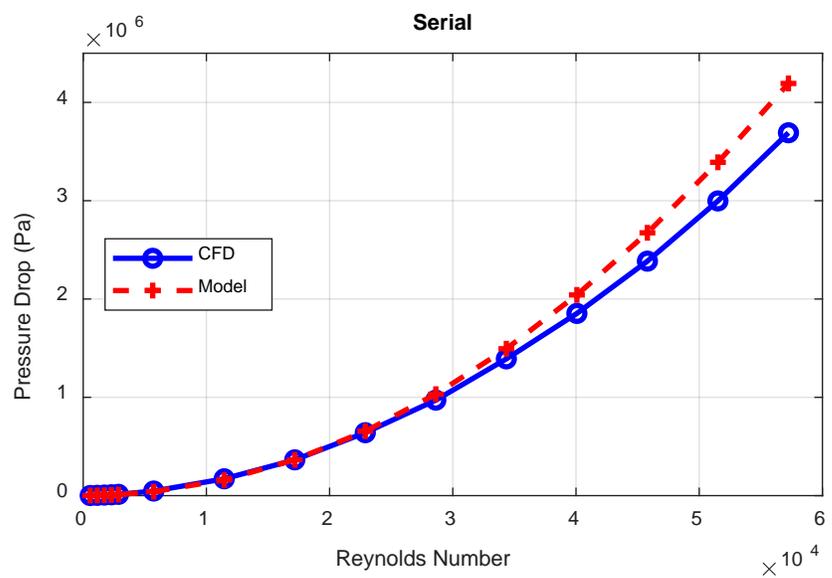
The straight path architecture focuses on major pressure losses, as it does not have any minor losses in the design. The serial path is a series of bends that is used to refine the minor elbow loss coefficient. The parallel path is used to analyze crosses.

#### 3.2.1 CFD Results Comparison

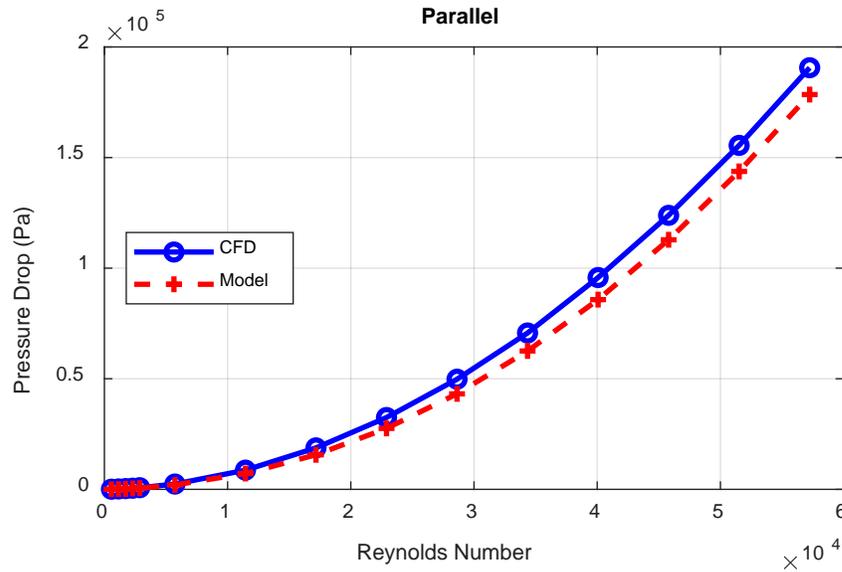
The CFD results are considered the “real” performance of the receiver, as experimental testing has not yet begun. The results in this section assume no pressure loss in crosses. Section 3.2.2 compares the different approaches for modeling the pressure drop in crosses. Figure 3.2 shows the pressure drop of each receiver design as a function of Reynolds number for CFD and the pressure model.



a)



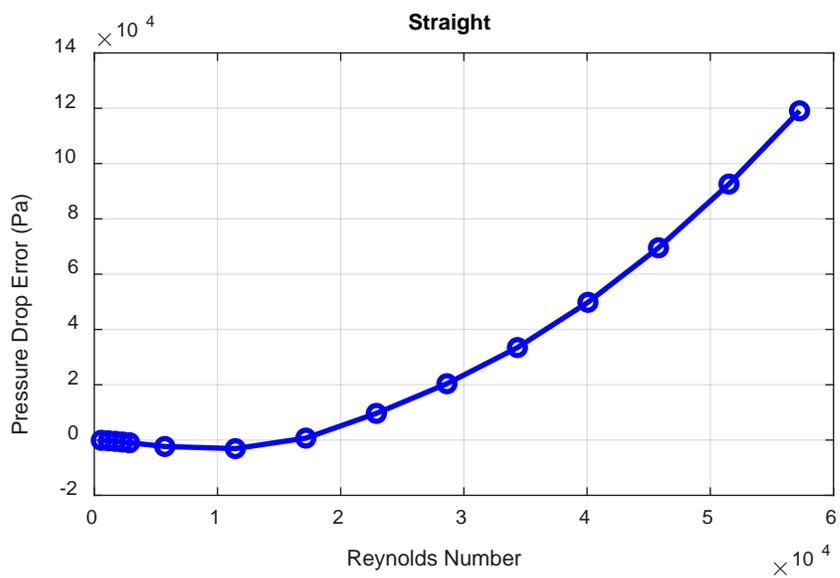
b)



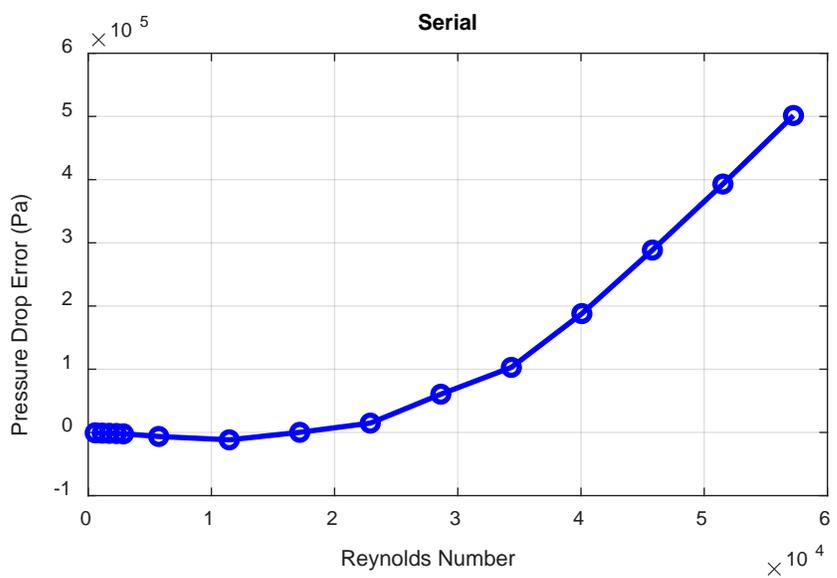
c)  
**Figure 3.2 Pressure Drop CFD Comparison for a) Straight b) Serial and c) Parallel Designs**

The performance of the pressure model is highly dependent on the path architecture. The straight path pressure model has a similar shape as the CFD results, but the pressure model curve is steeper than that of the CFD. This is likely due to the developing flow effects that are not accounted for in the pressure model. The serial path has strong agreement between the pressure and CFD models with a slightly steeper curve. The parallel pressure model has a similar curve but produces lower pressure drops than the CFD model and is discussed in more detail in Section 3.2.2.

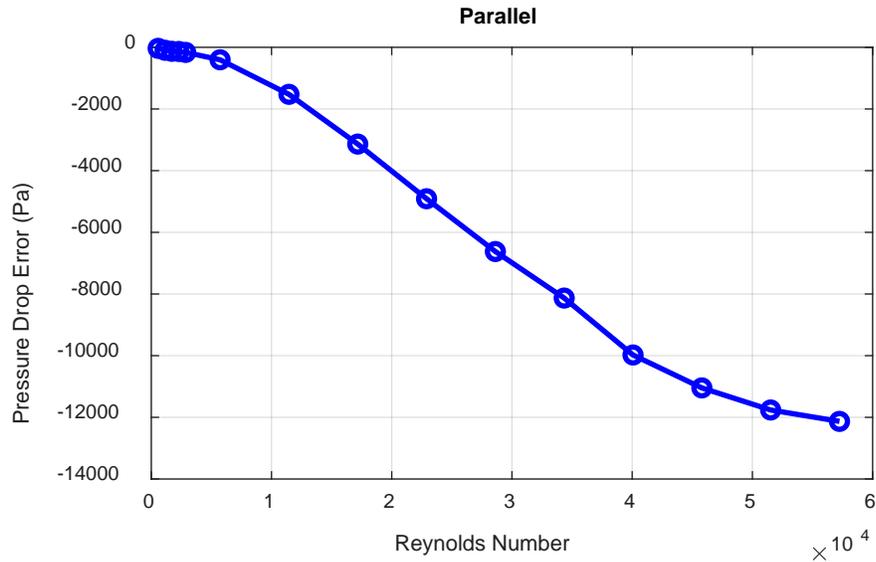
As with the thermal results, error is calculated as the model pressure drop minus the CFD pressure drop. The error results are shown in Figure 3.3.



a)



b)



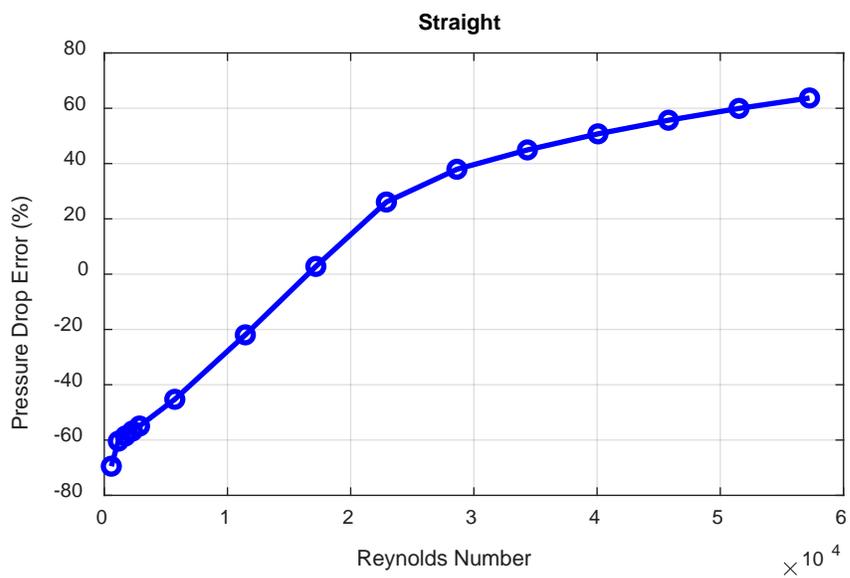
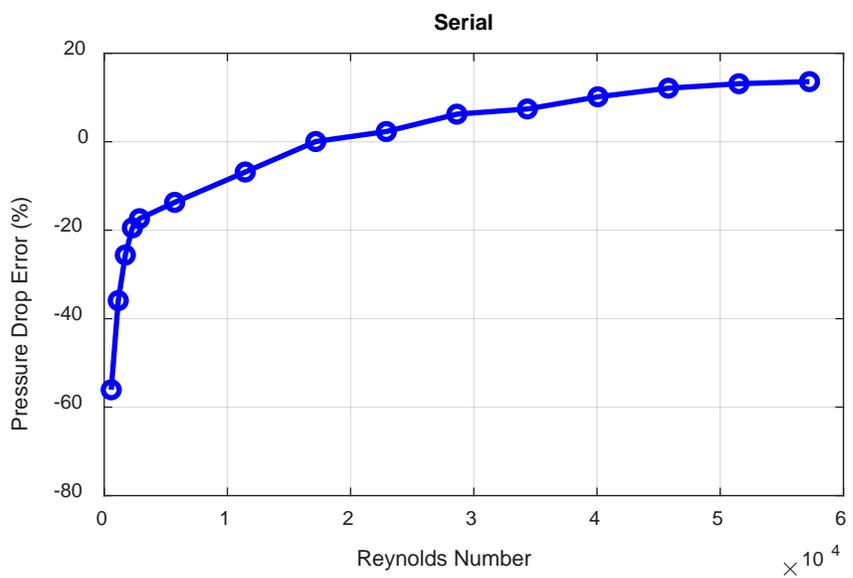
c)  
**Figure 3.3 Pressure Drop CFD Error for a) Straight b) Serial and c) Parallel Designs**

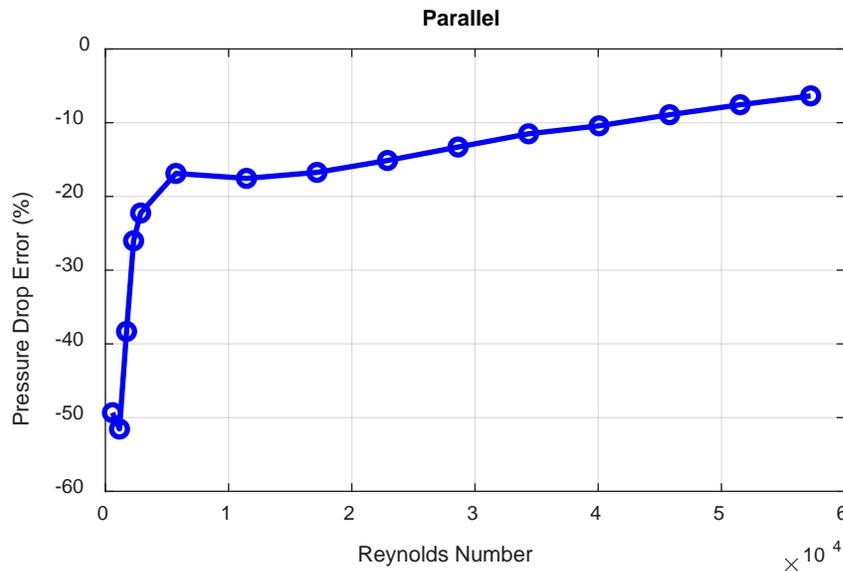
With each path design, the error magnitude increases as the velocity increases.

For every case, the overall pressure drop increases with velocity as well. To better understand the error, the percent error was calculated using Eq. (3.9).

$$\%error = \frac{\Delta P_{model} - \Delta P_{CFD}}{\Delta P_{CFD}} * 100 \quad (3.9)$$

where  $\Delta P_{model}$  is the pressure model's pressure drop and  $\Delta P_{CFD}$  is the CFD model's pressure drop. Figure 3.4 shows the percent error for each path.

**a)****b)**



c)  
**Figure 3.4 Pressure Drop CFD Percent Error for a) Straight b) Serial and c) Parallel Designs**

The percent error plots show the difference in accuracy between each path type. The straight path begins by underestimating the pressure drop, reaching a peak at -70%, before rising and eventually overestimating the loss. The serial path also starts by underestimating the loss, peaking at -56%, before asymptotically rising to an error near 14%. Similarly, the parallel path reaches a peak of -52% at lower velocities before slowly improving, ending at -6%.

Inverse to the behavior of the outlet temperature, low velocity cases result in lower pressure drops, which can cause the percent error to rise in magnitude, even if the actual drop difference is lower. For instance, the peak -70% percent error at 1 m/s for the straight path is only a difference of 94 Pa, whereas the 3% error at 20 m/s is a difference of 697 Pa.

As was mentioned previously, the increasing error as a function of inlet velocity for the straight path could be due to the relatively short path length. The developing flow

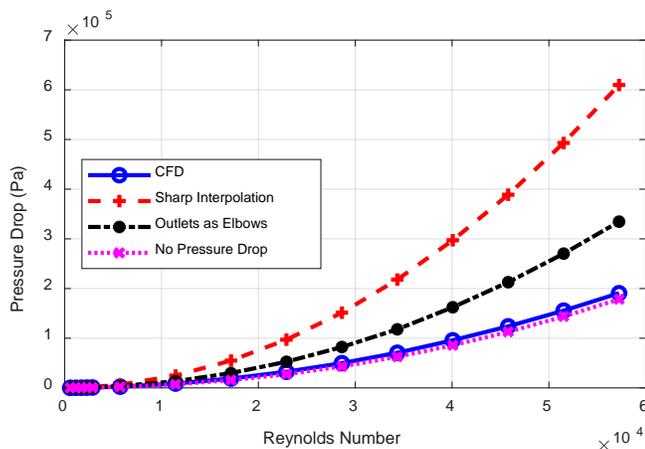
effects are not considered in the pressure model and likely decrease the overall pressure drop in highly turbulent flow. The serial path, which essentially is a longer straight path with added minor loss elbows, becomes more accurate as the velocity increases.

The pressure loss error for the parallel path is due to the assumption that crosses do not have a pressure drop. This is discussed in more detail in Section 3.2.2, which compares different modeling techniques for crosses with the CFD results.

### 3.2.2 Cross Analysis

The inconsistencies between the pressure model and the CFD results for the parallel path are due to the pressure drop calculation at the two crosses. This is proven by varying the pressure drop calculation at the crosses. Three modeling techniques were explored for calculating the pressure drop at a cross and these techniques were compared to the CFD results.

Figure 3.5 compares the pressure drop for each modeling method with the CFD results as a function of the Reynolds number. The three methods are: Sharp spline interpolation, treating outlets as elbows, and removing cross pressure drop entirely.



**Figure 3.5 Cross Pressure Drop Comparison**

From the results in Figure 3.5, the pressure drop overestimation from the Sharp interpolation and from treating the outlets as elbows clearly comes from the two crosses. This is evident because the pressure drop results are closer to CFD if the crosses' pressure loss is assumed to be zero. From Sharp and Nikfetrat's work, it is known that modeling a cross with a minor loss coefficient may not be suitable and does not capture the full behavior at the cross. Although treating each outlet as an elbow provides closer results, the model still produces pressure drops nearly twice as large as the CFD.

Although assuming the pressure drop at crosses is zero does not make physical sense, it matches the CFD results most accurately. In Sharp's work, certain cross configurations can result in zero or even negative pressure drops meaning that in some specific cases, assuming no pressure loss across a cross could be realistic. However, the likely reason the loss is most accurate with the zero pressure drop assumption is due to the relative insignificance of the drop compared to the rest of the path.

At the highest velocity, the no cross pressure drop assumption's pressure drop is 6% less than the CFD results. If the pressure calculation for every part of the path except for the crosses is assumed to be accurate, the two crosses would account for 3% of the total pressure drop each. While 6% is significant, it is considerably less than the error produced by any other modeling method.

## CHAPTER FOUR: OPTIMIZATION

The composite receiver's thermal and hydraulic model were developed, in part, to optimize the design of a receiver unit cell. An optimal receiver design would maximize the net energy gain from the unit cell, while operating within a certain strain constraint. The net energy gain is dependent on the thermal efficiency calculated in the thermal model and the pressure drop calculated with the hydraulic model. Energy absorbed by the receiver's working fluid is considered positive, and the energy it takes to run a pump over the pressure loss is negative. This cost relationship is discussed in more detail in Section 4.1.4.

A genetic algorithm, developed by Holland in 1975 [31], was chosen to optimize the receiver design because it can optimize continuous and discrete variables. For this work, channel diameter and fluid velocity are examples of continuous variables. The channel architecture of the microvascular pathway is parameterized by two discrete variables: path type and path value. Path type describes one of three path configurations: serial, parallel, or bifurcating. Path value sizes the path according to the path type. For example, for serial paths, path value determines the number of turns and for parallel paths, path value determines the number of parallel paths. Path value was constructed in such a way to generally make path types with the same path value have similar channel surface areas. This is discussed in more detail in Section 4.1.2.

The genetic algorithm was run multiple times with special conditions to ensure the algorithm was converging to a global optimum, rather than local. The initial

population seeding was varied from uniform to random, as described in Section 4.1.3.

Activation functions manipulated the fitness value to limit premature convergence. These functions are discussed in Section 4.1.5.

Section 4.2 discusses the optimization algorithm's results, showing strong convergence between every test case and generating an optimal design with a better "fit" than alternatives.

## **4.1 Genetic Algorithm Methodology**

A genetic algorithm involves a population of individuals that evolve to generate an optimal design. An individual is a collection of properties that describes a specific design case. For this work, an individual consists of four properties: diameter, velocity, path type, and path value. The population evolves each iteration of the algorithm via crossover and mutation. Crossover is the act of two individuals combining to form a new "child" and mutation randomly changes certain individuals' properties.

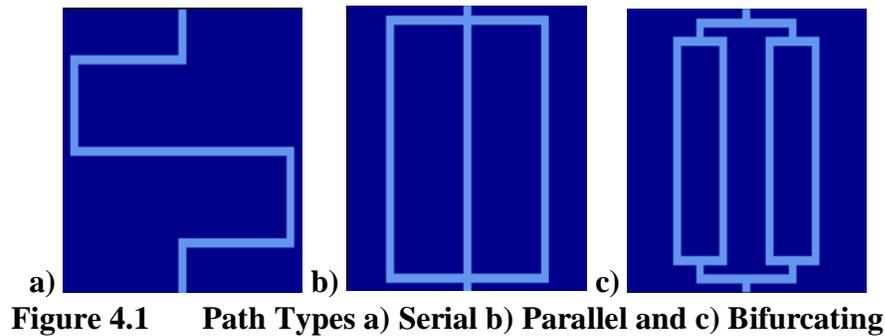
The evolution process proceeds for a set number of iterations before terminating. This is done because generally the best individual takes steps towards optimal fitness, rather than converging to a certain value. Validating that the algorithm has truly found the optimal design is accomplished by varying seeding and activation functions. The entire process, from seeding to validation, is described in the following subsections.

### **4.1.1 Design Variables**

There are four design variables that are optimized with the genetic algorithm. They are: channel diameter, fluid inlet velocity, path type, and path value. The channel diameter is uniform across the receiver, regardless of path architecture. The velocity

profile is constant at the inlet and varies with mass flow rate throughout the architecture, as described in Chapter 3.

Path type and path value, both briefly described previously, parameterize the path architecture. Path type is composed of three designs: serial, parallel, and bifurcating. An example of each path type is shown in Figure 4.1.



**Figure 4.1 Path Types a) Serial b) Parallel and c) Bifurcating**

The serial path type consists of a series of turns that direct the fluid back and forth to each side on the way to the outlet. The turns are spaced so that the distance between each horizontal segment is equal. The parallel path type consists of straight parallel channels. The horizontal spacing between each channel is equal. Finally, the bifurcating path type is a series of splitting channels that result in an equal division of mass flow rate in every parallel segment. The horizontal distance between each internal parallel path is equal.

Each path type is a function of path value, which determines the complexity of the architecture. For serial paths, path value determines the number of horizontal segments that are connected with bends. If  $n_{path}$  is the number of horizontal segments, it is related to path value with Eq. (4.1).

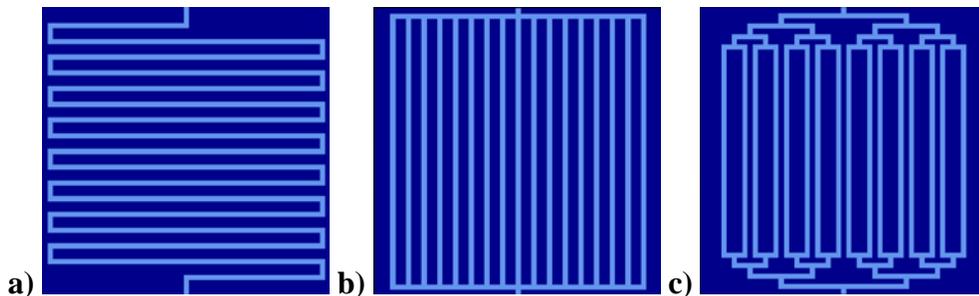
$$n_{path} = 2^{PathValue-1} + 1 \quad (4.1)$$

The same method of determining the number of horizontal segments in the serial path type is true for determining the number of parallel channels in the parallel path type. The number of parallel channels in a parallel path type is related to path value with Eq. (4.1) as well.

The purpose of relating path value with  $n_{\text{path}}$  in this way is based on the relationship path value has with the bifurcating path type. For bifurcating paths, path value determines the number of splits. Specifically, the number of splits is one less than path value. The path value is related to the number of internal parallel paths,  $n_{\text{internal}}$ , in the bifurcating path type with Eq. (4.2).

$$n_{\text{internal}} = 2^{\text{PathVal}-1} \quad (4.2)$$

To demonstrate the similarity in path length for path types with the same path value, a path value of 5 results in 17 horizontal segments for the serial path type, 17 parallel channels for the parallel path type, and 16 internal parallel channels for the bifurcating path type. Each example path type is shown in Figure 4.2. As is shown, path value was constructed in such a way to generally result in similar path lengths for path types with the same path value.



**Figure 4.2 Path Length Similarity for a) Serial b) Parallel and c) Bifurcating**

#### 4.1.2 Seeding

The genetic algorithm begins by creating a population of individuals, each with a set of properties that describe a specific receiver design. Generating this initial population is called population seeding. For genetic algorithms with a large design space, seeding is used to place individuals in optimal positions, to converge the algorithm more quickly [32]. For this application, the seeding method is varied to test the algorithm, and ensure that global optimums have been reached. This work implements three simple seeding techniques: random, max, and min.

Random seeding involves giving each individual a random set of properties. Diameter and velocity are randomly chosen between a maximum and minimum value. Path type is randomly selected from serial, parallel, and bifurcating path types. Path value is randomly selected from 1 to the max value that can fit within the grid size that represents the path design space. For random seeding, individuals do not have the same properties; they are varied randomly.

Max seeding gives every individual the maximum value for each property. Diameter and velocity are set to the maximum value determined as an input to the algorithm. Path type is randomly selected from serial, parallel, and bifurcating. Path value is set to the max value that can fit in the discretized design space. Min seeding uses the same process as max seeding, except with the minimum possible values.

As is shown in Section 4.2, each seeding type varies the initial performance of the designs, but every case converges to the same global optimums, regardless of seeding type.

### 4.1.3 Fitness Function

Each individual is simulated with the thermal and hydraulic models. This results in a known thermal efficiency and pressure drop across the pathway. In addition, the nominal strain is calculated, as discussed in Section 4.1.5. The genetic algorithm requires a single cost function that describes the fit for each individual. An ideal receiver maximizes thermal efficiency and minimizes pressure drop, and these two values need to be related to each other in the cost function. The cost function used in this genetic algorithm is shown in Eq. (4.3).

$$fit = \eta_{turbine} \eta_{receiver} A_s q_{solar} - \Delta P \frac{\dot{m}}{\eta_{pump} \rho} \quad (4.3)$$

where  $\eta_{turbine}$  is the efficiency of the turbine used to convert heat to electricity,  $\eta_{receiver}$  is the receiver thermal efficiency,  $A_s$  is the receiver top surface area,  $q_{solar}$  is the heat flux from the heliostat field,  $\Delta P$  is the pressure drop,  $\dot{m}$  is the mass flow rate of the fluid in the receiver,  $\eta_{pump}$  is the efficiency of the pump used to flow the fluid through the receiver, and  $\rho$  is the density of the fluid.

The cost function relates the energy the receiver cell produces with the amount of energy the pump requires to overcome the pressure drop along the fluid path. This relationship combines the thermal efficiency and pressure drop into one equation that is not arbitrarily based on weighting coefficients. From the way the equation is constructed, fitness values can be positive or negative, with positive values being better.

### 4.1.4 Activation Functions

Another way to ensure global optimums are not local is to apply activation functions to the fitness values. Activation functions are commonly used with neural networks [33] and for this application they manipulate how different individuals relate to

each other, either by artificially increasing or decreasing the difference in fitness values. This can prevent parameters from converging too quickly by encouraging a more thorough exploration of the design space. Alternatively, activation functions can also guide individuals to converge to more optimal designs, even if the improvement is small.

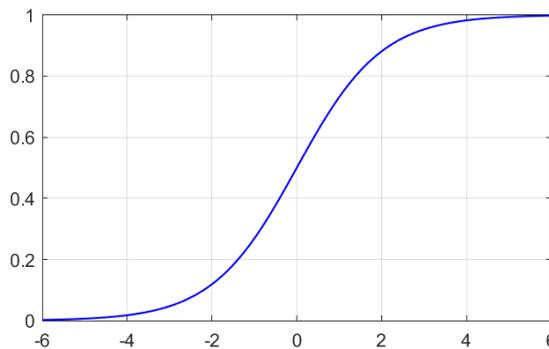
Four activation functions were considered in this work. Each function is a simple method that modifies the relationship between individuals. The first, linear activation, is simply the fitness function with no modifications. This is considered the base case. The square activation function squares the fitness value while maintaining its sign. This is shown in Eq. (4.4).

$$\text{Square}(fit) = \text{sign}(fit) * fit^2 \quad (4.4)$$

Square activation inflates the differences between cases, which should favor cases with higher fitness values. While it is beneficial to favor better cases, the solution could converge too quickly, resulting in a local optimum.

The third activation function studied is a sigmoid activation, also found in [33]. This function converts the fitness to a value between 0 and 1 and is centered at a fitness of 0. The sigmoid activation function is shown in Eq. (4.5) and plotted in Figure 4.3.

$$\text{Sigmoid}(fit) = \frac{1}{1 + e^{-fit}} \quad (4.5)$$



**Figure 4.3 Sigmoid Activation Function**

As is seen in the sigmoid plot, differences in fitness values are inflated near 0 and decreased otherwise. This should result in distinction between similarly performing cases near 0 while not heavily favoring better performing cases. The issue with this function is it focuses on fitness values near zero. To account for this, the variable sigmoid activation function was developed.

The variable sigmoid function, VarSig, was developed for this work to shift the sigmoid function depending on the fitness results. The first iteration of the algorithm starts with the sigmoid function centered at zero. Every following iteration centers the sigmoid function at the best case's fitness value. This is shown in Eq. (4.6).

$$VarSig(fit) = \frac{1}{1 + e^{-(fit - fit_{best})}} \quad (4.6)$$

where  $fit_{best}$  is the best fitness value from the previous iteration of the algorithm.

The purpose of shifting the sigmoid function is to maximize the differences in fitness values for cases near the best case. Cases with slightly better fitness values than the current optimal will be heavily favored over cases with slightly worse fitness values. This will increase the likelihood that top cases will get selected to be parents, resulting in more high potential children. The parent selection process is discussed in Section 4.1.6.

#### 4.1.5 Strain Constraint

The nominal strain in each receiver design is considered in the genetic algorithm. Unlike thermal efficiency and pressure drop, which are related to each other in the fitness function, the nominal strain is a hard constraint, meaning any cases with a strain higher than a certain value are removed from the population immediately. This was done to prevent any cases that are not mechanically feasible from being considered.

The nominal strain is calculated from the nominal stress and the modulus of elasticity of the receiver [34]. The nominal stress calculation is shown in Eq. (4.7).

$$\sigma_{nominal} = \frac{P}{\frac{A_{cross}}{A_{channel}} - 1} \quad (4.7)$$

where  $\sigma_{nominal}$  is the nominal stress,  $P$  is the fluid pressure,  $A_{cross}$  is the cross-sectional area of the receiver into the receiving surface, and  $A_{channel}$  is the projected area of the fluid channel. For this calculation, the pressure is considered to be the inlet pressure and does not account for the pressure loss over the fluid pathway, which is insignificant relative to the overall pressure.

Nominal strain is related to the nominal stress with Eq. (4.8).

$$\varepsilon_{nominal} = \frac{\sigma_{nominal}}{E} \quad (4.8)$$

where  $\varepsilon_{nominal}$  is the nominal strain and  $E$  is the modulus of elasticity of the receiver.

More research is being conducted to calculate principle and thermal strains in the receiver design; only nominal strain is considered in this work.

#### 4.1.6 Parent Selection

The genetic algorithm evolves by selecting parent individuals that combine to form children. The number of parents is the crossover parameter multiplied by the population size. Generally, parents should be high performing individuals so that their children have the potential to perform even better. However, the parent selection needs to maintain diversity to allow the entire design space to be explored. Therefore, randomness is key in selecting the parents.

To select parents, a roulette wheel of fit is formed. The roulette wheel is composed of every individuals' fitness value "stacked" to form a list. A random value

between 0 and the sum of fitness values is generated. This value corresponds to a location on the wheel, and therefore, an individual from the population. This individual is selected as a parent and the process repeats until a sufficient number of parents have been selected.

The roulette wheel of fit favors individuals with higher fitness values while still maintaining diversity. Because the selection is random, it is possible for the poor designs to be selected as a parents, but this is less likely. The activation functions adjust the wheel by manipulating the fitness values.

#### 4.1.7 Crossover

The process of combining parents to form a child is called crossover. This combination should evolve the population towards more fit individuals. The children's properties are a combination of the parents' properties; the specific relationship is dependent on the property type.

For continuous variables, the combination of parent properties is done by randomly selecting a value between the parents' values. This is shown in Eq. (4.7).

$$X_{child} = p(X_{parent,max} - X_{parent,min}) + X_{parent,min} \quad (4.7)$$

where  $p$  is a random value between 0 and 1,  $X_{child}$  is the child's property value,  $X_{parent,max}$  is the greater of the two parents' property values, and  $X_{parent,min}$  is the lesser of the two parents' property values. This combination of parent properties ensures that the child is similar to its parents but maintains its ability to improve the fitness.

The process of combining parents' path values is similar to that of the continuous variables. A random path value is selected within the range of the parents' values. The

only difference between the path value crossover and the continuous variable crossover is that the path value is a discrete integer value.

Path type crossover is different from the other properties because it cannot be represented numerically. If both parents have the same path type, that path type is passed to the child. Otherwise, the child is given a new random path type.

The crossover implementation is highly dependent on the specific genetic algorithm that is being developed. Other crossover functions were considered, such as allowing children to have larger or smaller values than parents or fitting normal distributions to the parents' values, but this simple implementation consistently produced strong results that converged to the same values, regardless of seeding or activation functions. The validity of the crossover method is shown in Section 4.2.

#### 4.1.8 Mutation

The final step in each iteration of the genetic algorithm is mutation. Mutation is the process of randomly changing certain individuals' property values in order to maintain diversity in the population. The number of individuals to mutate is determined by the mutation parameter, which is usually small. The algorithm loops through each individual and checks if a random number is less than the mutation parameter to see if that individual should be mutated. The best performing individual is excluded from this process to prevent a decrease in best fit.

If an individual is selected for mutation, one of their four properties are randomly chosen to be mutated. Mutating a property is simply assigning it a new random value. For velocity, diameter, and path value, this is selecting a random value between the maximum and minimum. For path type, this is selecting a new random path type.

As is discussed in Section 4.2, a mutation parameter of 0.1 encouraged a full exploration of the design space while still converging in a reasonable amount of time. As with crossover, the implementation of the mutation step is highly dependent on the genetic algorithm. Section 4.2 proves that this method maintains diversity and helps the population evolve to the optimal design.

#### 4.1.9 Termination

The genetic algorithm repeats the process of fitness evaluation, crossover, and mutation for a set number of iterations before terminating. During these iterations, the population evolves to better designs before reaching a limit and converging to the same case. The best design generally takes steps in rising fitness and does not converge to the optimal solution gradually. The average fitness of the population, however, converges to the best case gradually over the course of the algorithm. Because there is no way to know if the best case is truly optimal during the process, the algorithm runs for a set number of iterations, determined beforehand as an input.

To validate that the final best case is truly optimal, two indicators are considered. First, the trend of the average fitness is compared with the best fitness over the course of the algorithm. The average fitness should asymptotically reach the best fitness to prove that all individuals are converging to the same case. Second, the algorithm is run with multiple seeding types and activation functions, as discussed in Sections 4.1.2 and 4.1.4, respectively. The average fitness and best fitness are again analyzed, to ensure that every case reaches the same final value for each design variable.

The results of this optimization algorithm are discussed in Section 4.2, including a discussion on the validity of the crossover and mutation implementations, as well as the termination criteria.

## **4.2 Optimization Results**

The genetic algorithm was run in parallel on a 28 core Linux computing cluster. Multiple optimizations were run, with varying properties to ensure that the performance was consistent and reliable. The average algorithm took 3 hours to complete, given the parameters described in this chapter. The sCO<sub>2</sub> was modeled with temperature-dependent properties, unlike the models used to compare with CFD. A full discussion of sCO<sub>2</sub> material properties is found in Appendix A.

### 4.2.1 Design Space

Every optimization was run with the same model parameters as the CFD validation models. These parameters are described in detail in Chapter 2 and are included here in Table 4.1.

**Table 4.1 Optimization Model Parameters**

<b>Parameter</b>	<b>Value</b>
Solar Flux (W/m <sup>2</sup> )	1,000,000
Receiver Dimensions (L x W x H) (m)	0.02 x 0.02 x 0.01
Turn Radius (m)	0.001
Inlet Temperature (K)	873
Inlet Pressure (MPa)	20
Emissivity	0.88
Absorptivity	0.94
Receiver Thermal Conductivity (W / m <sup>2</sup> K)	27.5
External Convective Heat Transfer Coefficient (W / m <sup>2</sup> K)	8

The parameters not included in Table 4.1 (diameter, inlet velocity, path architecture) are the design variables optimized by the algorithm.

#### 4.2.2 Design Variable Bounds

The genetic algorithm optimizes the design variables within preset bounds. These bounds are either arbitrarily set or are a product of the discretized path design space. The path architecture is composed of a grid of discrete points, the size of which is constant and determined prior to the algorithm start. All of the results discussed in this section used a grid size of 31 x 31 cells, which is similar in size to the models replicated in CFD.

The diameter is constrained to ensure that parallel channels do not overlap, and that all automated designs can exist physically. To ensure this, the diameter is limited to 1.75x the size of a grid cell, which translates to a max diameter of 1.129 mm. The

velocity is arbitrarily constrained to be within 1 to 80 m/s. As is shown from the optimization results, the optimal velocity falls within this range.

The path value is constrained by the size of the grid. For this case, the max possible path value is 3, due to the size limitations of the bifurcating path type. As with velocity, the optimal case falls within this range. The design variable constraints are listed in Table 4.2.

**Table 4.2 Design Variable Bounds**

Variable	Minimum Value	Maximum Value
Diameter (m)	0.0003	0.001129
Inlet Velocity (m/s)	1	80
Path Value	1	3

#### 4.2.3 Algorithm Parameters

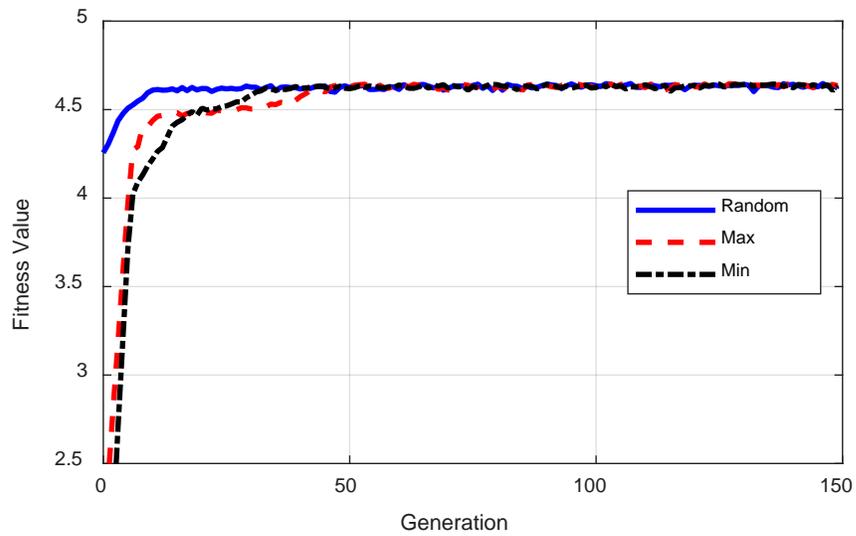
There are multiple parameters that control the genetic algorithm. These are the number of iterations, population size, crossover parameter, and mutation parameter. The value for each of these parameters is highly dependent on the specific genetic algorithm and are usually found through trial and error.

For this implementation, the number of iterations was set to 150, which is sufficient to converge any case, regardless of seeding or activation function. The population size was set to 50, which also converged quickly while still exploring the design space. The crossover parameter used was 0.4 and the mutation parameter was 0.1.

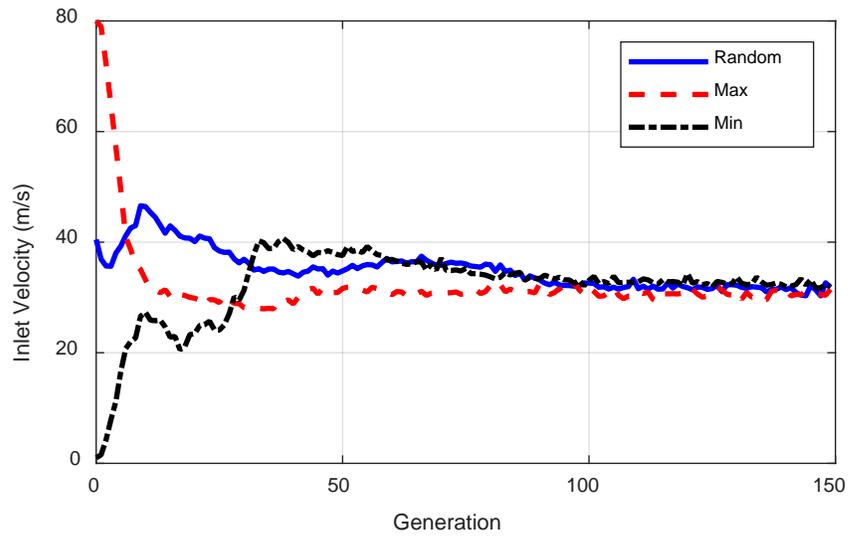
Each of the parameter values were chosen through trial and error from a collection of optimization runs. The agreement between every case, with any seeding and activation function, prove that the parameters are valid.

#### 4.2.4 Seeding Analysis

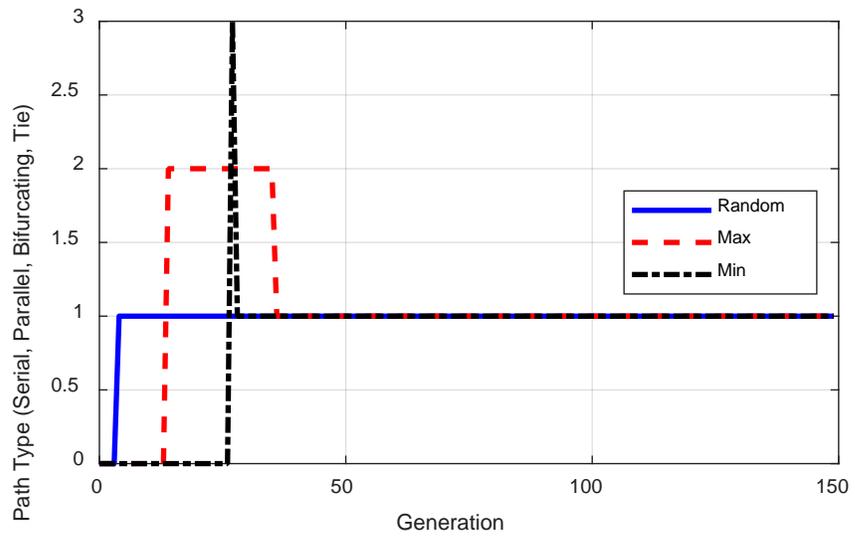
The population seeding was varied as discussed in Section 4.1.2. The purpose of manipulating the initial population seeding is to prove that the algorithm is converging to global optimums, and to ensure that the algorithm parameters were correctly chosen. The algorithm was run with the three different seeding types (random, max, and min) and the results are shown in Figure 4.4.



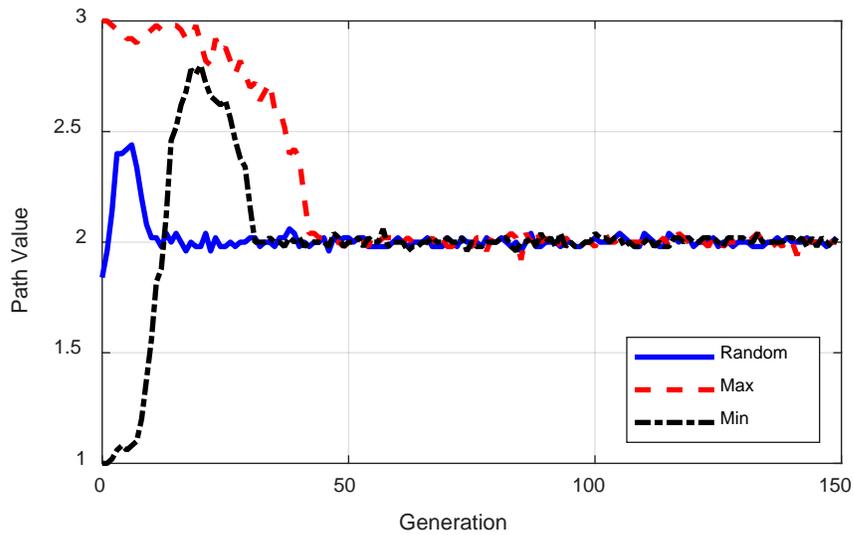
a)



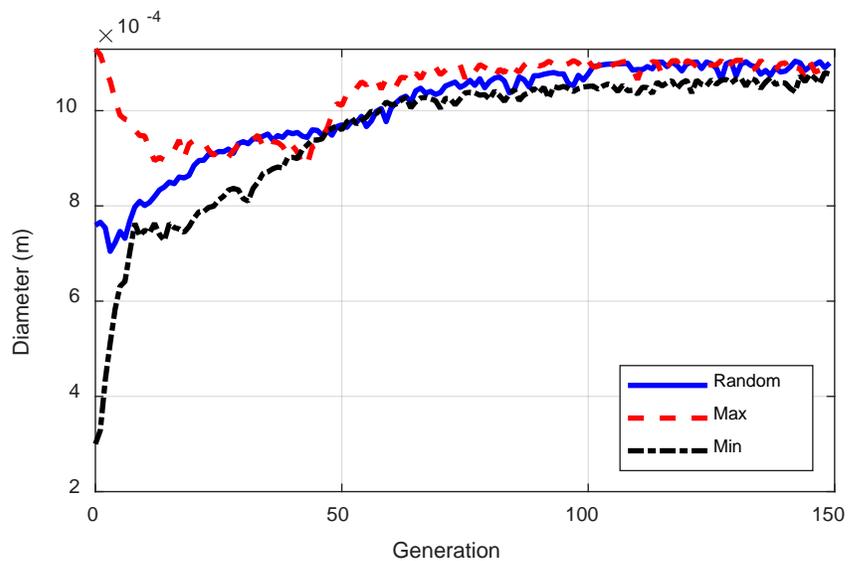
b)



c)



d)



e)

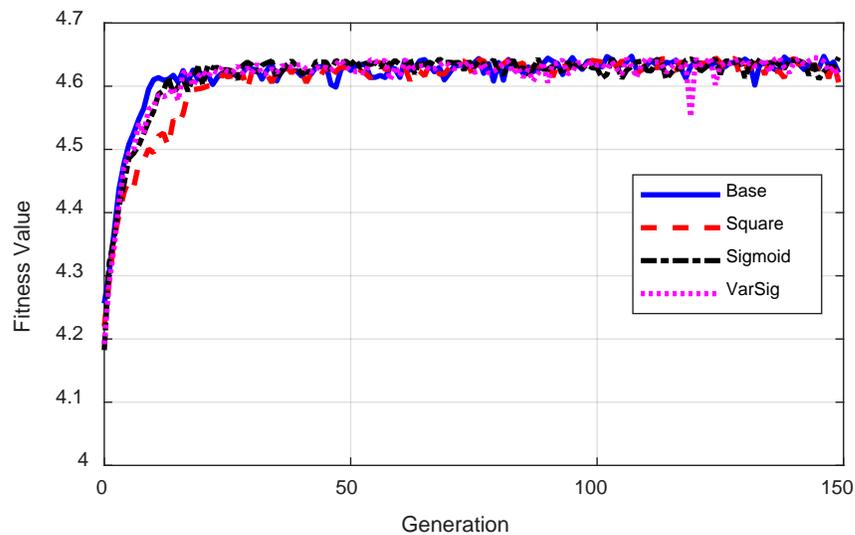
**Figure 4.4 Fitness and Design Variable Evolution for Various Seeding Types**

Figure 4.4 shows how the average value of each design variable and the fitness evolves over the course of the algorithm for each seeding type. The purpose of showing the average value, rather than the best value, is to compare how the different cases explore the design space and to check if they converge to the same values. As is shown,

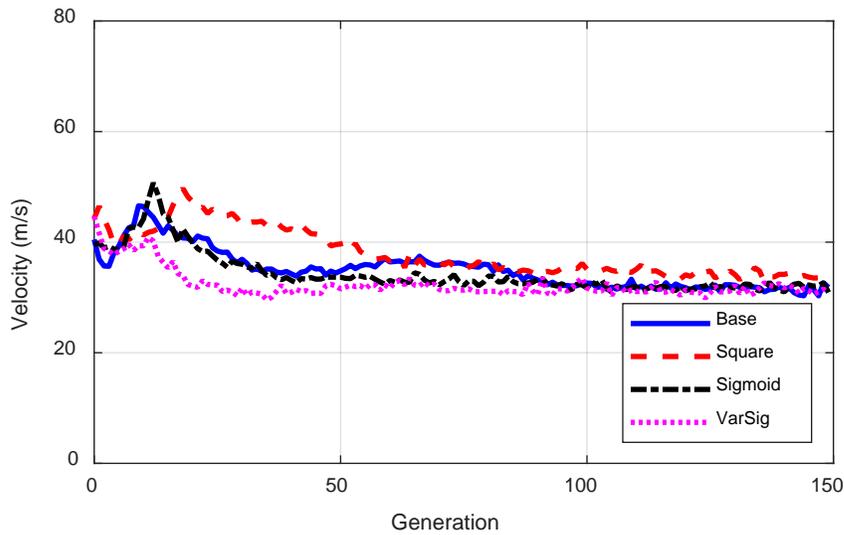
although each case starts at different averages, every case converges to the same values over the course of the algorithm. This is evidence that the algorithm is reaching global optimums and that 150 generations is sufficient for convergence.

#### 4.2.5 Activation Function Analysis

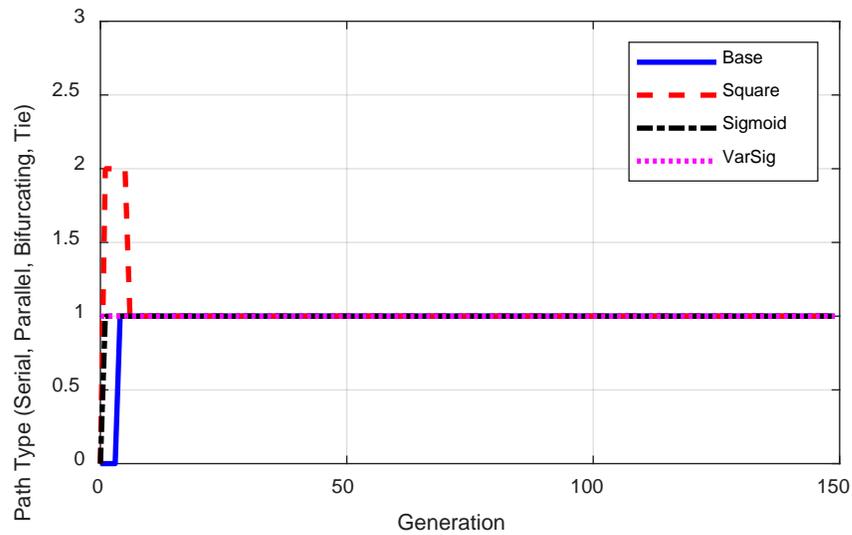
Similar to the initial population seeding, multiple activation functions were studied to confirm the validity of the algorithm. The functions are discussed in detail in Section 4.1.4. The activation functions manipulate the fitness value to prevent premature convergence and encourage full exploration of the design space. The four activation functions considered are: base (no activation function), square, sigmoid, and variable sigmoid. The algorithm was run with each of these activation functions and the results are shown in Figure 4.5.



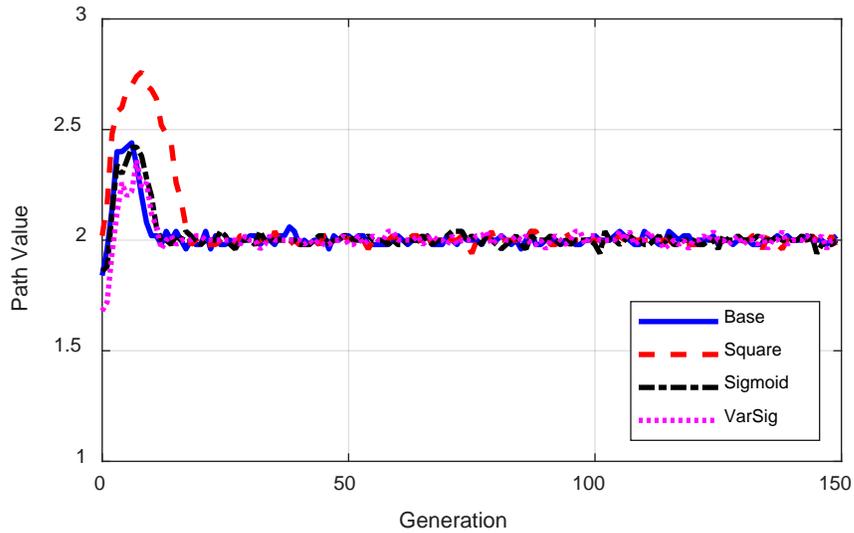
a)



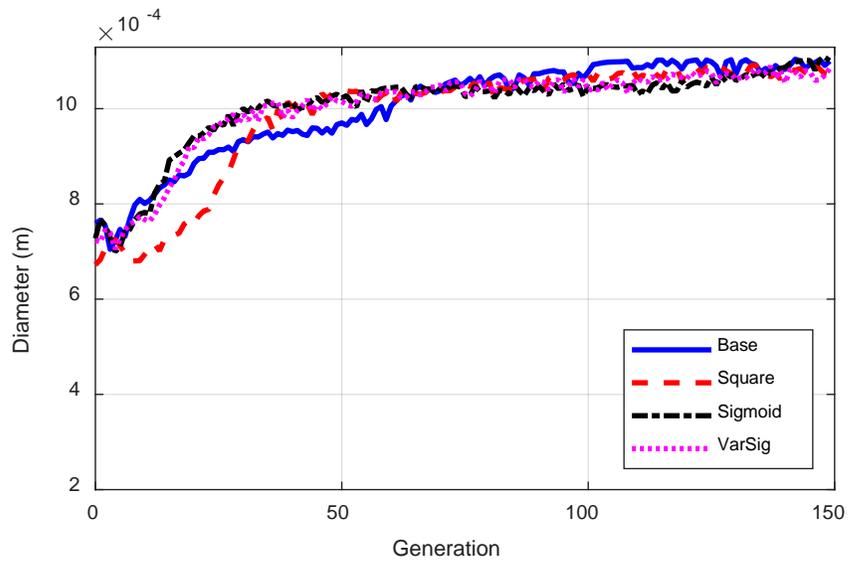
b)



c)



d)



e)

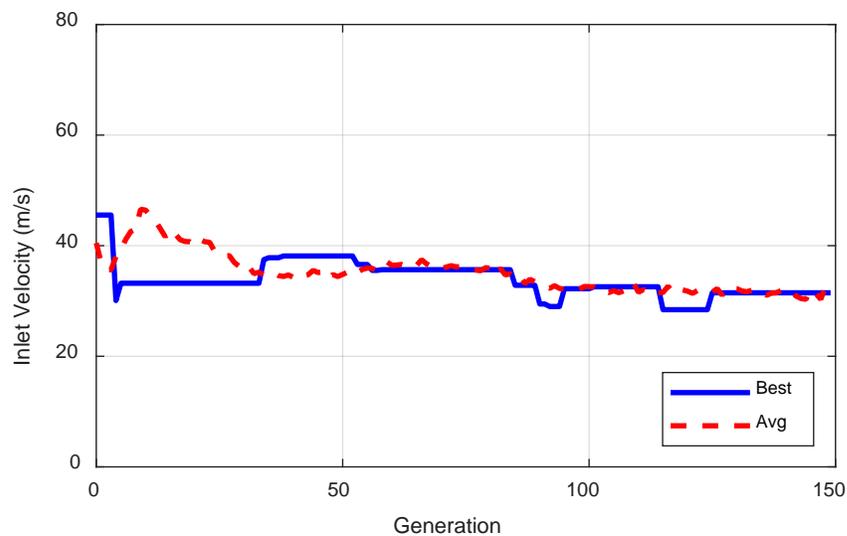
**Figure 4.5 Fitness and Design Variable Evolution for Various Activation Functions**

Figure 4.5 shows a strong agreement between each of the four activation functions. The average value for each design variable converges to the same value for each activation function case. As with the seeding analysis, the activation functions prove that the algorithm is reaching a global optimum during the 150 iterations.

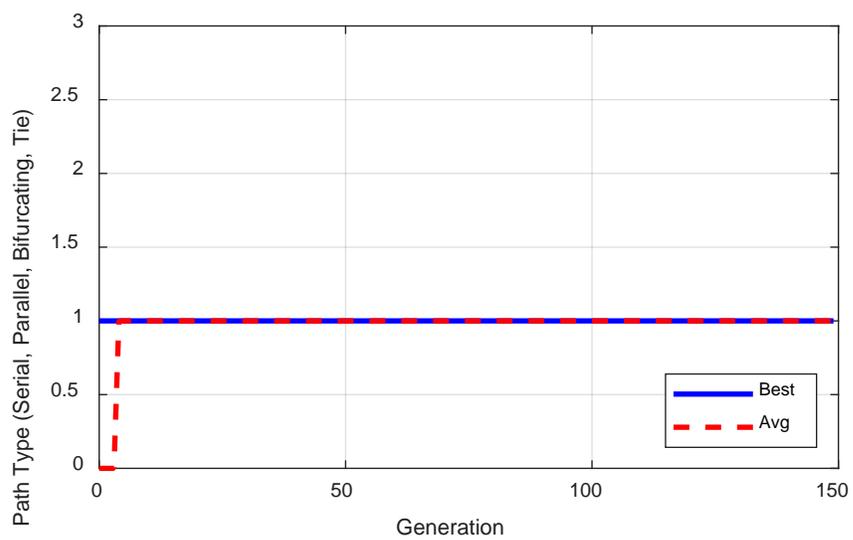
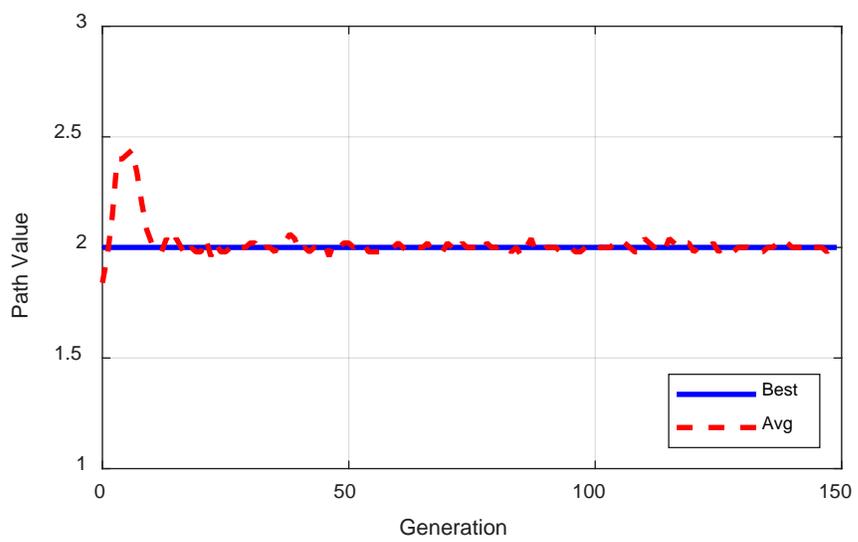
#### 4.2.6 Optimal Design

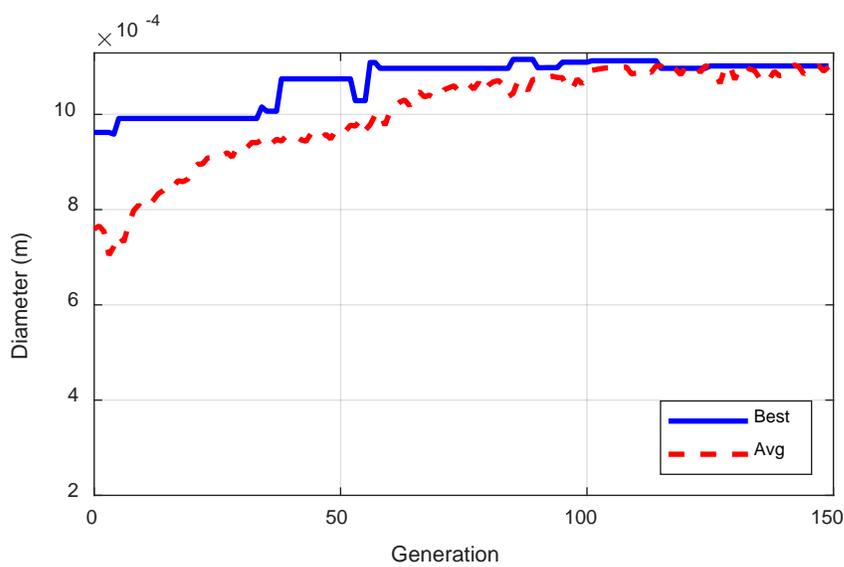
The seeding and activation function analyses prove that the genetic algorithm is operating correctly for the given design variables and constraints. The results in this section are from the base case, which is random seeding and no activation function. As was shown in the previous sections, the base case performs in the same manner as a case with any other seeding or activation function.

Figure 4.6 shows how each design variable evolves over the course of the genetic algorithm, showing the best performing individual and the average of every individual.



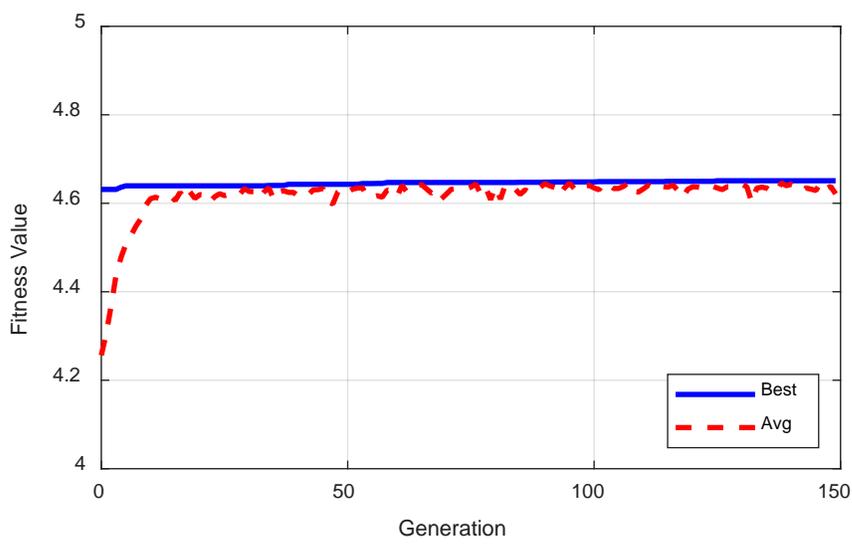
a)

**b)****c)**

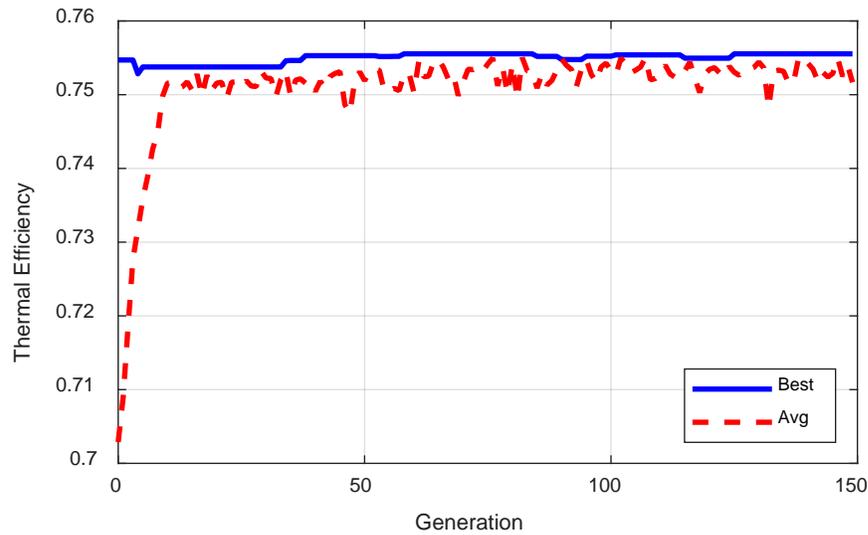


**d)**  
**Figure 4.6 Optimal Case Design Variable Evolution of a) Velocity b) Path Type c) Path Value and d) Diameter**

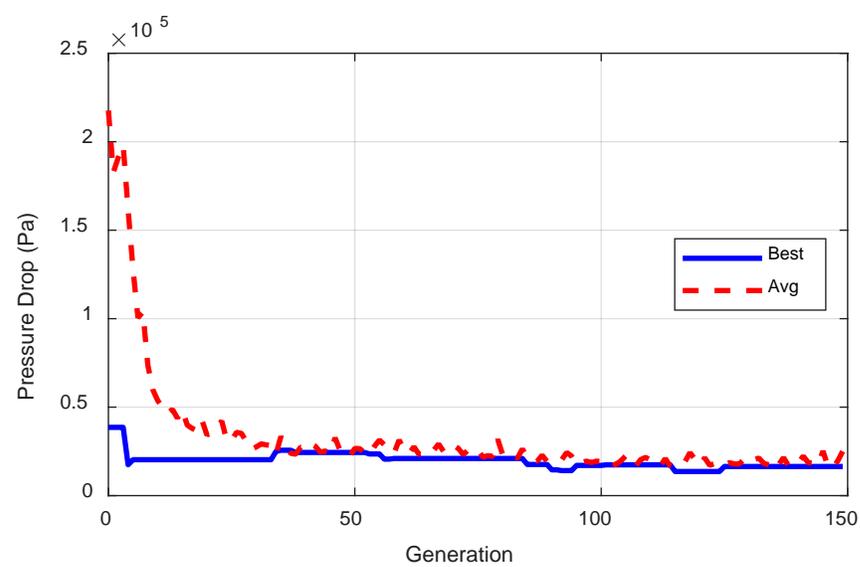
Figure 4.7 shows how key model outputs progress during the algorithm.



**a)**



b)



c)

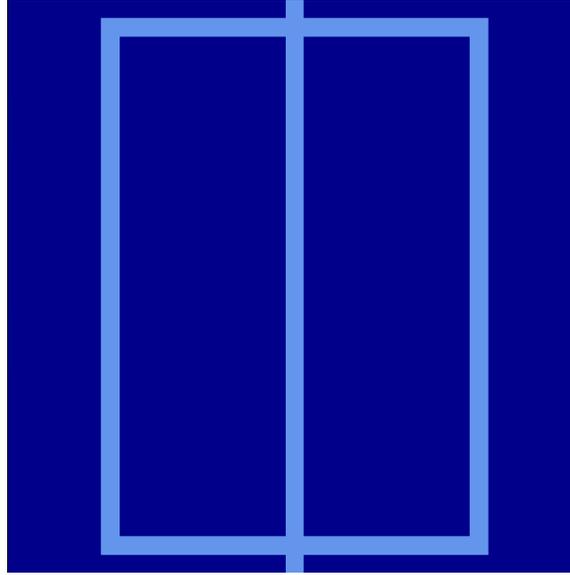
**Figure 4.7 Model Performance Evolution for the Optimal Case**

The algorithm progression shows that the design variables evolve to maximize thermal efficiency and minimize the pressure drop. As outlined in Chapter 2, the fluid velocity is linearly related to the Nusselt number for turbulent flow, which increases with the velocity. Larger Nusselt numbers result in better heat transfer to the fluid, which

increases the thermal efficiency of the receiver. This explains why the average velocity rises during the min seeding case. The velocity does not reach its maximum bound, however, due to its relationship with pressure drop. From Chapter 3, major and minor losses are quadratically dependent on the velocity. As the velocity rises, eventually the gain in thermal efficiency is outweighed by the rising pressure drop.

Diameter is maximized by the optimization. This occurs because the thermal efficiency and pressure drop improve with larger diameters, relative to the mass flow rate. Similar to velocity, and as detailed in Chapter 2, the Nusselt number increases with diameter. This causes the thermal efficiency to rise as the diameter increases. Unlike velocity, however, the pressure drop is inversely related to diameter, causing the relative pressure drop to decrease as the diameter increases. For these reasons, the diameter is maximized by both the thermal efficiency and the pressure drop.

The thermal efficiency improves as the total channel surface area increases. This is supported by the diameter being maximized in the optimization. The increase in thermal efficiency, however, is offset by a resulting increase in pressure drop. For this reason, the optimal design does not maximize path value, or result in the design with the maximum possible channel surface area. The path type and path value variables quickly converge to a parallel path with a path value of 2. This translates to the design shown in Figure 4.8.



**Figure 4.8 Optimal Case Pathway Design**

This simple parallel path has a low pressure drop and a high relative thermal efficiency. The full case design variables are shown in Table 4.3 and the outputs are shown in Table 4.4.

**Table 4.3 Optimal Case Design Variables**

<b>Parameter</b>	<b>Value</b>
Velocity (m/s)	31.48
Path Type	Parallel
Path Value	2
Diameter (m)	0.0011

**Table 4.4 Optimal Case Results**

<b>Output</b>	<b>Value</b>
Fitness	4.651
Thermal Efficiency (%)	75.55
Pressure Drop (Pa)	16,457
Nominal Strain (%)	0.029

The thermal efficiency quickly reaches an asymptote near 76%. To better understand the relationship between the cost function (fitness) and the thermal efficiency, an algorithm was run that optimized for only the thermal efficiency. The purpose of removing the pressure drop consideration was to find the maximum possible thermal efficiency for the given constraints. The bifurcating path type placed limitations on the maximum path value, so it was removed for this run. The maximum path value without the bifurcating path type is 4, instead of 3. Table 4.5 shows the resulting optimal design variables from this run, and Table 4.6 shows how the optimal design performs.

**Table 4.5 Optimal Thermal Efficiency Case Design Variables**

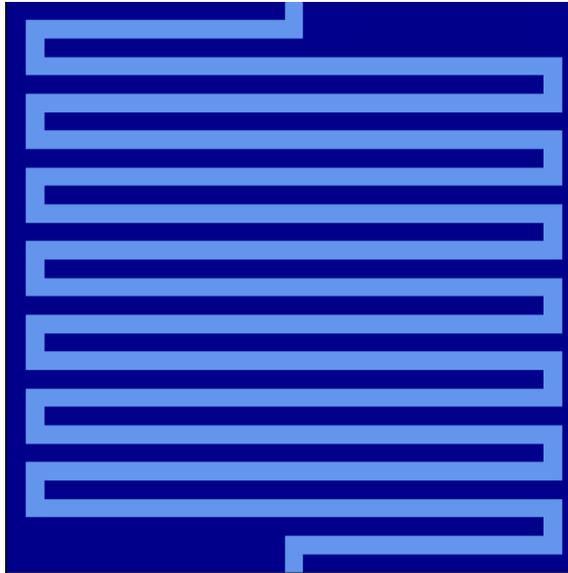
<b>Parameter</b>	<b>Value</b>
Velocity (m/s)	74.0
Path Type	Parallel
Path Value	2
Diameter (m)	0.0011

**Table 4.6 Optimal Thermal Efficiency Case Results**

<b>Output</b>	<b>Value</b>
Fitness	4.56
Thermal Efficiency (%)	75.79
Pressure Drop (Pa)	90,428
Nominal Strain (%)	0.028

Optimizing for thermal efficiency performed as expected; thermal efficiency increased but the resulting pressure drop was significantly larger causing the fitness value to decrease. This shows that the algorithm is weighing the efficiency with the pressure drop, and appropriately choosing designs that optimize for both.

The genetic algorithm is constrained to a limited number of path types and path values. To better understand these constraints, a design was generated and tested that was not in the design space of the optimization scheme. This design is shown in Figure 4.9 and is a serial path with 15 horizontal segments.



**Figure 4.9 Custom High Efficiency Path Design**

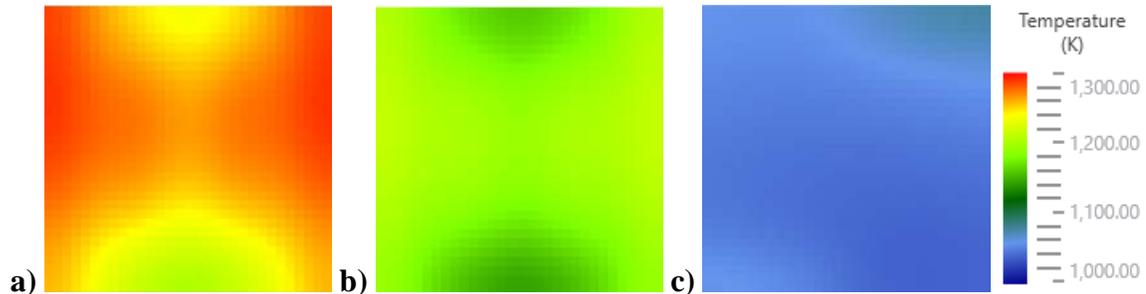
The path was modeled with the same parameters as the optimal thermal efficiency design detailed in Table 4.3. The results of this custom design are shown in Table 4.7.

**Table 4.7 Custom High Efficiency Case Results**

<b>Output</b>	<b>Value</b>
Fitness	-0.199
Thermal Efficiency (%)	76.48
Pressure Drop (Pa)	3,873,800
Nominal Strain (%)	0.338

The custom design produces a higher thermal efficiency than any design possible in the genetic algorithm. This shows the limits of the optimization scheme, although the design is far from optimal as the large pressure drop results in a negative fitness value. The existence of higher efficiency designs, however, is indicative of more optimal designs existing outside of the algorithm's design space. The limitations of the algorithm and possible solutions are discussed in Section 4.2.8.

To better understand the performance of the three designs discussed in this section, the surface temperature contours are plotted in Figure 4.9.



**Figure 4.10** Surface Temperatures for a) Optimal Design b) Optimal Thermal Efficiency Design and c) Custom High Efficiency Design

As seen in Figure 4.10, the design with the highest thermal efficiency, Figure 4.10c, results in the lowest temperatures on the receiver surface and the smallest difference in temperatures on the surface. The decreased surface temperatures are due to the working fluid absorbing more energy than the other designs and the decreased difference in temperatures shows that the heat is better spread out through the receiver. Future work on the thermal stress calculations and constraints could use the temperature difference on the surface as a part of the optimization objective function, as larger differences in temperatures on the surface could be mechanically less optimal.

#### 4.2.7 Strain and Pressure Constraints

The composite receiver can physically withstand a certain amount of strain. This limit is being researched currently, with a focus on thermal strains and the strain due to the pressurized working fluid. The nominal strain is the first step in analyzing the mechanical durability of the receiver and this calculation is discussed Section 4.1.5. An algorithm was run that limited the nominal strain to 0.05%, which is a conservative maximum allowable value.

The run with the strain constraint resulted in nearly the same optimal values as the base case discussed in Section 4.2.6. This is not surprising because the optimal case from the previous section had a nominal strain of only 0.029%. With the strain constraint, the optimal velocity was 32.26 m/s, the diameter was 0.0011 m, and the path architecture was the same.

Another optional constraint for the algorithm is the pressure constraint. The fitness function already weighs the pressure drop but it could be possible that any pressure drop over a certain value makes the design immediately infeasible. A hard constraint was placed on the designs so any case that passed this value was removed from the population. The maximum allowable pressure drop was set to 34 kPa, which was a limit chosen at the outset of the project.

The optimal base case without the pressure constraint had a pressure drop of less than 17 kPa. Similar to the strain constraint, the optimization with the pressure constraint performed in the same manner as the base case with no constraint. The optimal case with the pressure constraint had a velocity of 28.1 m/s, a diameter of 0.0011 m, and the same path architecture.

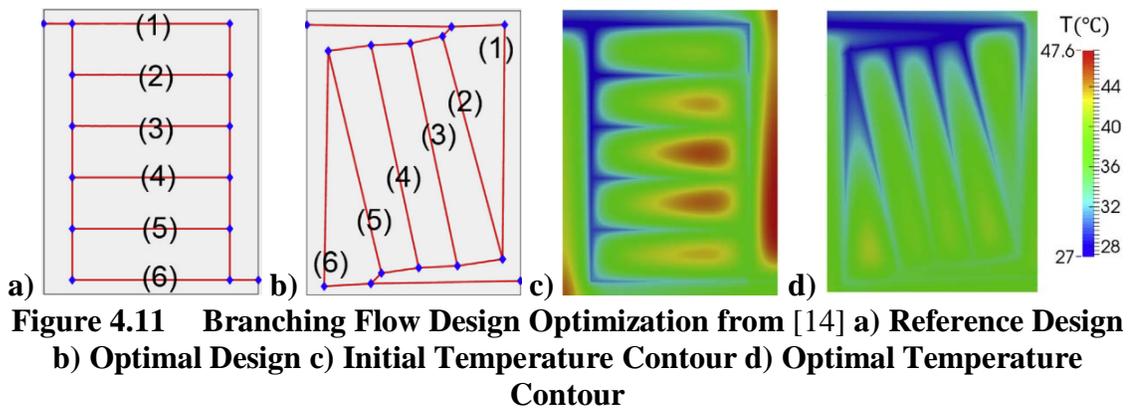
The strain and pressure constraints show that the current optimal design is well within the design requirements. The genetic algorithm successfully constrained the designs, and future work could add new constraints, such as maximum thermal strain or surface temperature gradients.

#### 4.2.8 Optimization Results Discussion

The genetic algorithm performed as expected given the constraints on the model. Each design variable showed consistent behavior, regardless of seeding or activation

function implementations. The fitness function also performed as expected, favoring the thermal efficiency but preventing the pressure drop loss from outweighing the thermal efficiency gains.

The low relative thermal efficiency of the optimal design can be attributed to the thermal model constraints and the limitations of the genetic algorithm. The thermal model is built on a grid of points. Each point can be connected only to neighboring points, but not in the diagonal direction. This means that the resistance network can only model paths with  $90^\circ$  turns or connections. The work by Tan et al. [14], mentioned in Chapter 1, showed that optimal designs do not have paths structured on this form of grid. The optimal designs, shown again here in Figure 4.11, have diagonal parallel paths. This path type cannot be replicated currently with the thermal and hydraulic models.



The grid-based structure of the thermal model also limits the design space the genetic algorithm explores. The CFD comparison designs were modeled on a  $19 \times 21$  sized grid. For this reason, the genetic algorithm operated on a similarly sized  $31 \times 31$  grid. This grid size limited the maximum path value to only 3. Larger grid sizes would allow the genetic algorithm to explore designs with more parallel paths, for instance, but

these designs have not been validated against CFD. This conservative approach to the optimization space likely contributed to poor performing designs.

Finally, the parameterization of the path architecture limited the scope of the optimization scheme. The path architecture is parameterized to two variables, which greatly limits the design space. Allowing the genetic algorithm full control of the path architecture, similar to the work by Dokken and Fronk [17], is not possible with the current implementation of the thermal model. The restraints imposed by the thermal model require significant oversight in the design of the path architectures, limiting the genetic algorithm's ability to find optimal designs.

Future work that allows the algorithm more design freedom is discussed in Chapter 5. The algorithm successfully optimizes within the constraints, even if the constraints limit the overall success of the optimal design.

## CHAPTER FIVE: CONCLUSION

### 5.1 Conclusion

A novel carbon-carbon composite gas-phase solar receiver has been modeled in this work. The receiver uses sCO<sub>2</sub> as the working fluid, increasing the operational temperature of traditional CSP receivers. The carbon composite is mechanically durable at high temperatures and able to support the high pressures of sCO<sub>2</sub> due to the unit cell numbering up approach. The receiver is composed of a collection of small unit cells connected in parallel.

A thermal and hydraulic model have been developed to simulate the performance of a receiver unit cell. The thermal model is built as a thermal resistance network that quickly solves the temperatures in the fluid and receiver body at steady state. The pressure model calculates the major and minor pressure losses along the fluid path to estimate the pressure drop over the unit cell. Both models are dynamically formed to simulate any path architecture.

The thermal and hydraulic models have been compared with extensive CFD simulations for validation as experimental testing data is not yet available. Three path architectures were considered to capture the scope of the design space: straight, serial, and parallel. A wide range of inlet velocities from 1 m/s to 100 m/s were simulated. The thermal model shows a strong agreement with CFD when comparing the outlet temperatures, with the error magnitude decreasing as the velocity increases. The pressure model also performs well when comparing the pressure losses.

The unit cell has been optimized with a genetic algorithm that considered four design variables: inlet velocity, channel diameter, path type, and path value. Path type and path value parameterize the path architecture into two variables. The algorithm was run with multiple seeding types and activation functions to ensure that the global optimum was found. The algorithm consistently produced the same design, a parallel path with three channels, that maximized the diameter and chose a velocity that limited the pressure drop while maintaining a high relative thermal efficiency. The strain and pressure drop were both constrained and the optimal design fit within these requirements.

## **5.2 Future Work and Recommendations**

The thermal and hydraulic models can be improved with further CFD analysis and experimental testing. The pressure drop calculation for crosses can be improved, either with correlations from CFD results or with special experimental testing. An area that needs to be validated is the material properties of sCO<sub>2</sub> and the carbon composite. As discussed in Chapter 2, the CFD models assumed constant sCO<sub>2</sub> properties to limit the complexity of the model and allow it to be solvable. The model developed in this work is capable of handling temperature-dependent properties, but this has not been validated. The temperature-dependent properties are discussed in more detail in Appendix A.

Expanding the design capabilities of the thermal and hydraulic models is also recommended. Currently, the thermal resistance network can only support path architectures with a constant diameter. Optimal designs, particularly designs with branching flow, would likely benefit from a large inlet header and smaller branching channels. The thermal resistance network is capable of handling varying diameters, but the pressure calculations for changing diameters have not been implemented.

Along with support for varying diameters, the model would benefit from allowing diagonal channels, rather than requiring channels to stay on a grid with 90° connections. The potential benefit of diagonal channels is discussed in Section 4.2.8, which relates the discussion to the work of Tan et al. [14]. Similarly, curved paths would further expand the design space.

The genetic algorithm performed as expected but can be improved by reducing the number of constraints. The biggest constraint on the algorithm was the path parameterization. By reducing the path architecture to two variables, the capabilities of the algorithm to optimize were greatly reduced. Using more variables to describe the path could expand the design space of the optimization, which would result in better performing designs.

Finally, an assumption of this work was each unit cell would have a single inlet and a single outlet. Introducing multiple inlets and outlets could prove to be beneficial and would give the genetic algorithm more freedom. Alternating flow directions could reduce the temperature difference on the receiver surface and more adequately transfer energy to the working fluid.

## REFERENCES

- [1] EIA, 2019, “What Is U.S. Electricity Generation by Energy Source?,” U.S. Energy Inf. Adm. [Online]. Available: <https://www.eia.gov/tools/faqs/faq.php?id=427&t=3>.
- [2] NREL, 2012, “Renewable Electricity Futures Study: Executive Summary,” Natl. Renew. Energy Lab., **1**, pp. 1–43.
- [3] U.S. DOE, 2017, “The SunShot 2030 Goals: 3¢ per Kilowatt Hour for PV and 5 ¢ per Killowatt Hour for Dispatchable CSP,” pp. 2–6.
- [4] Baharoon, D. A., Rahman, H. A., Omar, W. Z. W., and Fadhl, S. O., 2015, “Historical Development of Concentrating Solar Power Technologies to Generate Clean Electricity Efficiently – A Review,” *Renew. Sustain. Energy Rev.*, **41**, pp. 996–1027.
- [5] Islam, M. T., Huda, N., Abdullah, A. B., and Saidur, R., 2018, “A Comprehensive Review of State-of-the-Art Concentrating Solar Power (CSP) Technologies: Current Status and Research Trends,” *Renew. Sustain. Energy Rev.*, **91**(April), pp. 987–1018.
- [6] Mehos, M., Turchi, C., Vidal, J., Wagner, M., Ma, Z., Ho, C., Kolb, W., Andraka, C., and Kruizenga, A., 2017, “Concentrating Solar Power Gen3 Demonstration Roadmap,” Nrel/Tp-5500-67464, (January), pp. 1–140.
- [7] Rodríguez-Sánchez, M. R., Sánchez-González, A., Marugán-Cruz, C., and Santana, D., 2014, “New Designs of Molten-Salt Tubular-Receiver for Solar Power Tower,” *Energy Procedia*, **49**, pp. 504–513.
- [8] Ho, C. K., and Iverson, B. D., 2014, “Review of High-Temperature Central Receiver Designs for Concentrating Solar Power,” *Renew. Sustain. Energy Rev.*, **29**(1), pp. 835–846.

- [9] Luu, M. T., Milani, D., McNaughton, R., and Abbas, A., 2017, “Dynamic Modelling and Start-up Operation of a Solar-Assisted Recompression Supercritical CO<sub>2</sub> Brayton Power Cycle,” *Appl. Energy*, **199**, pp. 247–263.
- [10] Zada, K. R., Hyder, M. B., Kevin Drost, M., and Fronk, B. M., 2016, “Numbering-Up of Microscale Devices for Megawatt-Scale Supercritical Carbon Dioxide Concentrating Solar Power Receivers,” *J. Sol. Energy Eng. Trans. ASME*, **138**(6), pp. 1–9.
- [11] Hyder, M. B., 2017, “Numbering-Up of Parallel Microchannel Arrays to Megawatt Scale Supercritical CO<sub>2</sub> Solar Thermal Receivers,” M.S. thesis, Oregon State University.
- [12] Besarati, S. M., Goswami, D. Y., and Stefanakos, E. K., 2015, “Development of a Solar Receiver Based on Compact Heat Exchanger Technology for Supercritical Carbon Dioxide Power Cycles,” *J. Sol. Energy Eng. Trans. ASME*, **137**(3), pp. 1–8.
- [13] Lei, N., 2006, “The Thermal Characteristics of Multilayer Minichannel Heat Sinks in Single-Phase and Two-Phase Flow,” Ph.D. dissertation, The University of Arizona.
- [14] Tan, M. H. Y., Najafi, A. R., Pety, S. J., White, S. R., and Geubelle, P. H., 2016, “Gradient-Based Design of Actively-Cooled Microvascular Composite Panels,” *Int. J. Heat Mass Transf.*, **103**, pp. 594–606.
- [15] Abdoli, A., and Dulikravich, G. S., 2014, “Multi-Objective Design Optimization of Branching, Multifloor, Counterflow Microheat Exchangers,” *J. Heat Transfer*, **136**(10).
- [16] Aragón, A. M., Wayer, J. K., Geubelle, P. H., Goldberg, D. E., and White, S. R., 2008, “Design of Microvascular Flow Networks Using Multi-Objective Genetic Algorithms,” *Comput. Methods Appl. Mech. Eng.*, **197**(49–50), pp. 4399–4410.
- [17] Dokken, C. B., and Fronk, B. M., 2018, “Optimization of 3D Printed Liquid Cooled Heat Sink Designs Using a Micro-Genetic Algorithm with Bit Array Representation,” *Appl. Therm. Eng.*, **143**(January), pp. 316–325.

- [18] Windhorst, T., and Blount, G., 1997, "Carbon-Carbon Composites: A Summary of Recent Developments and Applications," *Mater. Des.*, **18**(1), pp. 11–15.
- [19] Sheehan, J. E., Buesking, K. W., and Sullivan, B. J., 1994, "CARBON-CARBON COMPOSITES," **24**, pp. 19–44.
- [20] Munson, B. R., Okiishi, T. H., Huebsch, W. W., and Rothmayer, A. P., 2013, *Fundamentals of Fluid Mechanics*, Wiley.
- [21] Jacimovic, B., Genic, S., and Lelea, D., 2018, "Calculation of the Heat Transfer Coefficient for Laminar Flow in Pipes in Practical Engineering Applications," *Heat Transf. Eng.*, **39**(20), pp. 1794–1800.
- [22] Gnielinski, V., 1976, "New Equations for Heat and Mass Transfer in Turbulent Pipe and Channel Flow," *Int. Chem. Eng.*, **16**(2), pp. 359–368.
- [23] Kakaç, S. (Sadık), Shah, R. K., and Aung, W., 1987, *Handbook of Single-Phase Convective Heat Transfer*, Wiley.
- [24] Math.NET, "Math.NET Numerics" [Online]. Available: <https://numerics.mathdotnet.com/>.
- [25] NVIDIA, "CuSOLVER" [Online]. Available: [docs.nvidia.com/cuda/cusolver/index.html](https://docs.nvidia.com/cuda/cusolver/index.html).
- [26] Vorst, H. A. van der, 1992, "Bi-CGStab: A Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, **13**(2), pp. 631–644.
- [27] Cheng, N.-S., 2008, "Formulas for Friction Factor in Transitional Regimes," *J. Hydraul. Eng.*, **134**(9), pp. 1357–1362.
- [28] Sharp, Z. B., Johnson, M. C., Barfuss, S. L., and Rahmeyer, W. J., 2010, "Energy Losses in Cross Junctions," *J. Hydraul. Eng.*, **136**(1), pp. 50–55.
- [29] Nikfetrat, K., Johnson, M. C., and Sharp, Z. B., 2015, "Computer Simulations Using Pattern Specific Loss Coefficients for Cross Junctions," *J. Hydraul. Eng.*, **141**(9), p. 04015018.

- [30] Cross, H., 1936, "Analysis of Flow in Networks of Conduits or Conductors," Eng. Exp. Stn., **34**, pp. 1–29.
- [31] Holland, J., 1975, "Adaptation in Natural and Artificial Systems : An Introductory Analysis with Application to Biology," Control Artif. Intell.
- [32] Victor Paul, P., Moganarangan, N., Kumar, S. S., Raju, R., Vengattaraman, T., and Dhavachelvan, P., 2015, "Performance Analyses over Population Seeding Techniques of the Permutation-Coded Genetic Algorithm: An Empirical Study Based on Traveling Salesman Problems," Appl. Soft Comput. J., **32**, pp. 383–402.
- [33] Tong, D. L., and Mintram, R., 2010, "Genetic Algorithm-Neural Network (GANN): A Study of Neural Network Activation Functions and Depth of Genetic Algorithm Search Applied to Feature Selection," Int. J. Mach. Learn. Cybern., **1**(1–4), pp. 75–87.
- [34] Herakovich, C. T., 2017, "Thin-Walled Pressure Vessels," *A Concise Introduction to Elastic Solids: An Overview of the Mechanics of Elastic Materials and Structures*, Springer International Publishing, Cham, pp. 77–81.
- [35] Lemmon, E. W., McLinden, M. O., and Friend, D. G., "Thermophysical Properties of Fluid Systems," *NIST Chemistry WebBook*, P.J. Linstrom, and W.G. Mallard, eds., National Institute of Standards and Technology, Gaithersburg MD.

APPENDIX A  
**sCO<sub>2</sub> Material Properties**

The thermal and hydraulic models rely on four sCO<sub>2</sub> material properties: specific heat, density, thermal conductivity, and viscosity. Each of these properties is dependent on the temperature and pressure of the fluid. The model was developed to assume a constant pressure, because the pressure drop in the receiver unit cell is insignificant relative to the overall pressure of the fluid. The temperature of the fluid, however, significantly changes as the fluid passes through the receiver.

To accurately model the sCO<sub>2</sub> properties, the thermal and hydraulic models are solved iteratively, with increasingly accurate temperature assumptions at each fluid node. First, the temperatures at each fluid node are assumed to be the inlet temperature. Next, the pressure is calculated, and the mass flow rate division is solved with the Hardy Cross method. The temperature network is then solved, and the resulting temperature values are compared with the assumed values. A new temperature assumption is made for every fluid node and the material properties are recalculated. The process repeats until the temperature assumptions converge. The property values are interpolated from data from the National Institute of Standards and Technology [35].

Ideally, the CFD models would operate in a similar manner, using temperature dependent properties. It is possible to define a material property with a polynomial in Fluent, but this caused the complexity of the model to increase to the point of making the model unsolvable in a realistic amount of time. Different approaches to the CFD modeling have been investigated, such as reducing the order of the fitted polynomials or using user defined functions. Unfortunately, a solution has not been found.

To compare the CFD simulations with the thermal and hydraulic model, constant properties were used. Because the average temperature of the fluid changes depending on the receiver parameters, such as diameter and inlet velocity, property values were selected at the inlet condition, 873 K and 20 MPa. These values are listed in Table A.1.

**Table A.1 sCO<sub>2</sub> Properties at 873 K and 20 MPa**

<b>Property</b>	<b>Value</b>
Specific Heat (J / kg K)	1249.63
Density (kg / m <sup>3</sup> )	116.76
Thermal Conductivity (W / m K)	0.065854
Viscosity (kg / m s)	3.89e-5

The CFD models serve to validate the heat transfer and pressure calculations. These validation efforts are still valid with constant properties because they are consistent between the model and CFD. The optimization process used dynamic properties to simulate the true performance of the receiver.