

AN ABSTRACT OF THE DISSERTATION OF

Douglas Benjamin Heymann for the degree of Doctor of Philosophy in
Mechanical Engineering presented on December 10, 2010.

Title: On the Optimization of Performance in Fractal-like Branching
Microchannel Heat Transfer Devices

Abstract approved: _____

Deborah V. Pence

Fractal-like branching microchannel flow networks have been found to improve wall temperature uniformity and decrease both pressure drop and flow power compared to arrays of straight microchannels. The present study seeks to maximize the benefits of fractal-like branching channels by means of a gradient-based optimization algorithm. The algorithm identifies the geometric parameters that yield the highest ratio of benefit to cost; the benefit being the heat transfer with cost being flow power. The stream-wise pressure and wall temperature distributions were determined numerically using one-dimensional models validated using experimental diagnostics and computational flow analyses. Pressure distributions were used to assess flow power, and wall temperature was used as an optimization constraint. Several geometric constraints were imposed during optimization to ensure sufficient bonding area for fabrication, to maximize

convective surface area and to prevent channel overlap. Three fractal-like devices were studied and optimized: a single-phase heat sink, a two-phase heat sink and a single-phase oil-to-water heat exchanger. The flow rate of the three devices was constrained using a maximum wall temperature constraint for the single-phase heat sink, the critical heat flux for the two-phase heat sink, and a fixed cold side mass flow rate and temperature rise in the case of the heat exchanger.

The optimized solutions were found to depend highly on both the geometric and flow constraints imposed. As the terminal channel width was reduced, flow power was reduced in both single-phase and two-phase heat sinks. However, the flow power was increased in the heat exchanger due to the constrained volumetric flow rate. In addition, results show that if a maximum wall temperature constraint is employed, single-phase flow outperforms two-phase flow in terms of reduction in flow power. However when constrained by a critical heat flux, two-phase networks outperform single-phase networks by up to 150%.

©Copyright by Douglas Benjamin Heymann
December 10, 2010
All Rights Reserved

On the Optimization of Performance in Fractal-like Branching
Microchannel Heat Transfer Devices

by

Douglas Benjamin Heymann

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 10, 2010
Commencement June 2011

Doctor of Philosophy dissertation of Douglas Benjamin Heymann presented on
December 10, 2010.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial and Manufacturing Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Douglas Benjamin Heymann, Author

ACKNOWLEDGEMENTS

I would first like to acknowledge and thank my Advisor, Deborah Pence, who has given me sage advice and gone beyond the call of duty when providing me with research and professional development opportunities. I would also like to thank Sourabh Apte, Ping Ge, Murty Kanury, Jim Liburdy, Vinod Narayanan, and Nancy Squires, all Faculty whom I have learned much from at OSU. I would also like to thank Brian Morgan and Nick Jankowski at the US Army Research Lab for the great graduate research internship opportunity. In addition, I want to thank Nancy Squires for her help in my first teaching assignment. Special thanks goes to all of the students in the Thermal Fluid Science group whom I have had the pleasure of sharing my research with, Ruander Cardenas, Dan Krebs, Younghoon Kwak, Saran Salakij, Mike Sabo and Christopher Stull. Special thanks goes to Mike Sabo and Christopher Stull for gathering the experimental heat exchanger data. Thank you to my committee, Sourabh Apte, Jim Liburdy, Vinod Narayanan, Deborah Pence and Nancy Squires as well as my Graduate Representative, Dr. Kathy Mullet. Finally and want to thank my parents, Martin Heymann and Barbara Heymann for their support of my education. Last, but not least, I would also like to thank my wife for her support and patience.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	2
2 Optimization of Single-Phase Fractal-like Branching Channel Heat Sinks	5
2.1 Introduction	5
2.2 Geometric parameters and constraints	9
2.3 Optimization Techniques	13
2.3.1 Gradient-Based Steepest Descent Search	13
2.3.2 Direct Numerical Search	15
2.3.3 Genetic Algorithm	16
2.4 Results	17
2.4.1 Comparison of Gradient-based, Genetic Algorithm and Direct Search Results	17
2.4.2 Gradient-Based Steepest Descent Results	20
2.4.3 Summary	40
2.5 Conclusions	41
3 Optimization of Two-Phase Fractal-like Branching Channel Heat Sinks	43
3.1 Introduction	43
3.2 Model	47
3.2.1 Fractal Geometry	47
3.2.2 Pressure and temperature	49
3.3 Optimization	51
3.3.1 Objective function	51
3.3.2 Constraints	52
3.3.3 Results and Discussion	65
3.4 Conclusions	72
4 Modeling a Single-Phase Fractal-like Branching Microchannel Heat Exchanger	74
4.1 Introduction	74
4.2 Model	76
4.2.1 Fractal Geometry	76

TABLE OF CONTENTS (Continued)

	<u>Page</u>
4.2.2 Pressure Drop	78
4.2.3 Energy Balance	79
4.2.4 Nusselt Number Determination	83
4.2.5 Fin Efficiency	87
4.3 Results	92
4.3.1 Algorithm validation	92
4.3.2 Experimental validation	94
4.3.3 Heat exchanger performance	97
4.3.4 Parametric variations subject to constrained mass flow rates	101
4.4 Summary and Conclusions	118
 Bibliography	 120
 Appendices	 126
A MATLAB Code	127
A.1 Single-Phase Heat Sink Optimization Code	127
A.1.1 One-Dimensional Single Phase Heat Sink Model	127
A.1.2 Optimization Algorithm	136
A.2 Two-Phase Heat Sink Optimization Code	144
A.2.1 Critical Heat Flux Estimation	144
A.2.2 Optimization	150
A.2.3 Constraints	156
A.2.4 Post Processing	157
A.3 Single-Phase Heat Exchanger Model Code	163
A.3.1 Model Function File	163
A.3.2 Parametric Search	172
A.3.3 Post Processing	176
B Fractal-like Heat Exchanger Parametric Analysis Figures	181

LIST OF FIGURES

Figure		Page
1.1	Representative flow networks with 16 branches emanating from the inlet plenum at the center of the disk, $m = 4$ branching levels and (a) decreasing channel length, and (b) increasing channel length . .	2
2.1	Schematic representation of branch level channel lengths and channel widths, and constraints on channel spacing	10
2.2	Representative division between valid and invalid geometries for $R = 20mm$, $w_m = 85\mu m$, $n_0 = 10$ and $m = 6$. Contours indicate flow power determined from the direct search	14
2.3	Minimized pressure drop versus m as a function of disk heat flux for $R = 20mm$ and $T_{w,max} = 70^\circ C$. Large circles denote optimal m values for each heat flux.	22
2.4	Minimized pressure drop flow network parameters, benefit-to-cost ratio and channel area versus m for $q'' = 1000W/cm^2$, $R = 20mm$, and $T_{w,max} = 70^\circ C$	24
2.5	Flow network parameters and benefit-to-cost ratio versus applied heat flux for $R = 20mm$, and $T_{w,max} = 70^\circ C$. Dashed lines illustrate similarities between pressure drop and flow power optimization results.	27
2.6	Minimized flow power versus m as a function of disk heat flux for $R = 20mm$, and $T_{w,max} = 70^\circ C$. Large circles denote optimal m values for each heat flux.	30
2.7	Minimized pressure drop and minimized flow power flow network parameters and performance coefficient versus R for $q'' = 200W/cm^2$, and $T_{w,max} = 70^\circ C$	32
2.8	Minimized pressure drop and minimized flow power flow network parameters and benefit-to-cost ratio versus $T_{w,max}$ for $q'' = 200W/cm^2$, and $R = 20mm$	34
2.9	Minimized flow power versus m as a function of minimal channel width constraint for $q'' = 200W/cm^2$, $R = 20mm$, and $T_{w,max} = 70^\circ C$.	36
2.10	Length scale ratio and total convective surface area versus m and a function of minimum terminal channel widths for $q'' = 200W/cm^2$, $R = 20mm$, and $T_{w,max} = 70^\circ C$	37

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
2.11	Minimized pressure drop and minimized flow power flow network parameters and performance coefficient versus $w_{m,min}$ for $q'' = 200W/cm^2$, $R = 20mm$, and $T_{w,max} = 70^\circ C$	39
3.1	Illustration off a fractal-like branching channel tree	46
3.2	Flow chart of the two-phase flow algorithm for solving Eqns. 3.5-3.9	48
3.3	Comparison of the two-phase and single-phase optimization results subject to a maximum wall temperature of $70^\circ C$, filled markers represent the two-phase algorithm and open markers with a dashed - - line represent the single-phase algorithm results	55
3.4	Illustration of liquid film and vapor core control volumes in a cross-sectional view of the channel used to formulate the critical heat flux model, the shaded region represents what the film in a rectangular channel may actually look like while the dashed line, - -, represents the approximation of the liquid-vapor boundary, flow is going through the page	57
3.5	Velocity profile as a function of streamwise channel distance for both the three-equation and five-equation ODE models for a circular tube 70 mm long, 500 μm in diameter, with a mass flux of 500 kg/m^2-s and R-134a as the working fluid	59
3.6	Liquid-vapor interface profile as a function of streamwise channel distance for both the three-equation and five-equation ODE models for a circular tube 70 mm long, 500 μm in diameter, with a mass flux of 500 kg/m^2-s and R-134a as the working fluid	60
3.7	Comparison of the numerical and analytical stability criteria for $w_m = 100\mu m$ and $H = 150\mu m$	63
3.8	Optimized flow power versus m for disk heat fluxes of 35–300 W/cm^2 , filled markers are two-phase flow networks subject to the stability constraint, open markers are two-phase flow networks <i>not</i> subjected to the stability constraint	65
3.9	Map of flow power versus β and γ with stability constraint for $w_m = 100 \mu m$, $H = 150\mu m$, $R = 20mm$, and $q'' = 200 W/cm^2$	66

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
3.10	Weber number generated from the optimized flow networks versus m , for $R = 20mm$, $H = 150\mu m$, $w_m = 50\mu m$, filled markers are flow networks subject to the stability constraint, open markers are networks not subject to the stability constraint	69
3.11	Evaporative to inertia force ratio, \bar{F}_{max} , generated from the optimized flow networks versus m , for $R = 20mm$, $H = 150\mu m$, $w_m = 50\mu m$, filled markers are flow networks subject to the stability constraint, open markers are networks not subject to the stability constraint	71
3.12	Optimal β , and γ as a function of m for $H = 150\mu m$, $R = 20mm$ not subject to the stability constraint	72
4.1	Illustration of a fractal-like branching channel tree	77
4.2	Representative control volumes for the energy balance	79
4.3	Example of temperature distribution obtained from the 3-D CFD model	84
4.4	Nusselt number as a function of x^* for copper and stainless steel interface materials determined from 3-D CFD model, circles (\circ) are data, the dashed line (- -) is the least squares fit	87
4.5	Comparison of analytical and 3-D CFD model of fin temperature distributions through both copper and stainless steel	88
4.6	Nusselt number as a function of x^* for copper and stainless steel interface materials determined from 3-D CFD model, adjusted for fin efficiency, circles (\circ) are data, the dashed lines (- -) is the least squares fit	90
4.7	Nusselt number correlations and present study from Eqn. 4.18 with $C_1 = 0.50$ and $C_2 = 0.56$ as a function of x^*	91
4.8	ε vs. NTU for $C^* = 0.019$ and $\dot{m}_c = 10 - 1000$ g/min obtained from the 1-D model, solid line (-) corresponds to co-flow, dashed line (- -) corresponds to counter flow	93

LIST OF FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
4.9 Bulk fluid and interface temperature profiles as a function of axial distance and grid size obtained from the 1-D model, the three sets of datas correspond to the hot bulk fluid temperature, interface temperature and cold bulk fluid temperature	95
4.10 Illustration of the testing apparatus used to experimentally test the fractal-like branching channel heat exchanger, geometry is outlined in Table 4.1	96
4.11 Experimental versus 1-D Numerical Model ΔT_{lm} for $\dot{m}_c = 60$ g/min, $\dot{m}_h = 200, 300$ and 400 g/min and three cold side inlet temperatures	97
4.12 Heat transfer vs. log-mean temperature difference for $C^* = 0.019$, and $\dot{m}_c = 10 - 130$ g/min obtained from the 1-D model, solid line (—) corresponds to co-flow, dashed line (- -) corresponds to counter flow	99
4.13 Performance parameter ϵ_p , versus mass flow rates of the hot and cold side of the heat exchanger obtained from the 1-D model for $C^* = 0.17$, and $\dot{m}_c = 10 - 130$ g/min obtained from the 1-D model, solid line (—) corresponds to co-flow, dashed line (- -) corresponds to counter flow	101
4.14 Benefit to cost ratio, ϵ_p versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ C$, $R = 20mm$ and $w_m = 50\mu m$	105
4.15 Heat exchanger hot side mass flow rate, \dot{m}_h versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ C$, $R = 20mm$ and $w_m = 50\mu m$	106
4.16 Heat exchanger UA versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ C$, $R = 20mm$ and $w_m = 50\mu m$	107
4.17 Overall heat transfer coefficient per unit area (U) versus axial distance of the straight channel geometry for $C^* = 0.17$, $T_{h,in} = 150^\circ C$ and $T_{c,in} = 25^\circ C$	109

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
4.18	Heat exchanger effectiveness, ε versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ C$, $R = 20mm$ and $w_m = 50\mu m$	110
4.19	Heat exchanger effectiveness, ε and benefit to cost ratio, ϵ_p for both the fractal and straight geometries versus terminal channel width, w_m for a $\Delta T_c = 20^\circ C$, and $R = 20mm$ in the counter-flow configuration	111
4.20	Change in ϵ_p and \dot{m}_h of Fractal 2 in co-flow as a function of R for $\Delta T_c = 20^\circ C$ and $w_m = 50\mu m$	114
4.21	Change in ϵ_p and \dot{m}_h of Fractal 2 in counter-flow as a function of R for $\Delta T_c = 20^\circ C$ and $w_m = 50\mu m$	115
4.22	Heat exchanger effectiveness, ε versus cold side mass flow rate, \dot{m}_c for Fractal 2 in the co-flow configuration with $w_m = 50\mu m$ and $\Delta T_c = 20^\circ C$	117
4.23	Heat exchanger effectiveness, ε versus cold side mass flow rate, \dot{m}_c for Fractal 2 in the counter-flow configuration with $w_m = 50\mu m$ and $\Delta T_c = 20^\circ C$	118

LIST OF TABLES

<u>Table</u>		<u>Page</u>
2.1	Comparison of optimized flow power results from the direct numeric search (DS), genetic algorithm (GA), and gradient-based steepest descent (SD) methods with purely geometric-based (GB) results ($R = 20 \text{ mm}$ and $T_{w,max} = 70^\circ C$)	18
2.2	Comparison of optimized flow power network traits from the direct numeric search (DS), genetic algorithm (GA), and gradient-based steepest descent (SD) methods with purely geometric-based (GB) results ($R = 20 \text{ mm}$ and $T_{w,max} = 70^\circ C$)	19
3.1	Range of optimization variables	64
4.1	Geometric parameters of the three heat exchanger geometries with $H = 250 \text{ }\mu m$, $t = 717 \text{ }\mu m$ and $R = 20 \text{ mm}$	98
4.2	Parameters of the cost benefit analysis of Section 4.3.4	102

LIST OF APPENDIX FIGURES

<u>Figure</u>	<u>Page</u>
B.1 Benefit to cost ratio of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	183
B.2 Benefit to cost ratio of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	184
B.3 Benefit to cost ratio of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	185
B.4 Benefit to cost ratio of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	186
B.5 Benefit to cost ratio of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	187
B.6 Benefit to cost ratio of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	188
B.7 Benefit to cost ratio of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	189
B.8 Benefit to cost ratio of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	190
B.9 Benefit to cost ratio of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	191
B.10 Benefit to cost ratio of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	192
B.11 Benefit to cost ratio of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	193
B.12 Benefit to cost ratio of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	194
B.13 Benefit to cost ratio of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	195
B.14 Benefit to cost ratio of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	196

LIST OF APPENDIX FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
B.15 Benefit to cost ratio of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	197
B.16 Benefit to cost ratio of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	198
B.17 Heat exchanger effectiveness of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	199
B.18 Heat exchanger effectiveness of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	200
B.19 Heat exchanger effectiveness of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	201
B.20 Heat exchanger effectiveness of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration	202
B.21 Heat exchanger effectiveness of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	203
B.22 Heat exchanger effectiveness of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	204
B.23 Heat exchanger effectiveness of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	205
B.24 Heat exchanger effectiveness of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration	206
B.25 Heat exchanger effectiveness of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	207
B.26 Heat exchanger effectiveness of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	208
B.27 Heat exchanger effectiveness of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	209
B.28 Heat exchanger effectiveness of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration	210

LIST OF APPENDIX FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
B.29 Heat exchanger effectiveness of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	211
B.30 Heat exchanger effectiveness of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	212
B.31 Heat exchanger effectiveness of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	213
B.32 Heat exchanger effectiveness of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration	214

LIST OF APPENDIX TABLES

<u>Table</u>	<u>Page</u>
B.1 Geometric parameters of the three heat exchanger geometries with $T_{c,in} = 20^{\circ}C$, $T_{h,in} = 150^{\circ}C$, $H = 250 \mu m$, $t = 717 \mu m$ and $R =$ $20 mm$	182

Nomenclature

Roman

a	Inlet plenum diameter constraint	
A	Cross sectional channel area	(m^2)
b	Terminal channel spacing constraint	
C	Heat capacity rate $\dot{m}C_p$	(W/K)
C^*	Heat capacity rate ratio $\frac{C_{min}}{C_{max}}$	
$C_{f,app}$	Apparent friction factor	(μm)
C_{LM}	Phase interaction parameter	
C_p	Specific heat	$(J/kg-K)$
d	Interior channel overlap constraint	
D_h	Channel hydraulic diameter	(m)
G	Mass flux	$(kg/m^2 s)$
Gz	Graetz Number	
H	Channel height	(m)
h	Heat transfer coefficient	(W/m^2-K)
h_{fg}	Latent heat	(J/kg)
k	Branch level index	
$k_{s,f}$	Thermal conductivity of the solid (s) or fluid (f)	$(W/m-K)$
L	Channel length	(m)
m	Total number of branch levels	
\dot{m}	Mass flow rate	(kg/s)

NTU	Number of transfer units $\frac{UA}{C_{min}}$	(μm)
Nu	Nusselt number	
n_0	Number of inlet plenum channels	
P	Pressure	$(Pascal)$
q	Heat transfer	$(Watts)$
q''	Disk heat flux	(W/m^2)
Q	Volumetric flow rate	(mL/min)
R	Disk radius	(m)
Re	Reynolds number	
T	Temperature	$(^{\circ}C)$
t	Interface material thickness	(μm)
\Re	Flow resistance	(kg/m^4s)
UA	Overall heat transfer coefficient	$(Watts/K)$
u_m	Average axial fluid velocity	(m/s)
w	Channel width	(m)
x	Streamwise distance	(m)
X	Lockhart-Martinelli Parameter	
z	Vapor quality	

Greek

α	Void fraction
β	Width ratio
γ	Length ratio

δ	Liquid film thickness	(<i>m</i>)
Δx	Discretization distance	(<i>m</i>)
ΔP	Pressure drop across the device	(<i>Pascals</i>)
ε	Heat exchanger effectiveness	
ϵ_p	Benefit to cost ratio	
η	Fin efficiency	
θ	Temperature difference $T_w - T_b$	(K)
λ_{lo}	Single-phase friction factor	
ρ	Density	(<i>kg/m³</i>)
$\sum R$	Thermal resistance	(<i>K/W</i>)
ϕ^2	Two-phase frictional pressure drop multiplier	

Subscripts

c	Cold side
CHF	Critical heat flux
e	exit
h	Hot side
i	Discretization index
in	Inlet plenum
k	Branch level index
l	Liquid phase
m	Terminal branch level
max	Maximum

min	Minimum
s	Solid interface
v	Vapor phase

Dedication

To my wife, thanks for being there.

On the Optimization of Performance in Fractal-like Branching
Microchannel Heat Transfer Devices

Chapter 1 – Introduction

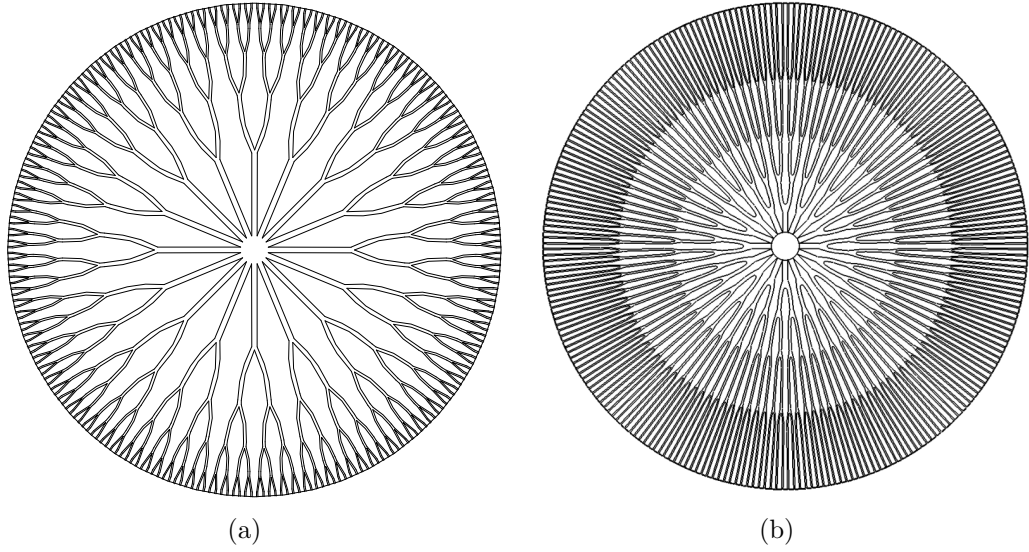


Figure 1.1: Representative flow networks with 16 branches emanating from the inlet plenum at the center of the disk, $m = 4$ branching levels and (a) decreasing channel length, and (b) increasing channel length

Since the groundbreaking findings of Tuckerman and Pease [1] on the advantages and drawbacks of microscale flow, countless studies have sought to further increase the heat transfer capabilities, and mitigate pressure drop and flow instabilities associated with single-phase and flow boiling at the microscale. The specific topic of the present study is on the modeling and optimization of fractal-like branching microchannels. Representative fractal-like flow networks are shown in

Fig. 1.1, illustrating the difference between fractal-like designs with (a), decreasing channel lengths and (b), increasing channel lengths. One-dimensional models of the the temperature and pressure distributions through the fractal-like microchannel heat transfer devices form objective functions to be minimized through a gradient-based optimization algorithm. Constraints on the geometry and flow are employed to mimic real-world operating conditions. Three separate studies are presented in this dissertation: (i) The Optimization of Single-Phase Fractal-like Branching Channel Heat Sinks, (ii) The Optimization of Two-Phase Fractal-like Branching Channel Heat Sinks and (iii) Modeling s Single-Phase Fractal-like Branching Microchannel Heat Exchanger. While all three of these manuscripts studies a different type of fluidic system, what ties all three together is the end goal of finding an optimum flow network subject to real-world physical constraints.

The uniqueness of the three manuscripts stems from the analysis of the fractal-like branching micro-channel flow networks. These fractal-like flow networks mimic designs found in nature, i.e. mammal and plant vascular networks. The flow networks have constant width and length ratios, i.e., each the ratio of the subsequent downstream channel length and width to the upstream length and width are fixed, respectively. Moreover, following each bifurcation (split of the channel into two downstream channels) the thermal and hydrodynamic boundary layers were found to re-develop [2, 3]. The re-development of the boundary layers following each bifurcation results in an enhancement of the heat transfer coefficient, but also an increase in the wall shear stress. Thus, the purpose of this dissertation is to find an optimal balance between benefit (heat transfer) and cost (flow power) by varying

the geometry and flow conditions in fractal-like microscale heat transfer devices.

Chapter 2 – Optimization of Single-Phase Fractal-like Branching Channel Heat Sinks

2.1 Introduction

Disk-shaped heat sinks with branching microchannels were first proposed by Pence [4] as a means to reduce both the pressure drop and the maximum streamwise wall temperature difference observed in parallel microchannel heat sinks. A fractal-like branching flow network, inspired by nature, was studied. Representative fractal-like flow networks in a disk-shaped heat sink configuration are shown in Fig. 1.1. Flow enters the network from the inlet plenum located at the center of the disk and exits at the periphery. Each channel emanating from the inlet plenum bifurcates into two narrower channels, each of which in turn bifurcates. This repetitive pattern is considered fractal-like because the ratio of the channel widths and channel lengths between the consecutive branch levels are fixed. The objective of the present study is to develop an optimization algorithm to identify geometric characteristics, subject to operating and fabrication constraints, of a fractal-like flow network that achieves the minimum single-phase flow power for a desired heat removal.

The gradient-based optimization algorithm used in the present study employs the one-dimensional model developed by Pence [5] for predicting pressure and wall

temperature distributions in fractal-like branching channel networks. The model is based on developing laminar flow and heat transfer under a constant wall heat flux condition. Hydrodynamic [6] and thermal [7] boundary layers were assumed to redevelop following each bifurcation. Alharbi, et al. [2,3] validated the model using three-dimensional computational fluid dynamic and thermal analyses, respectively. Using the one-dimensional model of Pence [5], Pence and Enfield [8] investigated the influence of several geometric parameters on the pressure drop and maximum wall temperature. It was determined that increasing the number of branch levels resulted in reduced flow resistance, but also resulted in an increase in the maximum predicted wall temperature.

Using constructal theory, Bejan [9] designed optimal flow networks that minimized global flow resistance between a single point and a volume. Bejan and Errera [10] extended the analysis to simultaneously minimize flow and thermal resistance while optimizing the cooling of a volume experiencing uniform heat generation. Using the disk shaped concept proposed by Pence [4], Lorente et al. [11] employed constructal theory to minimize cooling path lengths from the center of a planar disk to a series of equally spaced points on the disk circumference. It was concluded that minimizing channel lengths yielded flow resistances very similar to those based on a minimization of flow resistance, which were presented in a parallel study by Wechsato et al. [12]. In the same disk configuration, Wechsato et al. [13] later minimized both the flow and thermal resistances. For a fixed flow rate the thermal resistance was minimized with non-branching channels, whereas flow resistance was minimized with increased numbers of branch levels. In these

optimization studies of constructal networks [9–13], the flow was assumed laminar with fully developed flow, variable length and width scale ratios were considered, and a fixed coolant volume was imposed. Gonzales et al. [14], also assuming fully developed flow, optimized the heat transfer and fluid resistances by varying the number of downstream branches coupled to an upstream branch.

In the present optimization study of fractal-like flow networks, more restrictive fabrication constraints than have been previously considered are taken into account. In addition to a maximum wall temperature constraint, the present study also includes a restriction on the inlet plenum diameter, which is highly influenced by the number of channels connected to the inlet plenum. Several constraints imposed by the fabrication process, such as the minimum channel width and minimum spacing between channels to allow sufficient bonding area, are also imposed. Optimization is performed for a fixed physical space, in this case the disk diameter, as opposed to a fixed fluid volume. The objective of the optimization is to find the flow network that minimizes flow power or pressure drop while providing a required cooling load and adhering to a maximum wall temperature constraint. Results of the gradient-based optimization are validated with a direct search over the entire range of variables and are compared with results from a genetic algorithm.

In the direct numeric search, each parameter is varied at discrete intervals over a specified range. The flow power for each parametric permutation is computed using the one-dimensional model. Finally a search of the entire set of results is conducted to determine the geometry that minimizes flow power, the process being very time intensive.

The gradient-based steepest descent search starts at a single point within the parameter space, evaluates the gradient, and moves in incremental steps in the direction of the minimum [15]. The steepest descent approach has two distinct advantages over the direct search. First, there is a reduction in the number of executions of the one-dimensional model to determine the optimized flow network. The second advantage is continuous parameters with finer resolution than can be achieved using the direct search and its discrete intervals. A major drawback of the steepest descent approach is its inability to distinguish between a local and global minimum. However, repeating the search for a range of randomly selected starting points within the parameter space can mitigate this drawback.

Inspired by the role genetics plays in the evolutionary process [16], the genetic algorithm starts with a randomly selected population of potential solutions within the parameter space. After assessing the quality of these potential solutions, new solution sets are created in which the desired traits from the best possible solutions of the original set are combined and in some cases, improved. The process is repeated until variations in parameters are insignificant. An advantage of the genetic algorithm is that it is not hindered by the presence of local minima in its search for a global minimum. The drawback of this method is that the solution to which it converges can depend upon the size of the initial population.

2.2 Geometric parameters and constraints

For applications conducive to disk-shaped heat sinks, specified parameters include the disk radius, R , and applied heat flux, q'' . For many applications, including electronic cooling applications, the heat sink has a maximum junction temperature limit. Using a resistance analogy, the junction temperature limit can be related to the maximum wall temperature, $T_{w,max}$, of the flow network. The wall temperature can be reduced by lowering the inlet fluid temperature, T_{in} , and/or by increasing the inlet volumetric flow rate, \dot{V}_{tot} . The inlet temperature of the working fluid, which is water for the present study, is fixed at 20°C. Increasing the flow rate is done at the expense of increased pressure drop, hence, increased flow power. Therefore, the lowest flow rate that just meets a specified maximum wall temperature limit is selected.

The available fabrication technique imposes additional constraints, generally on the maximum depth of the microchannels, H , the narrowest channel width, $w_{m,min}$, and the minimum inter-channel and terminal channel spacing, $w_k d$ and bw_m , respectively, as noted in Fig. 2.1. In general, holding all other geometric and flow parameters constant, an increase in channel depth increases the channel wall area, which for a fixed heat flux applied to a heat sink of highly conductive material results in a decrease in maximum wall temperature. An increase in channel depth also increases the cross-sectional area of the channels, which for a fixed flow rate results in a decrease in pressure drop and flow power. Thus the channel depth is set at the maximum allowed by the fabrication technique, which for the present study

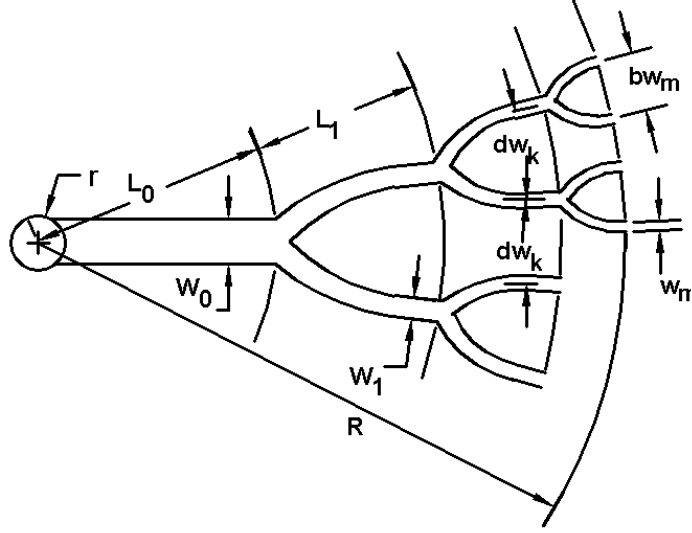


Figure 2.1: Schematic representation of branch level channel lengths and channel widths, and constraints on channel spacing

is $150 \mu m$. The smallest channel width that can be fabricated is assumed to be $50 \mu m$. The center-to-center spacing of the terminal channels around the periphery of the disk, as well as between channels in other levels of the flow network, must be sufficiently large to provide a surface to which the layers forming the heat sink can be bonded together. Channel spacing limitations are discussed in more detail later in terms of constraints on the geometry of the flow network.

Free to vary in the optimization are the width ratio, β , and length ratio, γ , between consecutive branch levels, which are defined by

$$\beta = \frac{w_{k+1}}{w_k} \quad (2.1)$$

$$\gamma = \frac{L_{k+1}}{L_k} \quad (2.2)$$

and where w_k and L_k represent the width and length, respectively, of a channel within branch level k (see Fig. 2.1). Note that k varies from 0 through m , with m denoting the total number of branch levels. The 0^{th} level branches, denoted by n_0 , are those that emanate from the inlet plenum.

Ranges of β , γ , w_m , m , and n_0 are specified for each disk of radius, R . In addition, four fabrication constraints on the network geometry are imposed in the optimization algorithm. The first is that the inlet plenum radius is restricted to less than or equal to 10 percent of the disk radius. The parameter a represents the ratio of inlet plenum radius to disk radius. The inlet plenum radius is determined from its circumference, which is equal to the number of 0^{th} level channels, n_0 , times their width, w_0 . The value of w_0 can be assessed from

$$w_0 = \frac{w_m}{\beta^m} \quad (2.3)$$

yielding the following first geometric constraint, which constrains the inlet plenum diameter

$$\frac{n_0 w_m}{\beta^m} \leq 2\pi R a \quad (2.4)$$

The second and third geometric constraints are both applied to the terminal channel geometry, i.e. there must be sufficient material between channels to support bonding of a top disk to finalize the flow network. A dimensionless parameter, b , is defined as the center-to-center spacing between the terminal (m level) channels

divided by the width of the terminal channels, w_m . This parameter is allowed to vary between a minimum value, b_{min} , and a maximum value, b_{max} . A similar spacing parameter, d , is defined for the levels $k = 0$ through $m-1$, and for the present analysis is set equal to b_{min} .

To ensure that the terminal channels are adequately spaced, the circumference of the disk divided by b_{max} and by b_{min} must bracket the sum of the widths of the terminal channels as in

$$\frac{2\pi R}{b_{max}} < 2^m n_0 w_m < \frac{2\pi R}{b_{min}} \quad (2.5)$$

Eq. (5) imposes two separate constraints on the terminal channel spacing at the periphery of the disk. For the remaining branch levels, i.e. $k = 0$ through $m-1$, a minimum center-to-center spacing of $w_k d$ is ensured between channels of width w_k , if the following inequality

$$\left(\frac{2\pi R \beta^m}{n_0 w_m} - 1 \right) \frac{\sum_{i=0}^k \gamma^i}{\sum_{i=0}^m \gamma^i} > d (2\beta)^{k+1} - 1 \quad (2.6)$$

is valid. Eq. (6) represents the fourth geometric constraint, a constraint on the internal channel spacing, which is imposed at each level k . Although ranges of m and n_0 are specified, the upper limit on m is constrained by

$$2^m < \frac{2\pi R}{b_{min} n_0, \min w_m, \min} \quad (2.7)$$

as determined from Eq. 2.5.

To summarize, the following needs to be specified by the flow network designer:

(I) single values for geometry variables a , b_{min} , b_{max} , d , H , and R , (II) ranges for geometry values m , n_0 , β , γ , and w_m , and (III) single values for operating variables q'' and $T_{w,max}$. If solving using the direction numeric search, increments also need to be specified for β , γ , and w_m . Within the parametric space defined by ranges of β , γ , and w_m there may be numerous possible flow configurations. Of these, many will not adhere to the geometric constraints imposed by Eqs. (4-6) and are considered invalid geometries. Figure 2.2 shows a representative range of valid length and width ratios for a fixed R , w_m and n_0 . Contours in Fig. 2.2 indicate flow power determined by a direct search [8]. Although the branching level, m , is also a flow configuration variable, its value is fixed in Fig. 2.2 to simplify visualization of the valid space.

2.3 Optimization Techniques

The primary method of optimization for the present study is the gradient-based steepest descent (SD) search, which is validated using the direct numeric search (DS) and compared with results from the genetic algorithm (GA).

2.3.1 Gradient-Based Steepest Descent Search

Although a relatively straight forward optimization approach, discrete, integer values of m impose challenges to the steepest decent search. If an optimization is performed for each m value, however, this problem is alleviated. As noted in Enfield

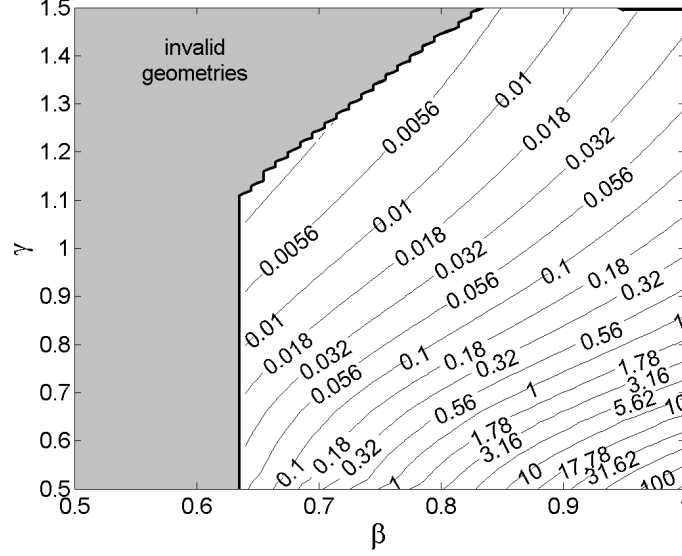


Figure 2.2: Representative division between valid and invalid geometries for $R = 20\text{mm}$, $w_m = 85\mu\text{m}$, $n_0 = 10$ and $m = 6$. Contours indicate flow power determined from the direct search conducted by Enfield et al. [8]

et al. [8], the discrete nature of n_0 creates steps, or local minima, in the geometric constraint boundaries that render complicated search paths along the boundaries that are difficult to calculate. Fortunately, n_0 is also represented by an integer value. Therefore, to simplify the steepest descent search, the search is conducted in a region defined by a single value of m and a range of integer values of n_0 . The steepest descent approach proceeds as follows. A starting point that meets all constraints is chosen arbitrarily for a given m and n_0 . Using predefined increments of β , γ , and w_m , six points $(\pm\Delta\beta, \pm\Delta\gamma, \pm\Delta w_m)$ surrounding the arbitrarily selected point are identified and the pressure drop or flow power is assessed. Using these seven points, the local gradient of the function to be minimized is numerically

determined at the arbitrarily chosen point. Incremental steps are taken in the negative gradient direction until the local value of the function in question, for example the flow power, stops decreasing. At this new point, the gradient is again determined, and the value of the function at the current point is compared to those at neighboring points. If the current point is not a minimum, incremental steps are taken in the new direction, i.e. in the negative gradient direction. The process is repeated until a minimum, defined by the gradient tending toward zero, is achieved. For a given value of m , the optimum flow network, defined by the lowest pressure drop or flow power, is identified for each n_0 . The minima for all n_0 values are then compared to find the optimum for a fixed value of m . By comparing the minimum values of pressure drop or flow power for each valid m , the final optimized flow geometry is determined.

2.3.2 Direct Numerical Search

In a direct numerical search, all possible parametric permutations of the flow geometry are considered for a given disk radius. Using Eqs. (2.4-2.6), flow geometries violating any of the constraints are removed from further consideration. The one-dimensional model by Pence [5] is run for all valid flow geometries to determine the flow rate that just yields the acceptable maximum wall temperature for a specified heat flux. This requires an iterative process. Because the model also provides the total pressure drop, the minimum flow power among all valid flow geometries can be determined. The flow network configuration yielding the minimum flow power

is specified as optimal.

2.3.3 Genetic Algorithm

For each m and n_0 combination, the genetic algorithm searches for the optimum combination of β , γ , and w_m that yields the lowest flow power that just meets the $T_{w,max}$ constraint. An initial population is created from 50 random combinations of traits β , γ , and w_m that satisfy the geometric constraints. Each of these possible solution combinations, referred to as individuals, is evaluated for its fitness. The fitness function is the one-dimensional model by Pence [5], where the fitness value is the inverse of the pumping power. The individuals are then ranked from the highest fitness value to the lowest. The two individuals with the highest fitness values, termed elite individuals, are passed unaltered to the next generation population set. The remaining individuals in the top 40% fitness rankings are selected for crossover. During crossover, individuals are randomly paired to create offspring by combining traits from each individual. The traits of the remaining individuals in the population are mutated to ensure that all constraints are satisfied. The mutants, offspring, and elite individuals form a new population. The process is repeated with each new population until variations in the magnitude of the fitness values throughout the population is less than the precision of the computer (2×10^{-16}).

2.4 Results

Gradient-based steepest descent optimization results are presented for two separate searches, one in which pressure drop is minimized and the other in which flow is power minimized. Comparison of this optimization method with direct numeric and genetic algorithm searches are limited to the flow power minimization criterion. Flow geometries for six different heat fluxes [10, 30, 100, 200, 500, 1000 W/cm^2], five disk radii [10, 20, 30, 40, 50 mm] and six maximum wall temperatures [40 50, 60, 70, 80, 90 $^{\circ}C$] were considered. The range considered for w_m was 50 to 150 μm , with 1 μm increments employed for the direct search. The ranges for β and γ were constrained between 0.5 and 1.5 and between 0.5 and 1.0, respectively. Increments of 0.01 were used for the direct search. Values for m and n_0 were allowed to vary from 2 to 9 and from 3 to 100, respectively. The remaining geometric parameters were held fixed and assigned values $H = 150 \mu m$, $a = 0.1$, $b_{min} = d = 1.5$, and $b_{max} = 2.5$.

2.4.1 Comparison of Gradient-based, Genetic Algorithm and Direct Search Results

Outlined in Table 2.1 are flow powers minimized using the gradient-based steepest descent (SD), the direct numeric search (DS), and genetic algorithm (GA) for $R = 20 \text{ mm}$, $T_{w,max} = 70^{\circ}C$ and three values of heat flux: 10, 500, and 1000 W/cm^2 . Flow powers, \wp , from the steepest descent are within 10% of those from the direct

Table 2.1: Comparison of optimized flow power results from the direct numeric search (DS), genetic algorithm (GA), and gradient-based steepest descent (SD) methods with purely geometric-based (GB) results ($R = 20 \text{ mm}$ and $T_{w,max} = 70^\circ\text{C}$)

$q'' \text{ (W/cm}^2\text{)}$		DS	GA	SD	GB
10	$\wp \text{ (mW)}$	0.19	0.2	0.19	0.20
	$Q \text{ (mL/s)}$	0.21	0.26	0.22	0.183
	$\Delta P \text{ (kPa)}$	0.91	0.77	0.82	1.07
	ϵ_p	5.9E5	5.7E5	6.3E5	5.8E5
500	$\wp \text{ (mW)}$	1336	1661	1219	1304
	$Q \text{ (mL/s)}$	10.6	12.3	10.6	8.99
	$\Delta P \text{ (kPa)}$	126	135	115	145
	ϵ_p	4246	3415	4652	4350
1000	$\wp \text{ (mW)}$	10,980	11,300	10,000	11,300
	$Q \text{ (mL/s)}$	26.3	24.7	26.3	29.3
	$\Delta P \text{ (kPa)}$	418	458	381	385
	ϵ_p	1033	1004	1134	1004

search. Flow powers minimized using the genetic algorithm differ by as much as 25% from direct search results. Also included in Table 2.2 are the total volumetric flow rate, pressure drop and benefit-to-cost ratio associated with these optimal flow powers. Shown in Table 2.2 are the parameters, or traits, of the corresponding optimal flow networks. Included are width ratio, length ratio, terminal channel width, and the numbers of branch levels and 0^{th} order channels. Geometric-based (GB) results are discussed later. Each optimization algorithm considered suffers from some drawback. The direct search suffers from excessive computational requirements and limited resolution. As an example, 525,207 permutations of flow and geometrical conditions were computed in 36 hours to find the minimum flow

Table 2.2: Comparison of optimized flow power network traits from the direct numeric search (DS), genetic algorithm (GA), and gradient-based steepest descent (SD) methods with purely geometric-based (GB) results ($R = 20 \text{ mm}$ and $T_{w,max} = 70^\circ\text{C}$)

$q'' \text{ (W/cm}^2\text{)}$		DS	GA	SD	GB
10	β	0.67	0.65	0.64	0.67
	γ	1.09	0.98	1.06	1.08
	$w_m(\mu m)$	52	53	51	50
	m	6	7	7	7
	n_0	21	11	11	13
500	β	0.76	0.76	0.75	0.67
	γ	1.19	1.06	1.18	1.08
	$w_m(\mu m)$	50	53	50	50
	m	4	4	4	7
	n_0	78	79	79	13
1000	β	0.76	0.84	0.75	0.67
	γ	1.19	1.28	1.18	1.08
	$w_m(\mu m)$	50	51	50	50
	m	4	3	4	7
	n_0	78	144	79	13

power condition using the direct numerical search. The resolution of β and γ were 0.01 and w_m was $1 \mu m$ for this direct search. The steepest descent is fast, but requires runs with various starting points to avoid local minima. For each starting point, and for the same parameter space as the direct search, minimum flow power condition was obtained within 23.4 seconds using the same computational resource. The genetic algorithm is faster than the direct search but considerably slower than the steepest descent. For the same parameter space as the gradient-based and direct numerical searches, a minimum in flow power condition was determined in

2300 seconds. In addition, the genetic algorithm suffered from not always finding a converged solution, a condition that may have resulted from a lack of diversity due to a small initial population. However, increasing the population size results in an increase in the convergence time. In comparing results to the direct search, the steepest descent method, as implemented, yielded only global minima.

The parameters β , γ , and w_m that characterize the optimized geometries appear to be fairly independent of optimization method used (within approximately 15%). Differences between the different methods are mainly influenced by the increment size in the direct search and the initial population size employed in the genetic algorithm.

2.4.2 Gradient-Based Steepest Descent Results

For each discrete value of disk radius, R , maximum wall temperature constraint, $T_{w,max}$, and disk heat flux, q'' , the geometric parameters β , γ , and w_m that yielded the lowest flow power or pressure drop for a given m and n_0 were determined using the steepest descent optimization method.

2.4.2.1 Pressure drop minimization

Minimal values of pressure drop are shown in Fig. 2.3 as a function of the number of branch levels, m , for applied heat fluxes ranging from 10 to 1000 W/cm². Assumed is a disk-shaped heat sink having a radius of 20 mm and constrained by a maximum

wall temperature of 70°C. Circles are used to identify the optimum number of branch levels, m , for each specified heat flux. Observed in Fig. 2.3 is that as the heat flux increases the pressure drop increases and the optimal value of m decreases. For a fixed flow network configuration, as the heat flux increases, the flow rate must also increase so as not to violate the maximum allowable wall temperature constraint. This results in an increase in pressure drop.

Although pressure drop is plotted as a function of m in Fig. 2.3, m is not the only parameter changing. Rather, changes in m are accompanied by changes in n_0 . For example, if the width ratio, length ratio, terminal channel widths and terminal channel spacing are all fixed, a unit decrease in m yields an automatic doubling of n_0 . This assumes the inlet plenum constraint (Eq. 2.4) is not violated. Also changing with m is the convective surface area defined as

$$A_w = n_0 L_0 \left[\sum_{k=0}^m 2^{k+1} \gamma^k \left(H + \frac{w_m}{\beta^{m-k}} \right) \right] \quad (2.8)$$

The surface area is based on an assumption of constant heat flux applied to four walls of the channel. Evident from Eq. 2.7 is that independent increases in n_0 and in m result in increases in convective surface area. The magnitude by which surface area increases depends upon the values of m , n_0 , β and γ . However, for a fixed disk heat flux and flow rate an increase in convective area yields a decrease in wall temperature. Because the optimization process identifies the lowest flow rate that just achieves the maximum wall temperature constraint, the optimal flow rate is also decreased with increases in convective surface area. An increase in m

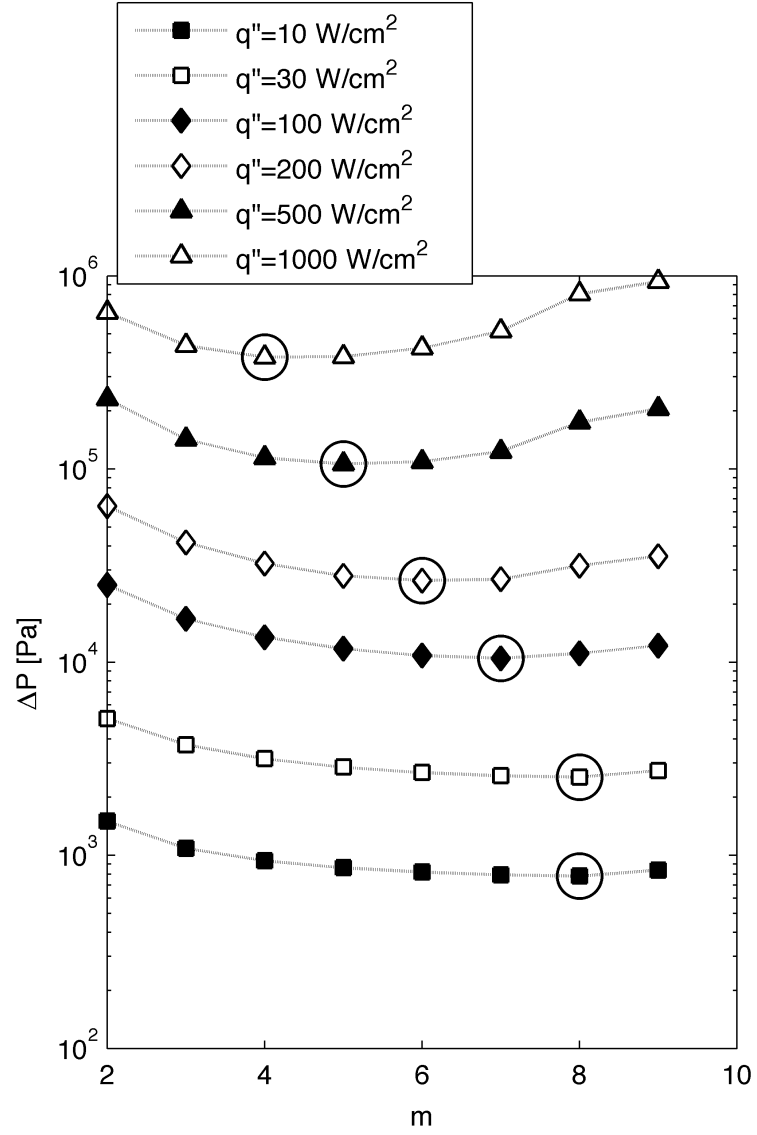


Figure 2.3: Minimized pressure drop versus m as a function of disk heat flux for $R = 20\text{mm}$ and $T_{w,max} = 70^\circ\text{C}$. Large circles denote optimal m values for each heat flux.

while holding all other parameters constant, including n_0 , results in a reduction in pressure drop due to a decrease in flow resistance. This occurs because the channel lengths in each branch level become shorter while the width of the widest channel increases. An increase in n_0 while holding all other parameters constant, including m , results in a decrease in flow rate through each channel. As noted previously, changes in these parameters are not independent of changes in other parameters. However, in general, increases in A_w , m , and n_0 can be thought to result in decreases in flow rate, pressure drop and flow rate, respectively.

The relationship between the convective surface area, A_w , and parameters n_0 , β , γ , and w_m are shown as a function of m in Fig. 2.4. The disk diameter is fixed at 20 mm, subjected to a heat flux of 1000 W/cm^2 and limited to a maximum wall temperature of 70°C . The terminal channel width, w_m , is made non-dimensional using the fixed channel depth, H , and identified with a * superscript. As noted previously, a unit decrease in m tends to yield twice the number of n_0 , which is illustrated in Fig. 2.4a by the exponential decay in n_0 with increases in m . As m is decreased and n_0 increased, there is a noticeable increase in the convective surface area, A_w , in Fig. 2.4f. Also contributing to the increase in convective surface areas are increases in β and γ , as noted in Figs. 2.4b and 2.4c, respectively. Evident from Eq. 2.8 and by comparing Fig. 1.1(a) ($\beta = 0.7$) to Fig. 1.1(b) ($\gamma = 1.4$), holding all other parameters constant, higher values of γ tend to increase the convective area. This trend is observed for m less than 7 in Fig. 2.4c. However, as is noted by the large value of w_m^+ at $m = 9$ in Fig. 2.4d, γ is also influenced by the terminal channel width. This anomalous value of w_m^+ is addressed later.

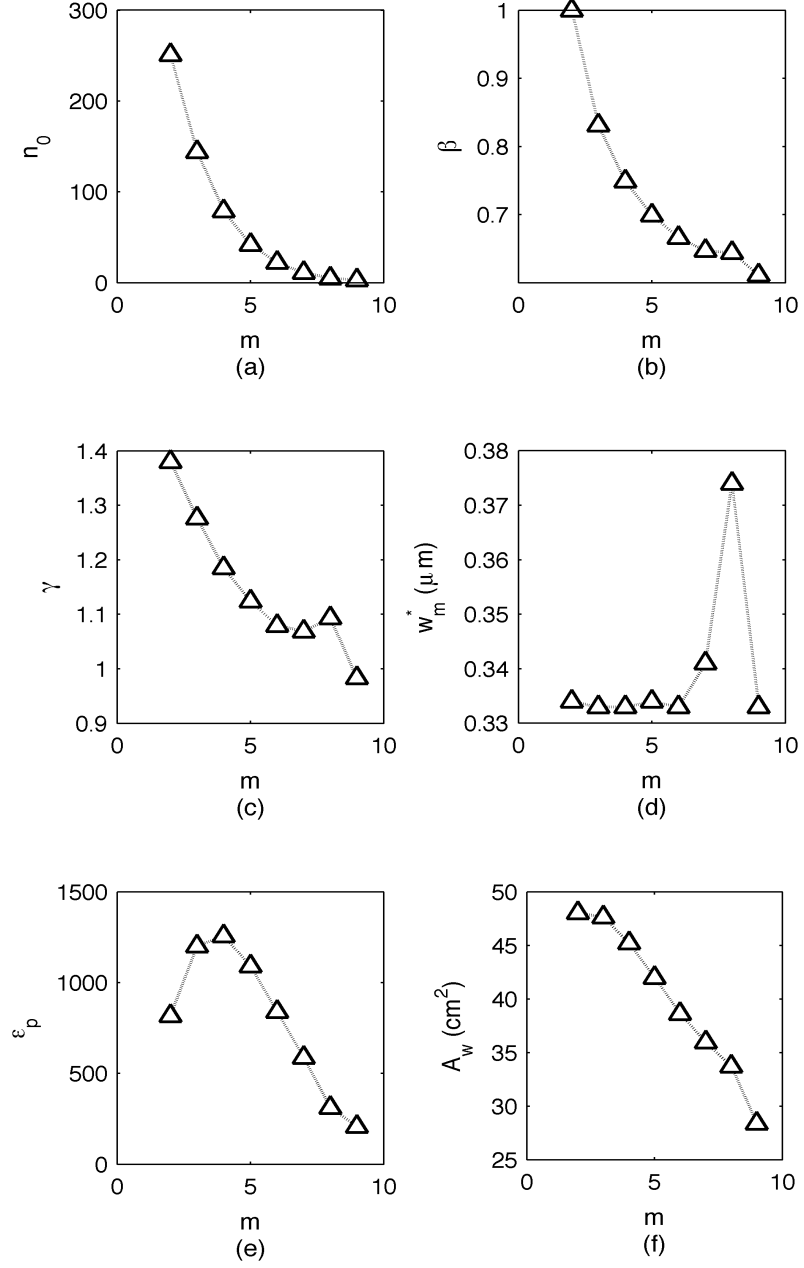


Figure 2.4: Minimized pressure drop flow network parameters, benefit-to-cost ratio and channel area versus m for $q'' = 1000\text{W}/\text{cm}^2$, $R = 20\text{mm}$, and $T_{w,max} = 70^\circ\text{C}$.

Values of β in Fig. 2.4b are influenced by the inlet plenum diameter constraint. For a unit decrease in m , in which case n_0 is doubled, the sum of the widths of the n_0 channels must adhere to the inlet plenum constraint. Two ways in which this may be achieved are by decreasing w_m or increasing β . It is clear from Fig. 2.4d that the minimum channel width constraint of $50 \mu m$, corresponding to a dimensionless width of 0.33, is reached for most values of m . This is an indication that, if permitted by fabrication constraints, smaller terminal channel widths should be considered. For β equal to the upper limit of unity, as is the case for $m = 2$, the channel widths do not increase with decreasing k , but rather stay the same allowing for maximum convective area. As the number of branch levels, m , increase, β must decrease in order to adhere to the inlet plenum and interior channel spacing constraints.

Shown in Fig. 2.4f for this high heat flux condition of 1000 W/cm^2 is that the convective surface area is highest for low values of m and high values of n_0 , with a peak value at $m = 2$. The benefit-to-cost ratio is defined as the rate at which thermal energy is extracted divided by the flow power,

$$\epsilon_p = \frac{q'' \pi R^2}{\dot{V} \Delta P} \quad (2.9)$$

and shown in Fig. 2.4e on semi-log axes. A peak in ϵ_p occurs at $m = 4$. Both the convective area peak and the peak value of ϵ_p are close to $m = 3$, which corresponds to the minimum pressure drop for the 1000 W/cm^2 heat flux in Fig. 2.3. Because the pressure drop, not the benefit-to-cost ratio nor convective area, was optimized,

the peak values do not correspond identically.

Referring back to Fig. 2.3, wherein the cause for increases in pressure drop with heat flux was clear, the cause for decreases in the optimal value of m with increasing heat flux is less clear. Recall that each different symbol in Fig. 2.3 corresponds to the optimal flow network for a given value of m and disk heat flux. The optimal m value for a given heat flux, i.e. the one that yields the smallest pressure drop, is identified in Fig. 2.4 with a circle. In Fig. 2.5 are the geometric parameters and ϵ_p corresponding to the optimal values of pressure drop for each disk heat flux noted in Fig. 2.3. These parameters are plotted as a function of disk heat flux. It is evident from Figs. 2.5a and 2.5b that as the disk heat flux is increased, the decreases in the optimal number of m observed in Fig. 2.3 are accompanied by increases in n_0 . In agreement with observations made in Fig. 2.4, higher values of n_0 are generally accompanied by higher values of β and γ . However, the influence on convective surface area resulting from simultaneous decreases in m and increases in n_0 , β and γ is not exactly clear.

For the minimal pressure drop cases (i.e., circles) in Fig. 2.5a and 2.5b, changes in the disk heat flux from 200 to 500 W/cm² result in a decrease in m from 6 to 5, whereas n_0 approximately doubles from 22 to 42. The values of β and γ at their respective heat fluxes are available from Figs. 2.5c and 2.5d, respectively. Using these numbers in Eq. 2.8 and accounting for changes in L0, there is a 30% increase in surface area for the optimized configuration at 500 W/cm² compared with that at 200 W/cm². The competing effects of decreased flow rates accompanying increases in A_w and n_0 compared with the increased pressure drop accompanying

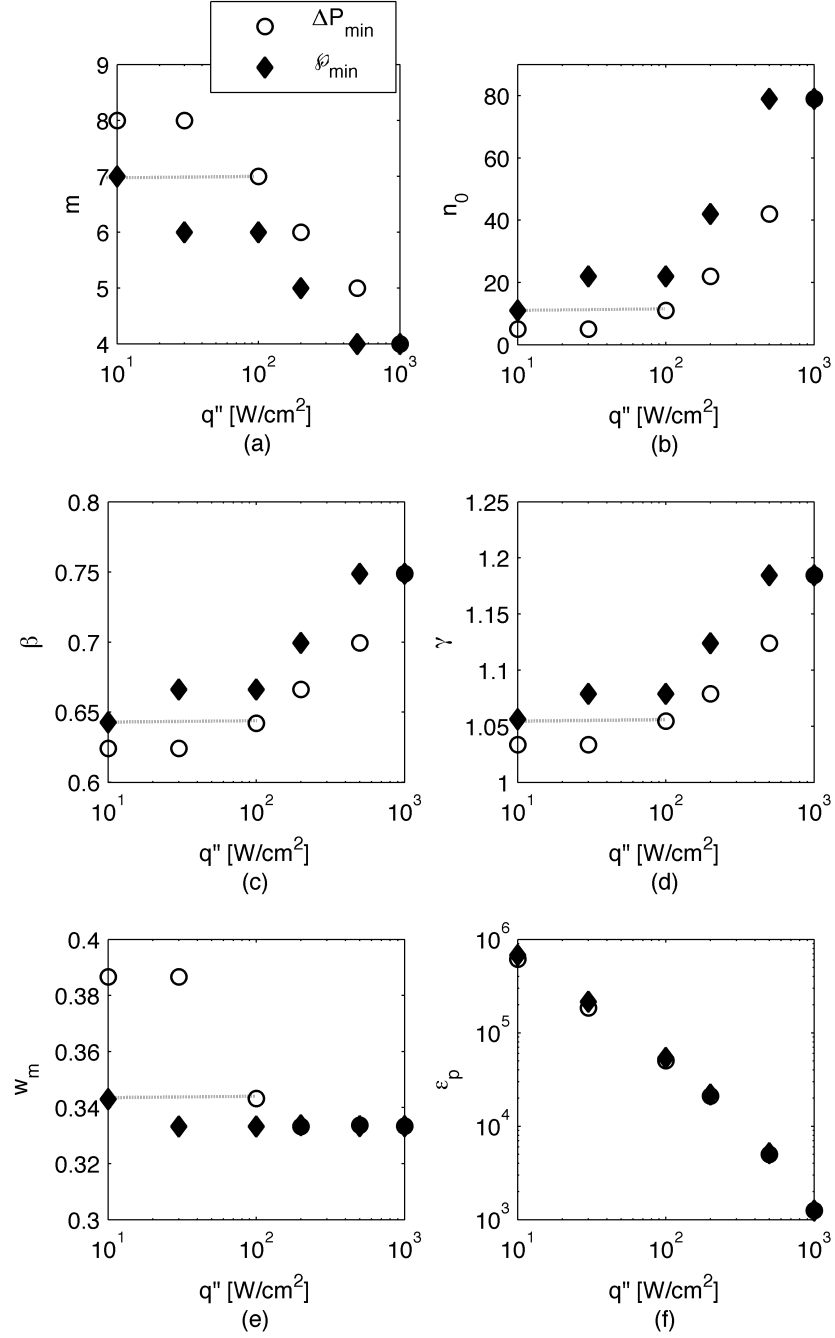


Figure 2.5: Flow network parameters and benefit-to-cost ratio versus applied heat flux for $R = 20mm$, and $T_{w,max} = 70^\circ C$. Dashed lines illustrate similarities between pressure drop and flow power optimization results.

a decrease in m contribute to the observed minimum in Fig. 2.3 for a given heat flux.

In Fig. 2.5, the optimal flow network for the highest heat flux of 1000 W/cm^2 has the highest n_0 and lowest m . To achieve cooling within the desired maximum wall temperature constraint, significantly high mass flow rates are required as are high convective surface areas, the latter of which are maximized through use of very narrow channels. Because pressure drop is inversely proportional to the hydraulic diameter to the fourth power for a given laminar flow rate, it is desired to reduce the flow rate as a means of minimizing pressure drop. Increases in m result in scaled decreases in channel length at each level, thereby providing a scaled reduction in frictional pressure drop. On the other hand, increases in n_0 yield a proportional decrease in flow rate, which in turn yields a proportional decrease in frictional pressure drop. As noted in Fig. 2.5e, the lower limit of $50 \text{ } \mu\text{m}$ ($w_m^+ = 0.33$) is again regularly reached. Competing effects resulting from a desire to maximize thermal transport while minimizing flow resistance are responsible for the trends in the optimal m values observed in Fig. 2.3 with increasing disk heat flux. Shown in Fig. 2.5f on log-log axes is the benefit-to-cost ratio as a function of heat flux. The overall performance of the heat sink, assessed using ϵ_p , decreases as heat flux is increased. In order to understand the trend in ϵ_p , recall that for constant thermophysical properties

$$q''\pi R^2 = \dot{V}\rho C_p (T_{ex} - T_{in}) \quad (2.10)$$

from which it is evident that

$$\dot{V} \propto q'' \quad (2.11)$$

Recall also that the pressure drop across a channel for laminar flow is related to flow rate as follows

$$\Delta P \propto \dot{V} \quad (2.12)$$

Substituting Eqs. 2.10-2.12 into Eq. 2.9 yields $\epsilon_p \propto 1/q''$. On log-log axes, this relationship should be linear. Fig. 2.5f shows a nearly linear relation, with the slight departure a consequence of employing temperature dependent thermo-physical properties in the one-dimensional model.

2.4.2.2 Flow power minimization

Having studied the flow network optimization based on minimizing pressure drop, the minimal flow power assessed for each discrete value of m is plotted in Fig. 2.6 for disk heat fluxes ranging from 10 to 1000 W/cm². The trends are very similar to those exhibited in Fig. 2.3. Again, the number of branch levels characterizing the optimal flow network for each disk heat flux is identified with circles. However, with the exception of optimal geometries for $q'' = 1000\text{W}/\text{cm}^2$, the optimal flow geometries for minimized flow power have fewer branch levels than in those optimized for minimal pumping power for the same heat flux. This is also evident in Fig. 2.5a.

The geometric parameters corresponding to the optimal values of flow power

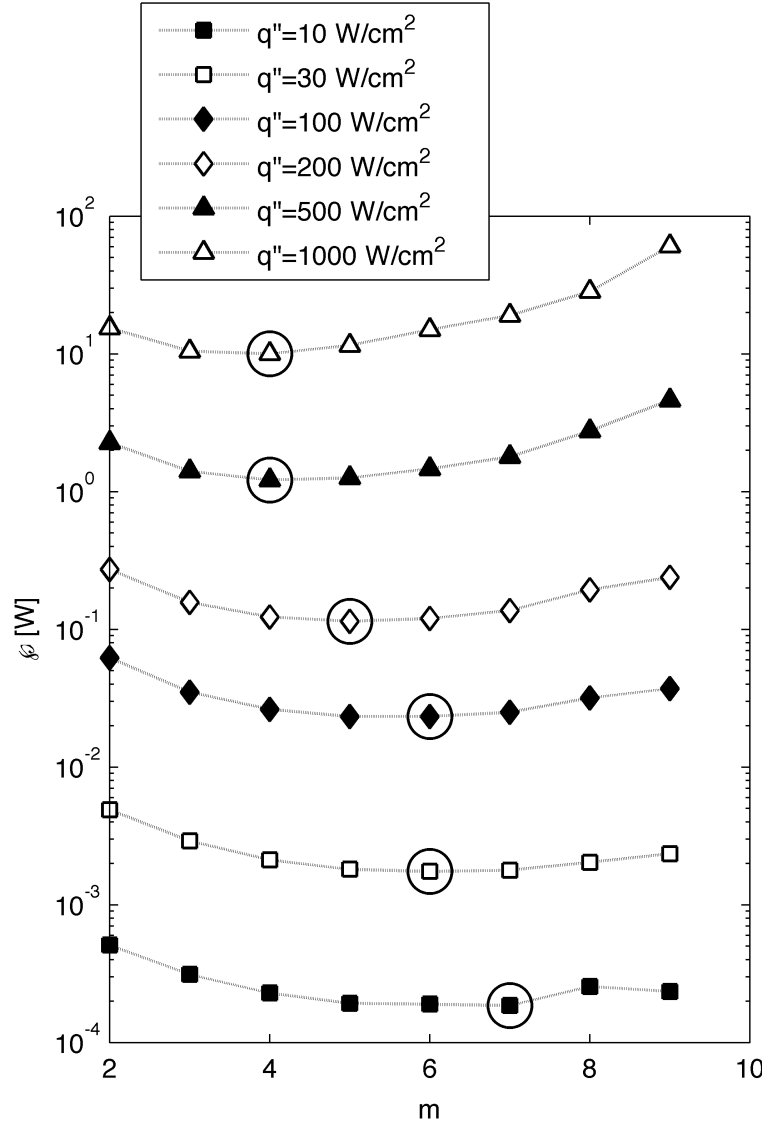


Figure 2.6: Minimized flow power versus m as a function of disk heat flux for $R = 20\text{mm}$, and $T_{w,max} = 70^\circ\text{C}$. Large circles denote optimal m values for each heat flux.

for each disk heat flux noted in Fig. 2.6 are provided in Fig. 2.5. For minimal flow power optimization, as was the case for minimal pressure drop optimization, increases in surface area are necessary for increases in disk heat flux. As noted previously, this can be achieved by either an increase in m or an increase in n_0 . However, increasing n_0 has the added benefit of increasing the cross-sectional flow area and thereby decreasing the flow rate. Because pressure drop is directly proportional to volumetric flow rate, the flow power is proportional to the square of the volumetric flow rate. Therefore, achieving the necessary increase in surface area via increases in n_0 yield better solutions and lower optimal m values when optimizing flow power as opposed to optimizing pressure drop. Consider the optimum m value in Fig. 6a for $q'' = 10 \text{ W/cm}^2$ for the case in which flow power is minimized. This is the same optimum m value (see dashed line) for $q'' = 100 \text{ W/cm}^2$ when pressure drop is minimized. Examination of the corresponding dashed lines in Figs. 2.5b- 2.5e reveals that all other optimized geometric parameters, n_0 , β and γ and w_m^+ , are identical for $q'' = 10 \text{ W/cm}^2$ based on minimized flow power and for $q'' = 100 \text{ W/cm}^2$ based on minimized pressure drop. The one heat flux in Fig. 2.5 for which this is an exception is 1000 W/cm^2 , in which case the two flow networks are identical. This is not unexpected, however, because of the combined need for both narrow channels and high flow rates for such high heat flux cooling results in the pressure drop being the primary contributor to the flow power.

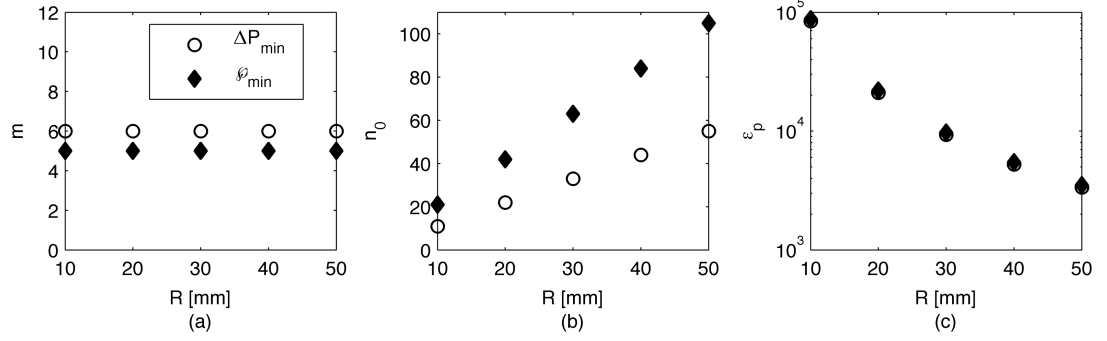


Figure 2.7: Minimized pressure drop and minimized flow power flow network parameters and performance coefficient versus R for $q'' = 200 \text{ W/cm}^2$, and $T_{w,max} = 70^\circ\text{C}$.

2.4.2.3 Parametric variation of disk radius

Figure 2.7 illustrates the change in m , n_0 , and ϵ_p as a function of disk radius for both minimized pressure drop and minimized flow power cases. Held fixed are the disk heat flux at 200 W/cm^2 and maximum wall temperature constraint at 70°C . As was noted in the discussion of Fig. 2.5 in which R was fixed and q'' was varied, increases in n_0 were achieved at the expense of reduced values of m . However, in Figs. 2.7a and 2.7b, values of m remain constant and independent of R while values of n_0 are observed to increase linearly with R . In addition to m , the remaining geometric parameters β , γ , and w_m^+ remained unchanged with R . Therefore, as the disk radius is increased, the lengths of the individual branch levels increase. To fill the outer periphery with channels within the center-to-center spacing constraints provided, i.e., $1.5 < b < 2.5$, the number of n_0 channels are also increased. Increases in n_0 , which would pose a problem at the inlet plenum for

fixed diameter disks, do not pose a problem in this case because the inlet plenum radius is a fixed ratio of the disk diameter. The inlet plenum constraint embedded in Eq. 2.4 can be rearranged to yield

$$\frac{n_0}{R} = \frac{2\pi a \beta^m}{w_m} \quad (2.13)$$

to predict the slopes observed in Fig. 2.7b. Recall that $a = 0.1$ for the present study and the fixed geometric parameters are available from Fig. 2.5. For the case of minimized pressure drop, where β is 0.666, w_m is 50 μm and m is 6, the predicted and measured slope is 1.1 mm^{-1} , compared with a slope of 2.1 mm^{-1} for the minimized flow power case in which β , w_m , and m are 0.7, 50 μm , and 5, respectively.

The benefit-to-cost ratio, ϵ_p , is shown on semi-logarithmic axes in Fig. 2.7c as a function of R . Substituting Eq. 2.12 into Eq. 2.9 to eliminate ΔP , followed by elimination of volumetric flow rate using the relation proportional to $q''2\pi R$ yields $\epsilon_p \propto 1/R^2$. This mathematical relationship is observed in Fig. 2.7c. The cause for the observed decrease in heat sink performance is due to the increase in flow resistance resulting from an increase in disk radius and, hence, the total length of the flow network. This increase in pressure drop outweighs the benefit of decreased flow rate that accompanies increases in both convective surface and cross-sectional flow areas. The convective surface area increases with both R and n_0 , whereas increases in cross-sectional flow area result from increases in n_0 .

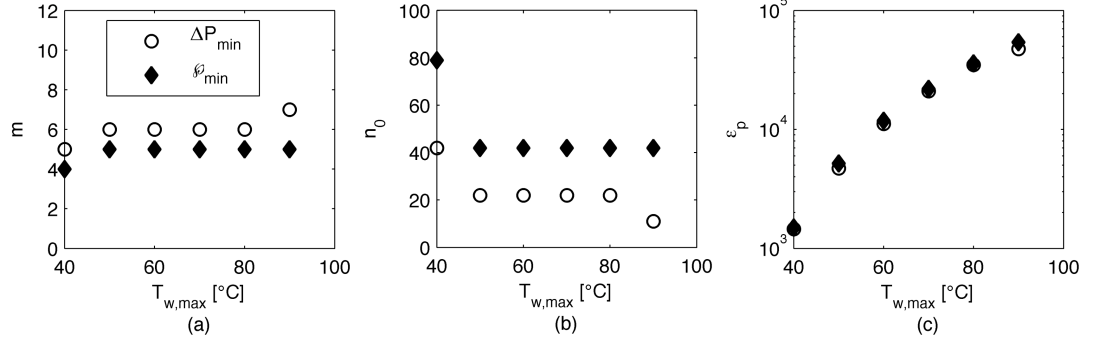


Figure 2.8: Minimized pressure drop and minimized flow power flow network parameters and benefit-to-cost ratio versus $T_{w,max}$ for $q'' = 200W/cm^2$, and $R = 20mm$.

2.4.2.4 Parametric variation of wall temperature constraint

Figure 2.8 shows that m and n_0 are invariant with $T_{w,max}$ between 50 and 90°C for the minimized flow power case and between 50 and 80°C for the minimized pressure drop case. At the lowest and most stringent temperature constraint considered, i.e. 40°C, more convective surface area and larger cross-sectional flow areas become necessary, resulting in optimum geometries with lower m and higher n_0 values. Conversely, at the highest and less stringent wall temperature constraint of 90°C, the minimized pressure drop flow geometry has an increased number of branch levels. The increased value of m and corresponding decreased value of n_0 result in a sufficient convective surface area to constrain the wall temperature. On the other hand, this combination of m and n_0 would be less ideal for achieving an optimal flow power because of the increase in flow rate needed to constraint the wall temperature and the subsequent increase in flow power.

2.4.2.5 Parametric variation of terminal channel width

Because the lower limit of the terminal channel width constraint was repeatedly reached during optimization, it is instructive to investigate lower terminal channel widths. Additional minimum terminal channel width constraints of 10, 20, 30, and 40 μm were considered. Figure 2.9 shows the minimized flow power as a function of m terminal channel width constraints between 10 and 50 μm . For each value of $w_{m,min}$ in Fig. 2.9 there is an optimal value of m , indicated with a circle. As $w_{m,min}$ decreases, more channels can be fit around the periphery of the disk. To connect terminal channels to the inlet plenum requires either an increase in the number of branch levels or an increase in the number of 0^{th} level branches. As mentioned previously, increases in m primarily yield decreases in pressure drop whereas increases in n_0 result in decreases in flow rate. Because pressure drop is inversely proportional to the hydraulic diameter to the fourth power for a fixed laminar flow rate, minimizing pressure drop via increases in m becomes increasingly more important than by reductions in flow rate as the minimum terminal channel constraint is reduced.

For 10, 20, and 40 μm minimal channel width constraints, there is an anomalous order of magnitude increase in flow power with a unit increase in m at the higher values of m . The primary reason for this is that as the terminal channel width gets smaller while the number of branch levels increases, the spacing constraints inside the disk severely reduce the number of valid flow network configurations. In particular, the values of γ associated with the anomalously high flow powers

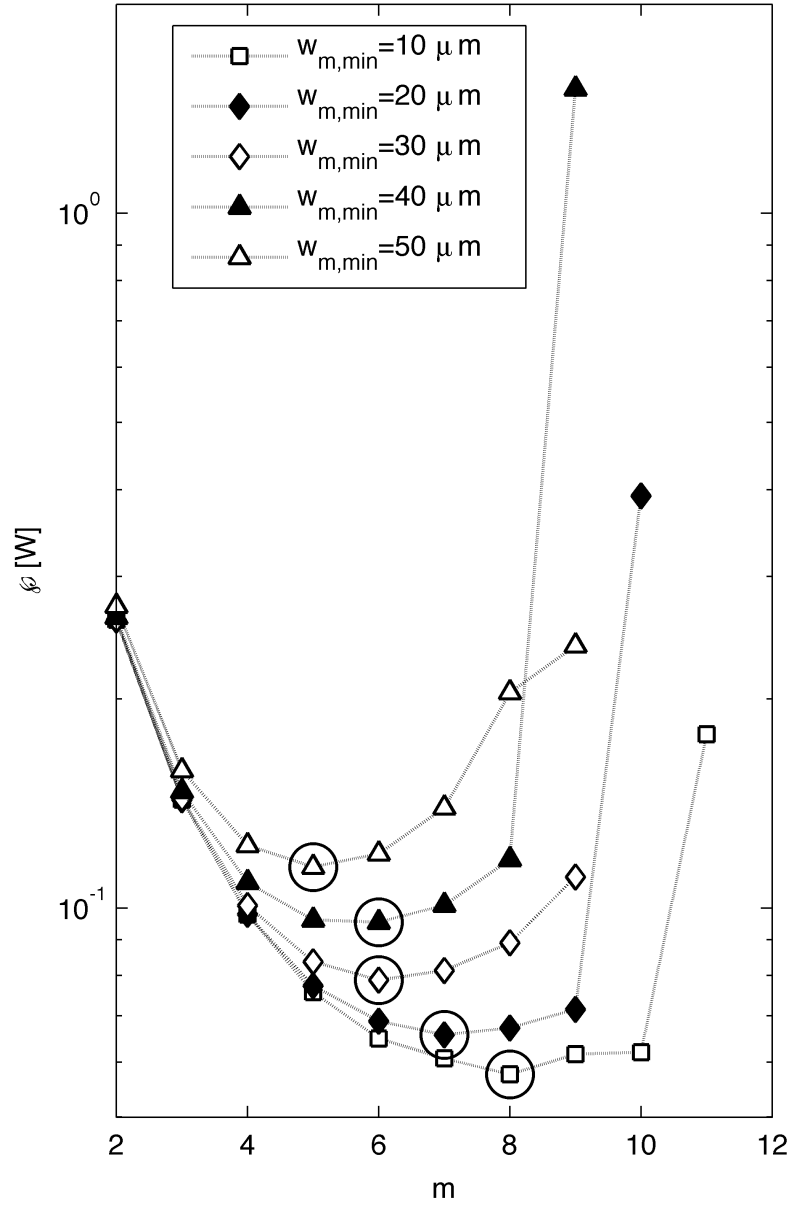


Figure 2.9: Minimized flow power versus m as a function of minimal channel width constraint for $q'' = 200W/cm^2$, $R = 20mm$, and $T_{w,max} = 70^\circ C$.

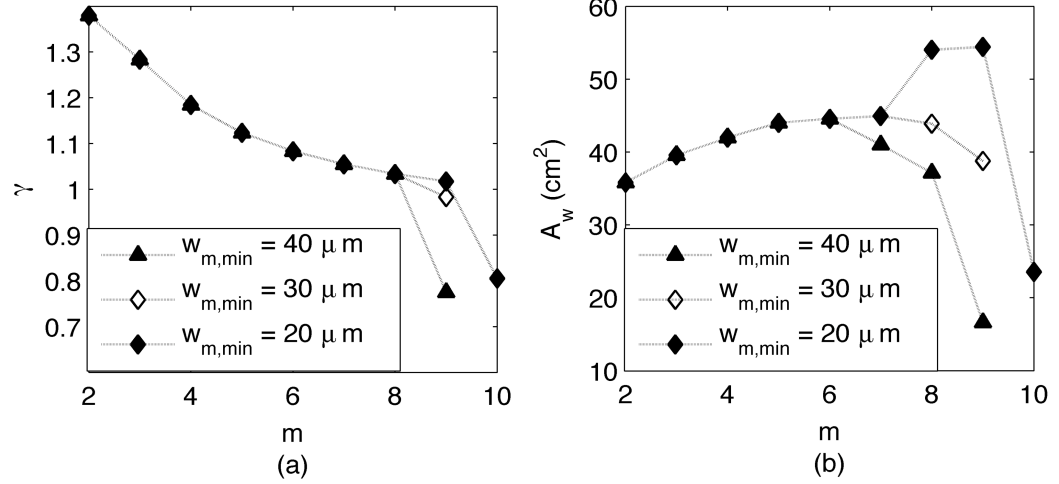


Figure 2.10: Length scale ratio and total convective surface area versus m and a function of minimum terminal channel widths for $q'' = 200W/cm^2$, $R = 20mm$, and $T_{w,max} = 70^\circ C$.

in Fig. 2.9 are generally very small. This is evident from Fig. 2.10a, in which γ is plotted as a function of m for the $w_{m,min} = 20, 30$ and $40 \mu m$ constraints in Fig. 2.9. Although the $30 \mu m$ constraint does not exhibit this anomalous behavior it does provide a nice comparison. The value of γ reduces significantly at the values corresponding to the anomalous flow powers, which are $m = 10$ and $m = 9$ for the $20 \mu m$ and $40 \mu m$ constraints, respectively. As noted from Fig. 2.10b there is also a significant decrease in convective surface area at these anomalous points. This is primarily a result of the significant decrease in the length scale ratio, γ , observed in Fig. 2.10a. The consequence of a decrease in γ on the convective surface area is evident from comparing Fig. 1.1a with $\gamma = 0.7$ to Fig. 1.1b with $\gamma = 1.4$. Recall that a decrease in convective surface area requires higher flow

rates to achieve the cooling necessary to maintain the maximum wall temperature constraint. Although larger values of m and smaller values of γ generally result in lower pressure drops, this advantage is overwhelmed by the disadvantage of high pressure drops resulting from high flow rate through the very narrow channels.

The geometric parameters that correspond with the optimal values of m for minimized flow power and minimized pressure drop, rerun with lower $w_{m,min}$ values, are shown in Fig. 2.11 as a function of $w_{m,min}$. Unlike in Fig. 2.5 where the value of m was always less than or equal, and n_0 , β and γ were always greater than or equal, for a flow network optimized for a minimal flow power compared to one optimized for a minimal pressure drop, there are no such clear trends in Fig. 2.11. The divergence from these previously observed trends occurs for $w_{m,min}$ less than 30 μm . Evident from Fig. 2.11e, the terminal channel width for flow networks in which pressure drop was minimized did not drop below 30 μm , even though allowed for by the constraints. This is mainly due to the considerable increase in pressure drop at these smaller channel dimensions. The impact of which is observed to influence the other optimized design parameters, including m , β and γ . For very narrow channels, flow power is more easily minimized with very low flow rates resulting from high convective surface and cross-sectional flow area networks expected from the m , n_0 , β and γ combinations shown in Fig. 2.11. Minimizing based on pressure drop also negatively impacts the benefit-to-cost ratio for $w_{m,min} < 30$, indicating that $w_{m,min}$ plays a more crucial role in optimizing for ΔP than for ϕ .

As a final note, as is event from Figs. 2.5d and 2.11d, no optimized flow network configuration has a length scale ratio less than unity. Although pressure drop

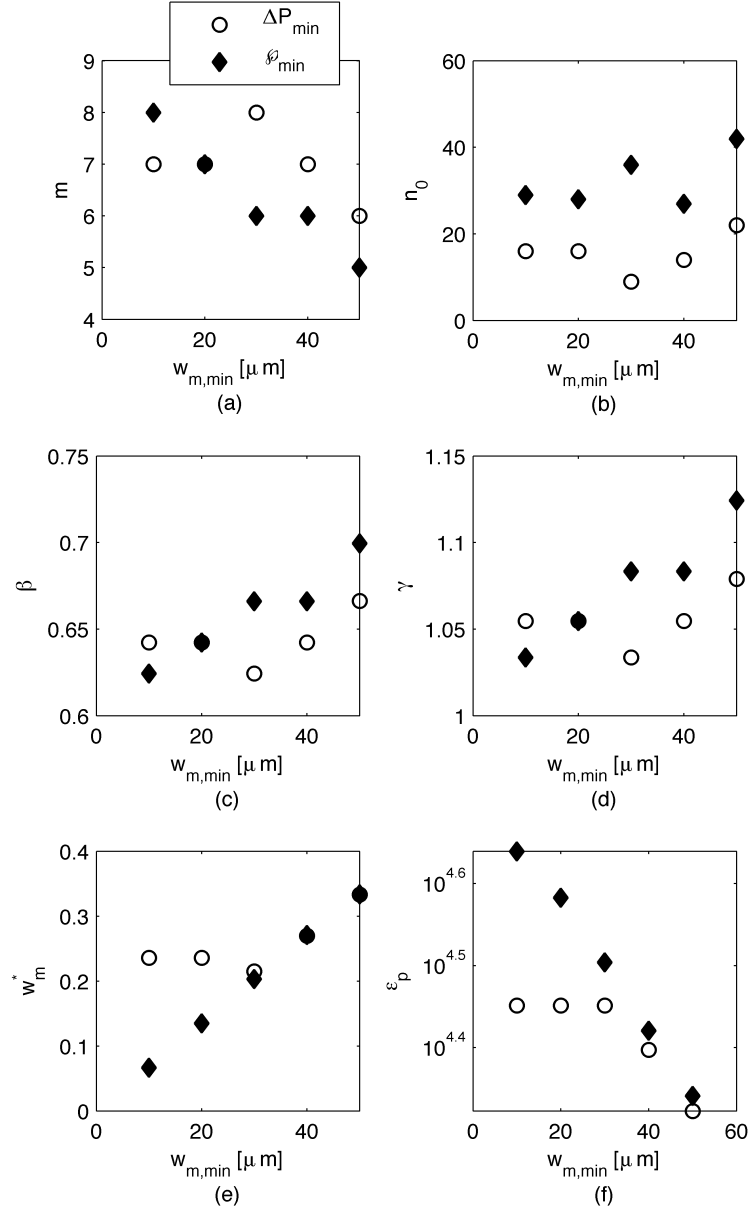


Figure 2.11: Minimized pressure drop and minimized flow power flow network parameters and performance coefficient versus $w_{m,\min}$ for $q'' = 200\text{W}/\text{cm}^2$, $R = 20\text{mm}$, and $T_{w,\max} = 70^\circ\text{C}$.

has previously been minimized using lower length scale ratios in absence of imposing a maximum wall temperature constraint [5], this wall temperature constraint severely limits the valid range of γ . The value of γ is also restricted by the internal and peripheral spacing constraints. For example, for $\gamma > 1$, the internal spacing constraint was most often violated at the bifurcation between the $k = 0$ and 1 branch levels, whereas for $\gamma < 1$, the violations occurred between the $k = (m - 1)$ and m branch levels.

2.4.3 Summary

Evident from Figs. 2.5 and 2.11, the steepest descent optimized parameters, β , γ , and w_m , based on flow power minimization vary by less than 7% over the range of q'' and w_m studied. This limited variation in β , γ , and w_m over a wide range of heat fluxes is due to the considerable influence of the fabrication constraints on the optimal flow network parameters. Recognizing the influence of fabrication constraints on network design, coupled with a recognition of the time necessary to program and implementation optimization codes, Pence [17] proposed a simple and purely geometric fractal-like flow network design approach for single-phase flows in disk shaped heat sinks. Using average values of 1.08 for γ and 0.67 for β , a comparison of ϵ_p for the optimized configurations with those from the geometric-based design yielded values within 11%. These geometrically based results are repeated here in Tables 2.1 and 2.2 and designated by GB. Note that although the values for m and n_0 in Table 2.2 are significantly different between the geometrically based

design and steepest descent approach for heat fluxes of 500 and 1000 W/cm^2 , the flow powers and benefit-to-cost ratios reported in Table 2.1 are within 13%. These results indicate the possibility that a number of optimal flow networks exist that can yield very similar performance results. It also appears that a single geometrically based flow network, using average length and width ratios determined by fabrication constraints, may provide near optimal performance without the need for extensive optimization.

2.5 Conclusions

Using a previously validated one-dimensional model for predicting pressure drop and wall surface temperatures, fractal-like flow networks in disk-shaped heat sinks were optimized. Flow geometries resulting from minimizing pressure drop were compared to those from minimizing flow power. A direct numeric search to find minimal flow powers was conducted as a means of validating the results of the gradient-based steepest descent search. Results between these two techniques are within 10%. A genetic algorithm was also employed as a potential alternative to the steepest descent technique. Compared with the direct search, these results were also within 10% for the 10 and 1000 W/cm^2 heat fluxes, but within 25% for the 500 W/cm^2 case. In addition, the time to a converged optimal solution was considerably longer for the genetic algorithm than for the steepest descent.

The heat flux, maximum wall temperature constraint, and geometrical constraints had a significant impact on the flow network design. In most cases, the

narrowest terminal channel widths were also found to be preferable. Benefit-to-cost ratios were found to decrease with increases in disk heat flux, in disk radius and in the minimum channel width constraint.

For higher heat flux cooling applications, fewer branch levels with many channels emanating from the inlet plenum are also found to be desirable. The optimal flow networks at higher heat fluxes exhibit larger β and γ values than at lower fluxes. These fractal-like flow networks allow for the highest convective surface and cross-sectional flow areas, a combination desired for high cooling capabilities with the lowest flow power requirements.

Results from optimization demonstrate the vital importance of employing constraints imposed by fabrication techniques in addition to the thermal and flow variables in a multivariable optimization algorithm. On the other hand, results from the geometrically based flow network design procedure outlined in Pence [17] suggest that good designs exist without a need for optimization. This is the case because the average gamma and beta ratios used by Pence [17] were significantly influenced by the geometric constraints.

Chapter 3 – Optimization of Two-Phase Fractal-like Branching Channel Heat Sinks

3.1 Introduction

Disk-shaped heat sinks with fractal-like branching microchannels were first proposed by Pence [4] as a means to reduce both the pressure drop and the maximum streamwise wall temperature difference observed in parallel microchannel heat sinks. A fractal-like branching flow network, inspired by nature, was studied. Representative fractal-like flow networks in a disk-shaped heat sink configuration are shown in Fig. 1.1. Flow enters the network from the inlet plenum located at the center of the disk and exits at the periphery. Each channel emanating from the inlet plenum bifurcates into two narrower channels, each of which in turn bifurcates. This repetitive pattern is considered fractal-like because the ratio of the channel widths and channel lengths between the consecutive branch levels are fixed. Fractal-like flow networks have been studied under single-phase and two-phase, or boiling conditions.

Bowers and Mudawar [18, 19] were among the first to study both the heat transfer and pressure drop characteristics of flow boiling in microchannels, the most significant finding was that the same heat transfer rate could be achieved with a lower two-phase flow rate compared to single-phase flow. However the

increase in heat transfer does not come without penalty, Daniels [20] found that for certain cases of heat flux and mass flow rate, the two-phase pressure drop is larger than the single phase, increasing the cost. The cost can be minimized through an optimization algorithm such as the ones employed by Heymann et al. [21]. The goal of this study is to assess the performance of a two-phase flow optimization algorithm under different constraints, how those constraints affect the performance of the algorithm and the heat sink design.

The gradient-based optimization algorithm used in the present study employs the one-dimensional model developed by Daniels [20] for predicting pressure and wall temperature distributions in two-phase fractal-like branching channel networks, assuming laminar flow and uniform wall heat flux. Hydrodynamic and thermal boundary layers were assumed to redevelop following each bifurcation. Daniels [20] experimentally validated the model for adiabatic boiling flow by reducing the exit pressure to sub-atmospheric conditions. Daniels [20] found that the pressure drop predictions agreed well with the experimental results. However, the void fraction distribution in the channel was not explained well by the model. Edward [22], studying the void fraction distribution for diabatic flow in branching channels found similar results, but noted that the total void fraction (integrated over the entire tree and averaged over time) predicted by the model and the experimental results agreed well, and the distribution discrepancy was due to instabilities forcing vapor backwards. Similarly, Cullion, et al. [23] found that when liquid momentum of the flow was not sufficient to counter the capillary forces an entrained bubble, that there was blockage in the branching channels. The constraints of

the present optimization approach seek to mitigate the issue of instabilities arising from the non-favorable force balance.

Kandlikar [24], identified the surface tension, evaporative momentum and liquid inertia forces, and noted that a positive downstream balance of the forces results in stable flow. Kandlikar [24] restricted the analysis to a straight channel, where the net surface tension force of a straight, elongated bubble is zero. Lee et al. [25] utilized geometries that result in a positive net force balance in the downstream direction. The geometries studied by Lee et al. [25] include expanding cross sectional area channels and channels with inlet orifices. For more complicated geometries the surface tension forces may not be zero, such as the fractal-like channels of the present study, although it should be noted that some a priori knowledge of the exact vapor distribution in the channel is required to model the evaporative force in the channel. However, most importantly, the reverse flow instability was mitigated in channels with decreasing downstream flow resistance.

There is an upper limit to the capacity of two-phase flow, at which any further increase in heat flux results in catastrophic failure to the system due to extremely high wall temperatures. Critical heat flux (CHF) is an additional phenomenon that constrains the operation conditions of the fractal-like branching channel heat sink. The knowledge of the conditions at which CHF occur is essential to avoiding it, Qu and Mudawar [26] developed a correlation for CHF in a straight microchannel and was found to agree with experimental data to a mean-average error of approximately 4%. Revellin and Thome [27] constructed a numerical model of ordinary differential equations to account for the interfacial wave troughs in contact with

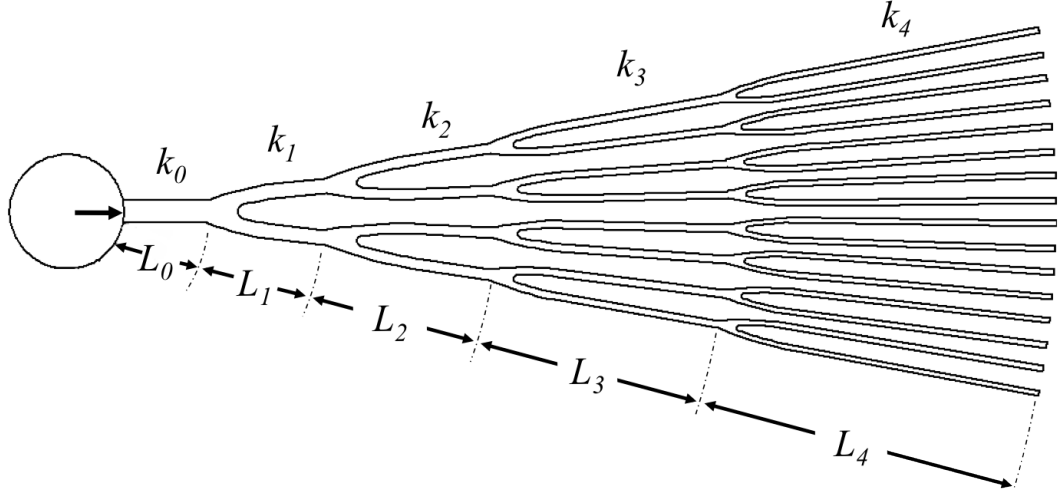


Figure 3.1: Illustration of a fractal-like branching channel tree

the wall, the CHF condition where the film thickness is still greater than zero. Other than avoiding CHF certain designs may increase it, allowing for higher heat fluxes. Revellin et al. [28] applied the model of [27] to a disk-shaped heat sink of constructal design and found the CHF to increase as the number of n_0 channels increased. Mauro et al. [29] also found the CHF to increase by utilizing a split flow system, in which there is a single outlet and one inlet. The present study avoids CHF by constraining the mass flow rate to greater than or equal to a critical value derived from a modified model of Revellin et al. [28], different than the model used to predict the two-phase heat transfer and pressure drop in the fractal-like branching microchannels.

3.2 Model

3.2.1 Fractal Geometry

The geometry of the fractal-like branching channels is defined solely by the disk radius R , channel height H , terminal channel width w_m , length and width ratios γ and β , the number of branching levels m , and the number of 0^{th} level channels n_0 , all other geometric parameters are derived from these variables. Each branch level is indexed by k , beginning with the 0^{th} and m^{th} level connecting the tree to the inlet and exit plenum, respectively. The structure of the fractal-like branching channel geometry is best illustrated in Fig. 3.1, the half tree, which has the same increasing length ratio of 1.41 as the geometry of Fig. 1.1(b). The fixed width ratio is

$$\beta = \frac{w_{k+1}}{w_k} \quad (3.1)$$

with the fixed length ratio

$$\gamma = \frac{L_{k+1}}{L_k} \quad (3.2)$$

The total channel length, L_{tot} is determined by subtracting the inlet plenum radius from the disk radius

$$L_{tot} = R - \frac{w_m n_0}{\beta^m 2\pi} = L_0 \sum_{k=0}^m \gamma^k \quad (3.3)$$

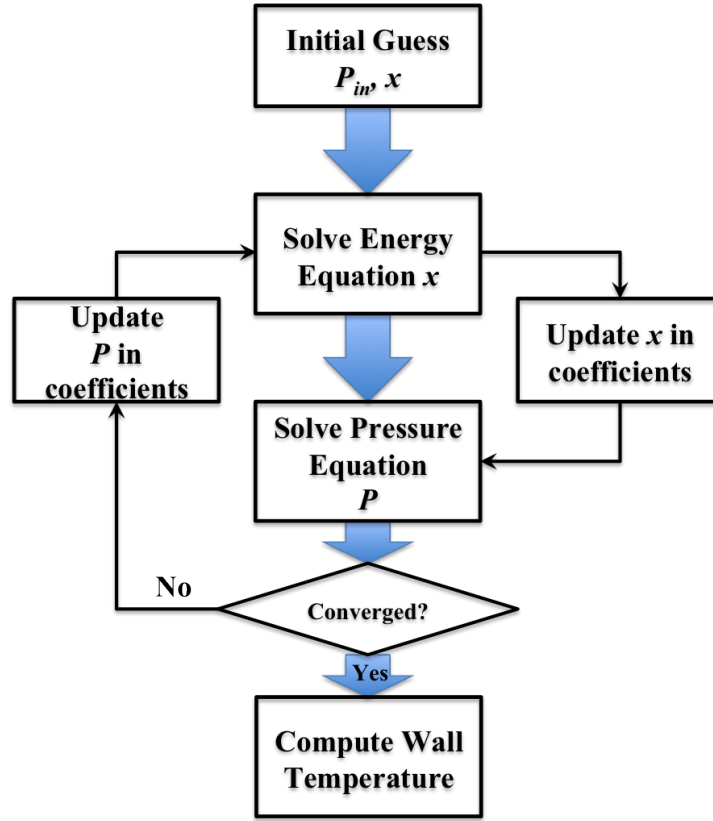


Figure 3.2: Flow chart of the two-phase flow algorithm for solving Eqs. 3.5-3.9

The 0^{th} level channel length can be ascertained from γ and the total channel length, L_{tot} , determined from Eqn 4.3 to get an expression for L_0

$$L_0 = \frac{L_{tot}}{\sum_{k=0}^m \gamma^k} \quad (3.4)$$

3.2.2 Pressure and temperature

The pressure and temperature profiles through the fractal-like microchannels are estimated with the two-phase flow model of Daniels [20] by solving the momentum and energy balance equations. The boundary conditions are a fixed, uniform inlet temperature and exit pressure as well as a uniform heat flux boundary condition imposed on the channels walls. The inlet temperature is set at 25°C, the exit pressure is set at 101.3 kPa-absolute (except under the maximum wall temperature constraint), and the disk heat flux is varied from 10 - 300 W/cm². The channels are made of silicon with a relatively high thermal conductivity of 130 W/m-K, thus the channel heat flux is assumed uniform and the equal to the disk heat flux multiplied by the disk area to channel area ratio.

The first of the two equations solved in the model is the energy balance, expressed in nodal form as

$$z_{i+1} = z_i + \frac{q''2(w + H)\Delta x}{\dot{m}h_{fg}} \quad (3.5)$$

The pressure is then solved for with the one-dimensional two phase momentum balance

$$P_{i+1} = P_i - \Delta P_a - \Delta P_f \quad (3.6)$$

Because the inlet temperature is in the sub-cooled liquid region, the algorithm of Pence [4] is used to compute the single phase pressure drop and temperature rise up to the location of incipient boiling where $T \geq T_{sat}$. From this point, and

onward downstream the two-phase pressure drop of Eqn. 3.6 is used to compute The change in momentum resulting from the expansion of vapor in the channel is characterized by the acceleration pressure drop, ΔP_a , derived in [20] and expressed as

$$\Delta P_a = -G^2 \left(\frac{z^2}{\rho_v \alpha} \Big|_{i+1} - \frac{z^2}{\rho_v \alpha} \Big|_i + \frac{(1-z)^2}{\rho_l (1-\alpha)} \Big|_{i+1} - \frac{(1-z)^2}{\rho_l (1-\alpha)} \Big|_i \right) \quad (3.7)$$

The frictional contribution to the two-phase pressure drop is expressed as

$$\Delta P_f = \frac{G^2}{\rho_l 2D_h} \int_i^{i+1} \phi_l^2 (1-z) \lambda_{lo} dz \quad (3.8)$$

where

$$\phi_l^2 = 1 + \frac{C_{LM}}{X} + \frac{1}{X^2} \quad (3.9)$$

with C_{LM} as the phase interaction parameter and X^2 as the Lockhart-Martinelli parameter [30]. The void fraction, α , is computed from the quality, z , by using the separated flow model of Zivi [31]. Because Eqns. 3.7 and 3.9 are dependent on z , the energy and pressure drop equations are coupled and must be solved iteratively.

The iterative algorithm for which the temperature and pressure are solved for is illustrated in Fig. 3.2. An initial guess of the inlet pressure, P_{in} , provides a starting point for which to evaluate properties in the energy equation and compute the pressure drop. Eqns. 3.5 and 3.6 are evaluated from the inlet plenum to the

outlet plenum, at which point the computed exit pressure is compared to the actual exit pressure at which point the inlet pressure guess is increased or decreased to compensate for the computed outlet pressure being lower or higher than the actual outlet pressure being too low or high.

3.3 Optimization

3.3.1 Objective function

The outcome of the optimization algorithm is to minimize the non-dimensional performance parameter, ϵ_p , expressed as the flow power divided by the heat transfer. The objective function is expressed mathematically as

$$\min \left\{ \frac{1}{\epsilon_p} = \frac{Q\Delta P}{q''\pi \left(R^2 - \left(\frac{n_0 w_m}{2\pi\beta^m} \right)^2 \right)} \right\} \quad (3.10)$$

The volumetric flow rate, Q , heat flux, q'' , and geometry are all specified while the pressure drop, ΔP is predicted from the aforementioned 1-D model. These quantities form the objective function of the optimization algorithm, where the ratio of flow power to heat transfer is minimized.

A gradient-based numerical optimization algorithm was employed, identical to that of [21], varying β and γ . Several starting points to the gradient-based algorithm were considered to test the convergence to local minima, no local minima were found, a result which will be confirmed in Section 3.3.3 was corroborated by

Heymann et al. [21] for single-phase flow.

3.3.2 Constraints

Geometric constraints

Four fabrication constraints on the network geometry are imposed in the optimization algorithm. The first is that the inlet plenum radius is restricted to less than or equal to 10 percent of the disk radius. The parameter a represents the ratio of inlet plenum radius to disk radius. The inlet plenum radius is determined from its circumference, which is equal to the number of 0^{th} level channels, n_0 , times their width, w_0 . The value of w_0 can be evaluated from

$$w_0 = \frac{w_m}{\beta^m} \quad (3.11)$$

yielding the following first geometric constraint, which constrains the inlet plenum diameter

$$\frac{n_0 w_m}{\beta^m} \leq 2\pi R a \quad (3.12)$$

The second and third geometric constraints are both applied to the terminal channel geometry, i.e. there must be sufficient material between channels to support bonding of a top disk to finalize the flow network. A dimensionless parameter, b , is defined as the center-to-center spacing between the terminal (m level) channels divided by the width of the terminal channels, w_m . This parameter is allowed to vary between a minimum value, b_{min} , and a maximum value, b_{max} . A similar spac-

ing parameter, d , is defined for the levels $k = 0$ through $m-1$, and for the present analysis is set equal to b_{min} .

To ensure that the terminal channels are adequately spaced, the circumference of the disk divided by b_{max} , and by b_{min} , must bracket the sum of the widths of the terminal channels as in

$$\frac{2\pi R}{b_{max}} < 2^m n_0 w_m < \frac{2\pi R}{b_{min}} \quad (3.13)$$

Eqn. 3.13 imposes two separate constraints on the terminal channel spacing at the periphery of the disk. For the remaining branch levels, i.e. $k = 0$ through $m-1$, a minimum center-to-center spacing of $w_k d$ is ensured between channels of width w_k , if the following equation

$$\left(\frac{2\pi R \beta^m}{n_0 w_m} - 1 \right) \frac{\sum_{i=0}^k \gamma^i}{\sum_{i=0}^m \gamma^i} > d (2\beta)^{k+1} - 1 \quad (3.14)$$

is valid. Eqn. 3.14 represents the fourth geometric constraint, a constraint on the internal channel spacing, which is imposed at each level k . Although ranges of m and n_0 are specified, the upper limit on m is constrained by

$$2^m < \frac{2\pi R}{b_{min} n_0, \min w_m, \min} \quad (3.15)$$

as determined from Eqn. 3.13. To summarize, the following needs to be specified by the flow network designer: (I) single values for geometry variables a , b_{min} , b_{max} , d , H , and R , (II) ranges for geometry values m , n_0 , β , γ , and w_m , and (III) a

single value for q'' . Within the parametric space defined by ranges of β , γ , and w_m there may be numerous possible flow configurations. Of these, many will not adhere to the geometric constraints imposed by Eqns. 3.12-3.15 and are considered invalid geometries. In addition to the geometric constraints, there must be some minimum flow rate, otherwise an optimization algorithm will yield solution with the absolute minimum flow power, i.e. a zero mass flow solution.

Maximum wall temperature

The volumetric flow rate must be constrained in order to avoid a null solution to the optimization problem, one way to achieve this is to set a maximum wall temperature to anticipate a maximum junction temperature, similar to Heymann et al. [21]. Wall temperature is inversely proportional to the volumetric flow rate, Q , therefore a maximum wall temperature sets a minimum volumetric flow rate. The minimum volumetric flow rate that satisfies the maximum wall temperature cannot be found explicitly because wall temperature is an output of the 1-D model, an iterative method is required.

A volumetric flow rate that satisfies the maximum wall temperature is found via the bisection method since trials with the shooting method proved unstable. Moreover, the outlet pressure was set to 6 kPa-absolute in order to effectively lower the saturation temperature to 36°C . While a refrigerant, such as R-134a has a similarly low saturation temperature, water was used because of its high latent heat; $h_{fg, \text{water}} = 2.4 \times 10^6 \text{ J/kg}$ compared to $h_{fg, \text{R134a}} = 2.2 \times 10^5 \text{ J/kg}$.

A gradient-based optimization of the two-phase objective function was performed subject to the geometric constraints and maximum wall temperature limit

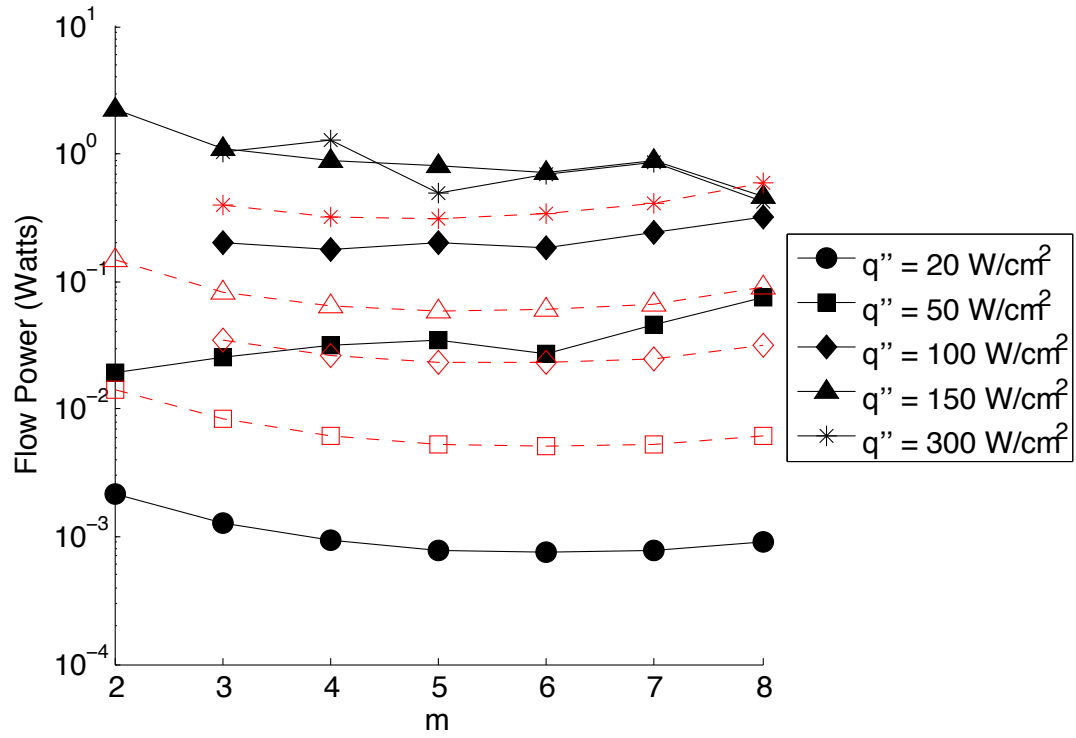


Figure 3.3: Comparison of the two-phase and single-phase optimization results subject to a maximum wall temperature of 70°C , filled markers represent the two-phase algorithm and open markers with a dashed - - line represent the single-phase algorithm results from [21]

of 70°C . The results from the single phase optimization of Heymann et al. [21] and the two-phase optimization are compared in Fig. 3.3 for heat fluxes ranging from 20 to 300 W/cm^2 .

The results of Fig. 3.3 indicate that even at sub-atmospheric pressures, the two-phase heat sink does not perform better than the single phase. With the exception of $q'' = 20\text{ W/cm}^2$, which resulted in single phase flow no matter what the exit pressure, all of the two-phase flow powers are greater than the single-phase, simply due to the increase in pressure drop for a given flow rate. Upon closer inspection of the data, it was found that the location of the maximum wall temperature was in the single phase region, immediately upstream of the point of incipient boiling, meaning that for all intents and purposes the maximum wall temperature occurs in the single-phase region, thus the volumetric flow rates will be similar. However, the advantage of two-phase flow (lower flow rate for the same heat transfer) is not taken advantage of because the minimum mass flow rate is set by a constraint in the single-phase regime, up to the point of incipient boiling. The larger two-phase pressure drop is what increases the flow power for the same heat flux and number of k levels.

Clearly, when a maximum wall temperature constraint of 70°C is applied to the design, there is no justification for two-phase flow as evident from the increase in flow power between the single-phase and two-phase devices in Fig. 3.3. However, if wall temperatures approximately 10°C higher than the saturation temperature are acceptable, the volumetric flow rate of the device may be constrained differently.

Critical heat flux

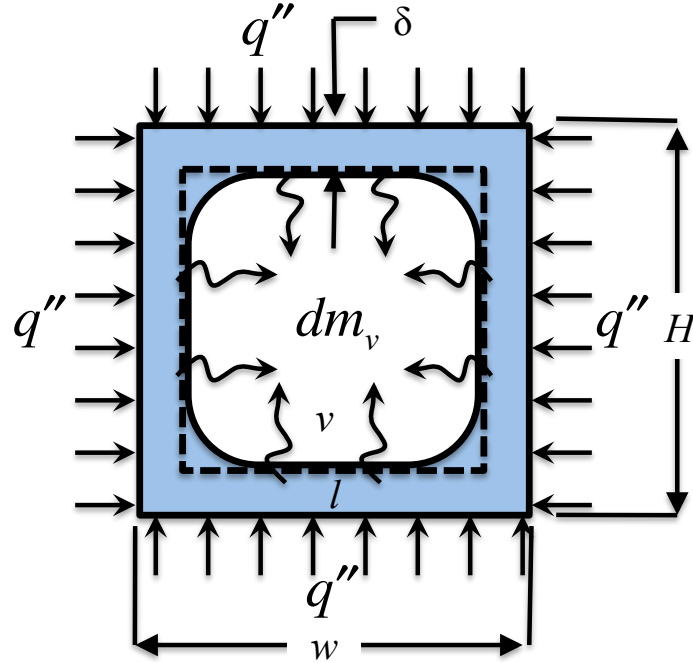


Figure 3.4: Illustration of liquid film and vapor core control volumes in a cross-sectional view of the channel used to formulate the critical heat flux model, the shaded region represents what the film in a rectangular channel may actually look like while the dashed line, - -, represents the approximation of the liquid-vapor boundary, flow is going through the page

The minimum mass flow rate through the fractal-like branching channel network may also be set by the critical heat flux (CHF) constraint, i.e. for a given heat flux, the mass flow rate must be great enough as to avoid CHF. A modified model of the CHF model first proposed by Revellin and Thome [27] is used to estimate the mass flow rate required for the applied heat flux to be equal to 50% of the CHF. The present algorithm differs from [27] in that no pressure drop is computed and the model is applied to rectangular channels rather than circular

used in the analyses of [27] and [28].

The model estimates CHF by solving the mass and energy equations describing the decay of liquid film surrounding the vapor core. Mass is transferred by phase change from the liquid film to the vapor core and the velocities and areas of the respective regions change along all along the length of the channel. The total mass flow rate that results in a zero liquid film thickness at the exit of the channel is assumed to be critical heat flux. The control volumes for the CHF model are illustrated in Fig. 3.4, the liquid film region is assumed to be a rectangular annulus with uniform film thickness, δ , the area of the liquid region can be solely characterized by the channel dimensions and δ as

$$A_l = wH - (w - 2\delta)(H - 2\delta) \quad (3.16)$$

and likewise the vapor area is expressed as

$$A_v = (w - 2\delta)(H - 2\delta) \quad (3.17)$$

Like the model of [27], the first is determined from the energy balance in the liquid region. Re-writing Eqn. 3.5 in differential form results in

$$\frac{d(\rho_v A_v u_v)}{dx} = \frac{q'' 2(w + H)}{h_{fg}} \quad (3.18)$$

Additionally, the conservation of mass at the interface between the liquid and vapor regions results in $d(\rho_v A_v u_v) = -d(\rho_l A_l u_l)$, and yields the second equation,

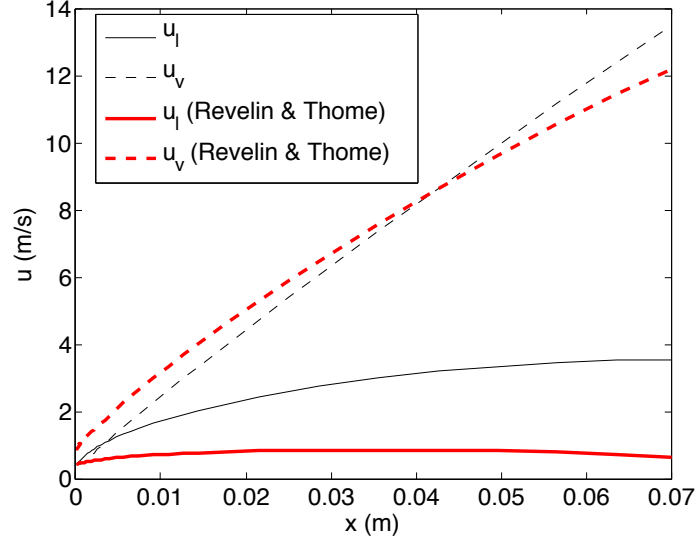


Figure 3.5: Velocity profile as a function of streamwise channel distance for both the three-equation and five-equation [27] ODE models for a circular tube 70 mm long, 500 μm in diameter, with a mass flux of 500 $\text{kg}/\text{m}^2\text{-s}$ and R-134a as the working fluid

similar to [27] as

$$\frac{d(\rho_l A_l u_l)}{dx} = \frac{-q'' 2(w + H)}{h_{fg}} \quad (3.19)$$

While Revellin and Thome [27] used the pressure drop equations and Young-Laplace equation to couple the velocity to pressure and film thickness to pressure, respectively, the present study simply uses the conservation of mass of the entire channel to link the film thickness and velocities. Because there is no mass source inside the channel, the derivative of the total mass flow rate with respect to the axial distance, x is equal to zero, i.e. $d\dot{m}/dx = 0$, this is expressed explicitly in

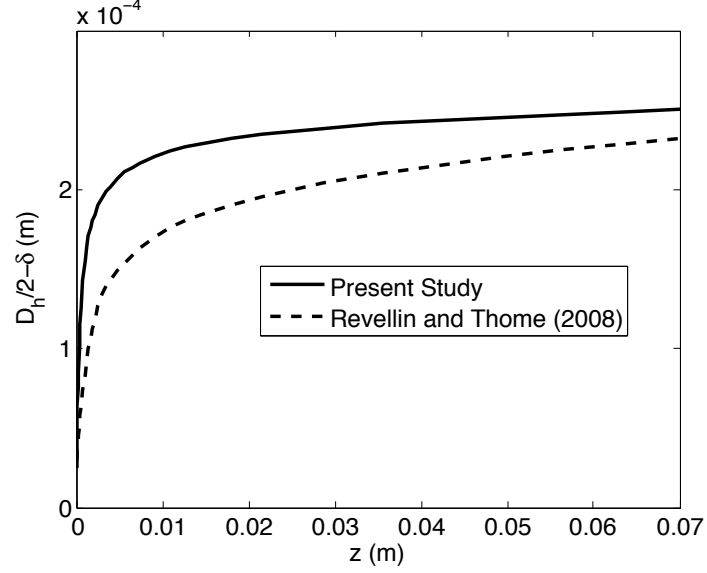


Figure 3.6: Liquid-vapor interface profile as a function of streamwise channel distance for both the three-equation and five-equation [27] ODE models for a circular tube 70 mm long, 500 μm in diameter, with a mass flux of 500 kg/m^2-s and R-134a as the working fluid

terms of u_l , u_v and δ and their derivatives as

$$\frac{d\dot{m}}{dx} = 0 = \rho_v u_v \frac{dA_v}{d\delta} \frac{d\delta}{dx} + \rho_v A_v \frac{du_v}{dx} + \rho_l u_l \frac{dA_l}{d\delta} \frac{d\delta}{dx} + \rho_l A_l \frac{du_l}{dx} \quad (3.20)$$

The Eqns. 3.18-3.20 are linear in terms of du_l/dx , du_v/dx and $d\delta/dx$ and can be solved to form three first order ordinary differential equations (ODE) to be solved by a Runge-Kutta algorithm. The inlet conditions are $\delta_{in} = 0.45D_h$, $u_{l,in} = u_{v,in} = \dot{m}/\rho_l A_{l,in}$ which are the same employed by Revellin and Thome [27]. The

relationships employed by Revellin et al. [28] at the branch levels are also used in the present study; immediately following each bifurcation, the conservation of void fraction and conservation of quality are used to determine the film thickness and velocities, respectively. In order to estimate the mass flow rate at CHF, the ODE solver is iterated using a bisection algorithm to find the mass flow rate that satisfies $\delta = 0$ at the exit plenum.

When the three-equation ODE model is compared to the five-equation ODE model for a circular tube 70 mm long, 500 μm in diameter, with a mass flux of 500 $\text{kg}/\text{m}^2\text{-s}$ and R-134a as the working fluid. Figure 3.5 illustrates the estimate liquid and vapor velocities as a function of streamwise channel distance of the three-equation model as well as Revellin and Thome's [27] five-equation model. One noticeable difference is the liquid film velocity of [27] is approximately 25% of the three-equation model, which is reasonable given the difference in film thickness, illustrated in Fig. 3.6, where the liquid-vapor interface profile is plotted as a function of streamwise channel distance.

At any point along the channel length, the film thickness of the five-equation model is at least twice that of the three-equation model, meaning that the mass flow rate of vapor between the three and five-equation models is not too different. In fact, the slope of the vapor mass flow rate for the five-equation model is approximately 80% of the three-equation model, which may be attributed to differing thermo-physical properties.

Stability

The stability criterion used in the present study is applied to the geometry of

the fractal-like channels, thus it is solely a geometric constraint. However, it is specifically applicable to two-phase flow and is different than the aforementioned geometric constraints that are for either single or two-phase flow. The fractal-like branching channel stability criterion was inspired by the work of Lee et al. [25] where geometries of decreasing downstream flow resistance resulted in stable flow. In the present study, a stable geometry is one in which the flow resistance of the channel decreases as the k level increase. First, the flow resistance, \Re , is defined as

$$\Re = \frac{\Delta P}{Q} \quad (3.21)$$

is an analogous to electrical resistance, where pressure represents the potential difference, and fluid flow, rather than electron flow is used. The flow resistance is re-written for Poiseuille flow, fractal-like geometry assuming uniform flow division at each bifurcation, and expressed purely in terms of the fractal-like geometric parameters as

$$\Re = \frac{\beta^{m-k} f Re \beta^{m-k} \left(\frac{w_m}{\beta^{m-k}} + H \right)^2 \nu}{w_m H (2w_m H)^2} \frac{\rho L_{tot} \gamma^k}{2 \sum_{k=0}^m \gamma^k} \quad (3.22)$$

The constraint for the stability criterion is established by taking the derivative of the flow resistance with respect to the k -level and setting equal to zero, i.e. $\Delta \Re / \Delta k = 0$. It is at this condition that the flow can be considered *quasi-stable*, that is, there is neither an increase or decrease in flow resistance as a function of the k level. Thus the criterion separates the regions of decreasing flow resistance and increas-

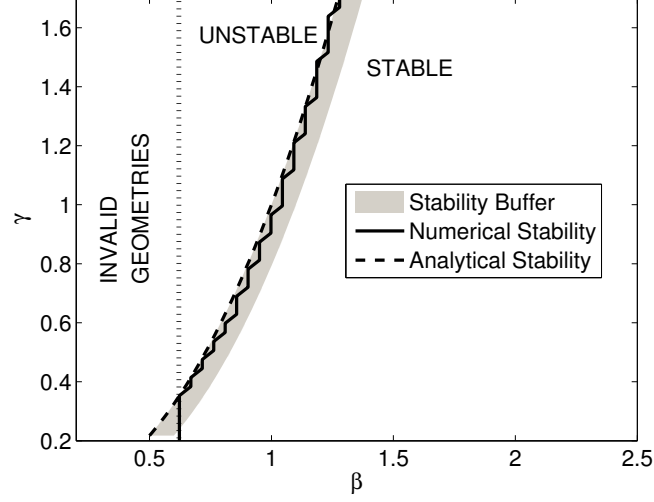


Figure 3.7: Comparison of the numerical and analytical stability criteria for $w_m = 100\mu m$ and $H = 150\mu m$

ing flow resistance as a function of an increasing k level. An explicit form of the stability constraint for γ in terms of β , w_m , and H is

$$\gamma \leq (\beta - 0.1)^{\left(\frac{3H+w_m}{H+w_m}\right)} \quad (3.23)$$

and is referred to as the analytical stability constraint. The extra buffer of 0.1 in Eqn. 3.23 is a safety factor designed to ensure stable and not quasi-stable conditions for the optimized solution, furthermore, the simplification was attained by setting $k = m$, resulting in the most restrictive constraint. To determine the mathematical validity of Eqn. 3.23, the actual flow resistances were computed using Eqn. 3.22 for a discrete range of β and γ , the change in flow resistance as a function of k level was computed and the solid line in Fig. 3.7 is referred to as the

Table 3.1: Range of optimization variables

	Min	Max
β	0.5	2.0
γ	0.5	2.0

numerical stability and represent the point at which the change in flow resistances is at or near zero. The analytical and numerical estimates for the flow stability criterion compare well, evident in Fig. 3.7, where Eqn. 3.23 is plotted with the numerical solution as a dashed line. The sawtooth pattern of the numerical solution is a result of the finite discretization of beta and gamma of ≈ 0.05 . The invalid geometries shown in Fig. 3.7 stem from the previously discussed geometric constraints. The analytical solution to the stability criterion provides another simple constraint that may be easily implemented in an optimization algorithm. Note however, that the stability constraint has not been experimentally verified and that the results presented in this analysis are strictly theoretical. Minimization of the two-phase objective function (Eqn. 3.10) is achieved by the quasi-newton (a gradient based) implementation of the Sequential Quadratic Programming algorithm. Optimization was performed for discrete values of m and n_0 and the range of the variables, β , γ , and w_m , are listed in Table 3.1. The remaining parameters were set as follows: $H = 150\mu m$, $R = 20mm$, $b_{min} = d = 1.5$, $b_{max} = 2.5$, $a = 0.1$, and $q'' = 0.5q''_{CHF}$.

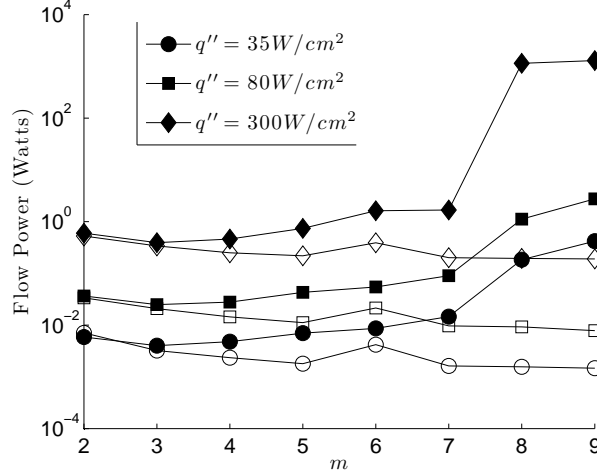


Figure 3.8: Optimized flow power versus m for disk heat fluxes of $35 - 300 \text{ W/cm}^2$, filled markers are two-phase flow networks subject to the stability constraint, open markers are two-phase flow networks *not* subjected to the stability constraint

3.3.3 Results and Discussion

Optimization with and without the stability constraint was performed over the aforementioned range and parametrical configuration. The preliminary results flow power results are illustrated in Fig. 3.8, results without the stability constraint are plotted as open markers, two-phase results with a stability constraint are plotted as open markers. The most obvious feature of Fig. 3.8 is the that an increase in applied disk heat flux, q'' , is accompanied with an increase in the optimum flow power, ϕ . The increase in optimum flow power with increase in heat flux was also observed by Heymann et al. [21], and was found to be a result of an increase in mass flow rate in order to meet the maximum wall temperature constraint. A similar effect exists in the two-phase results as well; the increase in heat flux

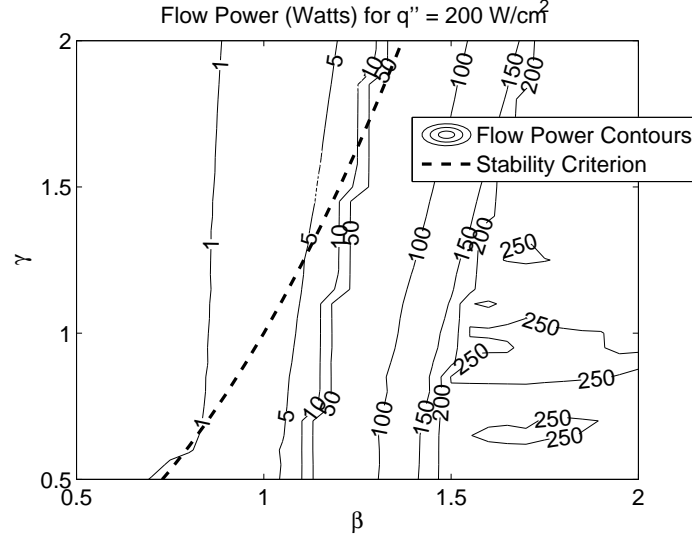


Figure 3.9: Map of flow power versus β and γ with stability constraint for $w_m = 100 \mu\text{m}$, $H = 150 \mu\text{m}$, $R = 20\text{mm}$, and $q'' = 200 \text{ W/cm}^2$

results in a higher mass flow rate in order to meet the CHF constraint, simply due to the fact that as heat flux is increased, the liquid film in the CHF model evaporates faster, thus more mass flow is required for the same CHF condition. The preliminary results of Fig. 3.8 also paint an account of the performance differences between two-phase flow with and without the stability constraint in the branching networks.

The two-phase flow networks (sans the stability constraint) consistently outperform the single-phase flow networks, simply due to the decrease in mass flow rate necessary for the equivalent heat transfer. The optimum m differs between the single and two-phase networks, but the improvement in optimum pumping power is between 8% and 62%. However, the stability constraint attenuates the

performance of the two-phase networks to the point that they are not competitive with the single-phase flow networks. Figure 3.8 illustrates the inferior performance of the two-phase networks constrained for stability according to Eqn. 3.23, the flow power of the stability-constrained networks is, for $m = 8$, an order of magnitude greater than the single-phase flow networks. It appears that there is a steep price in the form of flow power for stable flow and the best way to conceptualize the reason for the flow power increase is by mapping flow power as a function of β and γ .

The parameters of $w_m = 100 \mu m$, $H = 150 \mu m$, and $R = 20 mm$ were used to generate Fig. 3.9 with and $q'' = 200 W/cm^2$. Flow power is monotonically proportional and inversely proportional to β and γ , respectively, thus according to Fig. 3.9 the minimum flow power occurs for low β and high γ . Unfortunately, the stability constraint restricts the minimum, explaining the decrease in performance of flow networks subjected to the constraint. In summary, according to the analysis if stability is of great concern to the designer, a single-phase flow network is one to select in terms of performance. If the stability criterion is neglected, with the flow rate constrained to operate the device at 50% of CHF, the two-phase flow networks operate more efficiently than their single-phase counterparts.

The stability criterion of Eqn. 3.23 has been shown to increase the optimized flow power in the fractal-like heat sinks, thus it is necessary to evaluate whether or not stability is improved. The manner in which stability is assessed should be dependent on mass flow rate since the stability criterion of Eqn. 3.23 is purely geometric in nature, thus non-dimensional ratios of the relevant forces are used

to assess stability in the two-phase flow networks. The three important forces considered in this analysis are the evaporative expansion force, liquid inertia force and the surface tension force between the liquid and vapor interface. The first non-dimensional ratio, is the ratio of inertia to surface tension forces, commonly known as the Weber number, We . The second non-dimensional number used to assess the stability of the flow is the ratio of evaporative to inertial forces, used by Kandlikar [24] and Lee et al. [25], and is referred to in the present study as \bar{F}_k and expressed for each k level of the branching channel geometry as

$$\bar{F}_k = \frac{\rho_l}{\rho_g} \left(\frac{n_0 2^k q_w'' L_k 2(w_k + H)}{\dot{m} h_{fg}} \right) \quad (3.24)$$

The above expression assumes that all of the wall heat flux, q_w'' is absorbed in the latent heat exchange between the liquid and vapor phases at the front of a bubble occupying 99% of the length of the k^{th} channel. The parameter \bar{F}_k assumes a fixed vapor distribution within the microchannel, therefore it was not used as a stability constraint since any a priori knowledge of the vapor distribution is not available. In summary, the Weber number is proportional to the degree of stability, i.e., an increase in Weber number means more stable flow conditions; conversely, the parameter, \bar{F} is inversely proportional to the degree of stability – decreases in \bar{F} result in a lower evaporation force relative to the inertial force resulting in more stable flow conditions.

The maximum Weber number for all k levels of a particular branching channel geometry is plotted as a function of m in Fig. 3.3.3 for optimized flow networks

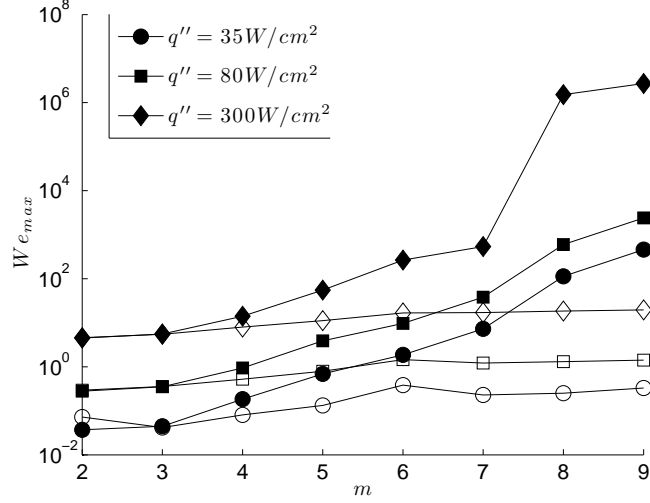


Figure 3.10: Weber number generated from the optimized flow networks versus m , for $R = 20\text{mm}$, $H = 150\mu\text{m}$, $w_m = 50\mu\text{m}$, filled markers are flow networks subject to the stability constraint, open markers are networks not subject to the stability constraint

with and also without being subject to the stability constraint of Eqn. 3.23. Both flow networks are shown to illustrate any possible improvement in the stability of the flow, lower Weber numbers mean that the surface tension force pushing the bubble upstream is greater than the inertia force pushing the bubble downstream, thus lower Weber numbers constitute more unstable flow conditions. It should be noted that the minimum Weber number out of each k level represents the worst-case scenario, however the minimum Weber number does not change with the use of the stability constraint, therefore the maximum Weber number is plotted in Fig. 3.3.3, representing the best case scenario. Figure 3.3.3 illustrates that by employing the stability constraint, for $m > 3$, the ratio of inertia to surface tension

forces increases, thereby stabilizing the flow. However, it should be noted that the degree of stability should be verified experimentally, all that can be determined from the present analysis is that stability is improved.

Additional evidence that stability is improved with the constraint of Eqn. 3.23 is evident by looking at the maximum ratio of evaporative to inertia force, \bar{F} for all k levels of the optimized branching channel networks. Figure 3.3.3 illustrates the maximum value of \bar{F} over all k levels of the optimized branching channel geometry with and without being subject to the stability constraint. Unlike the maximum Weber number, the maximum value of \bar{F} over all k levels represents the worst-case scenario, i.e., the case with the greatest evaporation force forcing the bubble upstream. The open markers in Fig. 3.3.3 indicate flow networks not subject to the stability constraint, filled markers indicate flow networks subject to the stability constraint. With the exception of the $m = 8$ and $m = 9$, with the inclusion of the stability constraint, the degree of stability is improved based on the reduction of \bar{F} in Fig. 3.3.3. It should be mentioned that the $m = 8$ and $m = 9$ flow networks for $q'' = 300W/cm^2$ result in single-phase flow, which is why the values for \bar{F} are so low, for this reason it is not a fair to make a judgement regarding the stability of those networks. In summary, the use of the stability constraint (expressed in Eqn. 3.23) does increase the optimum flow power by restricting the range of optimization, seen in Fig. 3.8, however improvement in the degree of flow stability is evident by examining the change in the non-dimensional parameters, We and \bar{F} in Figs. 3.3.3 and 3.3.3.

The optimal β and γ values of the solutions found in Fig. 3.8 are plotted in

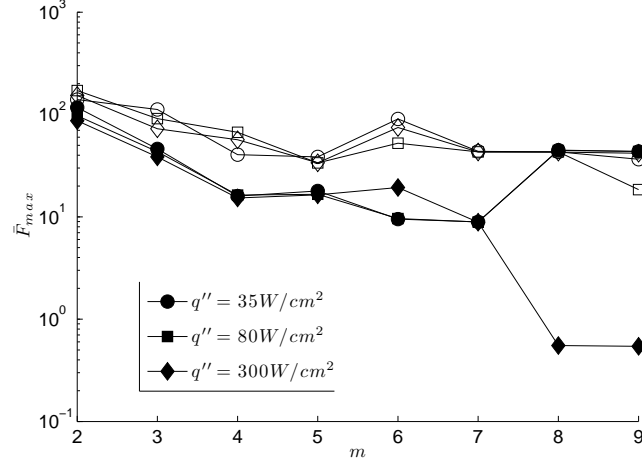


Figure 3.11: Evaporative to inertia force ratio, \bar{F}_{max} , generated from the optimized flow networks versus m , for $R = 20mm$, $H = 150\mu m$, $w_m = 50\mu m$, filled markers are flow networks subject to the stability constraint, open markers are networks not subject to the stability constraint

Fig. 3.12. Values of γ are plotted on the right most axes, while values of β are plotted on the leftmost axis of Fig. 3.12. The most obvious aspect of Fig. 3.12 is that the optimal values of β are consistently less than one, a finding corroborated by the flow power contours of Fig. 3.9. Furthermore, the dependence of the optimal values of β and γ are very weak with respect to heat flux. However, the dependence of the optimal geometric parameters of Fig. 3.12 with m is clearly evidence and is in part due to the changing active constraints.

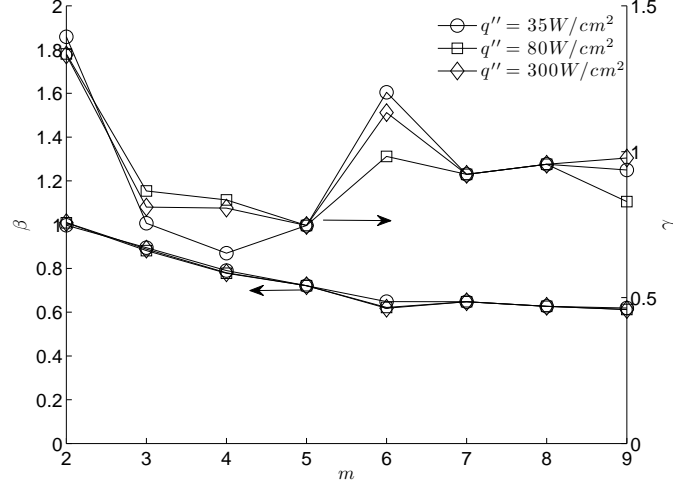


Figure 3.12: Optimal β , and γ as a function of m for $H = 150\mu m$, $R = 20mm$ not subject to the stability constraint

3.4 Conclusions

A one-dimensional model developed and validated by Daniels [20] was used to estimate the flow power of a two-phase network of fractal-like branching microchannels. The flow power formed an objective function to be minimized, subject to several geometric constraints to account for fabrication and practical limitations. Two methods were considered to constrain the mass flow rate, a maximum wall temperature limitation and a critical heat flux constraint. In addition, the effects of a stability constraint were examined.

First, it was found that the wall temperature constraint yielded two-phase flow networks with inferior performance compared to single-phase networks of Heymann et al. [21] due to the maximum wall temperature being immediately upstream of

the point of incipient boiling. Second, if the maximum wall temperature is not of great concern to the design, a two-phase fractal-like flow network constrained to 50% of critical heat flux outperformed the single-phase flow networks. Third, the performance of the two-phase flow networks was attenuated with the inclusion of the stability constraint.

In addition to optimizing over a greater range of heat fluxes and m values, this study may be improved by considering the cost and benefit of constraining the device to operate at different percentages of the critical heat flux.

Chapter 4 – Modeling a Single-Phase Fractal-like Branching Microchannel Heat Exchanger

4.1 Introduction

Disk-shaped heat sinks with fractal-like branching microchannels were first proposed by Pence [4, 5] as a means to reduce both the pressure drop and the maximum streamwise wall temperature difference observed in parallel microchannel heat sinks. A fractal-like branching flow network, inspired by nature and characterized by fixed length and width scale ratios between consecutive branch levels was studied. Representative fractal-like flow networks in a disk-shaped heat sink configuration are shown in Fig. 1.1. Flow enters the network normal to the page at the inlet plenum located at the center of the disk and exits parallel to the page at the periphery. Each channel emanating from the inlet plenum bifurcates into two narrower channels, each of which in turn bifurcates. This repetitive pattern is considered fractal-like because the ratio of the channel widths and channel lengths between the consecutive branch levels are fixed. The heat transfer coefficient is enhanced downstream of each bifurcation as the thermal boundary layer redevelops at the newly formed walls and the hydraulic diameter decreases. A three-dimensional computational fluid dynamic (3-D CFD) model employed by Alharbi et al. [2] confirmed the boundary layer redevelopment and validated the fractal-like branching

channel model of Pence [5]. The fractal-like heat exchanger makes use of two disks of identical branching channels configuration, for example in Fig. 1.1(a), with a thin interface separating the two fluid streams. For the present analysis, heat is transferred between a hot stream of Paratherm NF to a cold, single-phase stream of water.

The goal of the present study is to develop a reasonably accurate and efficient one-dimensional (1-D) model for estimating the heat transfer and hydrodynamic characteristics of a fractal-like branching microchannel heat exchanger. Reasonable accuracy is needed for comparing one design with another and an efficient algorithm is necessary to analyze a significant number of configurations within a reasonable time. A balance between accuracy and efficiency is beneficial for iterative optimization techniques such as gradient based searches or genetic algorithms.

In the present study, the performance of the heat exchanger is characterized by the ratio of heat transfer, q , over the flow power, $Q\Delta P$. Others have sought an optimized design by minimizing the entropy generation rate [32–34] as well as minimizing the size [35].

Heat exchangers using flow networks characterized as constructal have been studied using 1-D analyses in order to minimize the total flow and thermal resistances. Bejan [36] sought to minimize the bulk fluid temperature drop and thermal resistance by employing channels with short flow lengths to restrict flow to the developing region. da Silva et al. [37] sought to minimize the thermal and hydrodynamic resistances of counter-flow heat exchangers. In the same fashion as da Silva et al. [37], Zimparov et al. [38] minimized the thermal and hydrodynamic

resistances in a co-flow heat exchanger for three different network designs.

While there is little debate regarding the computational efficiency of 1-D models, Chong et al. [39] addressed their accuracy by comparing results from a simplified 1-D resistance model of heat transfer through a double layer heat sink with results from 3-D CFD simulations. The flow network consisted of an array of parallel channels. The 1-D model incorporated developing flow correlations from Shah and London [7] and a fin efficiency from Incropera and DeWitt [40] to account for thermal resistance in the walls of the heat sink. Results from the 1-D model with fin efficiency included compared well with the 3-D CFD simulations.

A detailed account of conduction in heat exchangers is provided by Shah and Sekulić [41]. It was concluded that conduction in the fins (sidewalls) and stream-wise conduction in the interface material between the two streams may significantly affect the predictive capabilities of a 1-D model.

4.2 Model

4.2.1 Fractal Geometry

For the present study, the geometry of the fractal-like branching channels is defined solely by the disk radius, R , channel height, H , terminal channel width, w_m , length ratio, γ , and width ratio, β , the number of branching levels, m , and the number of 0^{th} level channels, n_0 . The value of n_0 can be thought of as the number of fractal-like trees in a disk. All other geometric parameters are derived from these

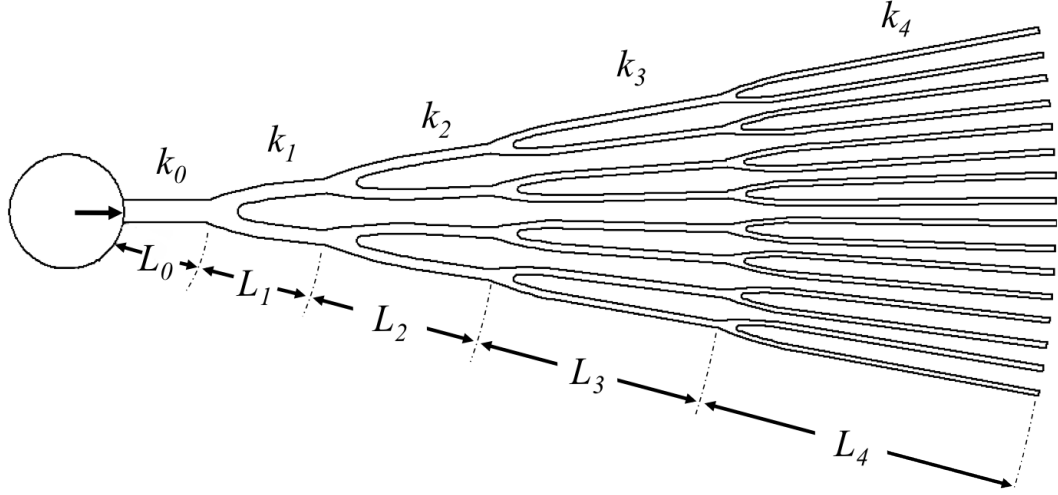


Figure 4.1: Illustration of a fractal-like branching channel tree

variables. Evident from Figs. 1.1(a) and 1.1(b) is that $n_0 = 16$. Each branch level, identified on the half fractal-like tree shown in Fig. 4.1, is indexed by k , with the 0^{th} level and m^{th} level connecting the tree to the inlet plenum and exit plenum, respectively. The width ratio is defined by

$$\beta = \frac{w_{k+1}}{w_k} \quad (4.1)$$

with the length ratio defined by

$$\gamma = \frac{L_{k+1}}{L_k} \quad (4.2)$$

The total channel length, L_{tot} , is the sum of the consecutive branch lengths and can be determined by subtracting the inlet plenum radius from the disk radius

$$L_{tot} = R - \frac{w_m n_0}{\beta^m 2\pi} \quad (4.3)$$

The length of the 0^{th} level channel can be ascertained from γ and the total channel length, L_{tot} , using

$$L_0 = \frac{L_{tot}}{\sum_{k=0}^m \gamma^k} \quad (4.4)$$

from which the consecutive channel lengths can be determined using Eqn. 4.2.

4.2.2 Pressure Drop

Because the momentum equation is not coupled between the hot and cold sides of the heat exchanger, the pressure drop through each side of the branching channel heat exchanger is determined in the same fashion as described by Pence [5]. In the 1-D model the entire path length, from the inlet plenum to the exit plenum is divided into discrete lengths, Δx . The outlet pressure is specified and the pressure distribution is computed using the 1-D pressure equation

$$p_i = p_{i+1} + \frac{1}{2} \rho u_m^2 C_{f,app} \frac{4\Delta x}{D_h} \quad (4.5)$$

given by White [6], where the $C_{f,app}$ factor accounts for the increased pressure drop in the hydrodynamically developing region. The solution to the pressure

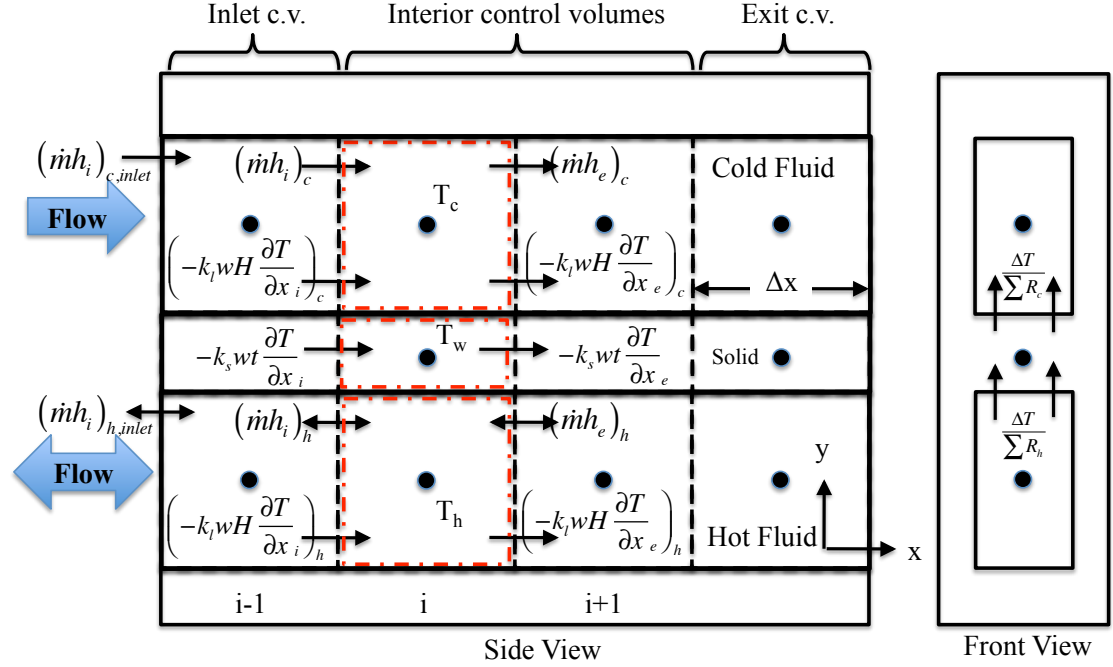


Figure 4.2: Representative control volumes for the energy balance

distribution is computed with little computational effort because Eqn. 4.5 is simply a backward substitution.

4.2.3 Energy Balance

In Pence [5], the walls of the fractal-like network were subjected to a uniform heat flux. With temperature specified at the inlet, the energy equation was easily computed with forward substitution. However, the heat exchanger model, a conjugate problem because the hot and cold stream energy balances are coupled with the material interface, requires a different approach to solving the energy balance. El-

emental control volumes for the hot and cold fluids and for the solid interface are illustrated in Fig. 4.2. Conduction in two directions (x and y) is included in the energy balance of the solid control volumes. Fluxes at the faces of the fluid control volumes include advection and conduction. To maintain the one-dimensionality of the model, the y-component of the energy transport is quantified by a sum of thermal resistances. For example the sum of the resistances, $\sum R$ accompanying a $\Delta T = (T_w - T_c)$ is defined by

$$\sum R_c = \frac{t}{2k_{mat}w_k\Delta x} + \frac{1}{2h_c\eta_{fin,c}(w_k + H)\Delta x} \quad (4.6)$$

The first term in Eqn. 4.6 represents the resistance due to conduction through 1/2 of the material interface. The second term in Eqn. 4.6 is convective resistance, the total of which is illustrated in the “Front View” of Fig. 4.2 where the sum of the resistances in between T_c and T_s is $\sum R_c$ and equivalently, the sum of the thermal resistances in between T_h and T_s is $\sum R_h$, eliminating the need to solve a *discretized* energy balance in the y-direction. The thermal resistance through the side walls of the interface material is addressed with the fin efficiency in a subsequent sub-section.

The longitudinal thermal resistance through the interface material is a result of the material type, cross-sectional area, A_c and total conduction length, L . To determine if the longitudinal conduction is significant compared to the change in

enthalpy of the fluid, Shah and Sekulić [41] give the non dimensional parameter,

$$\lambda = \frac{k_{mat} A_c}{L (\dot{m} C_p)_{min}} \quad (4.7)$$

which gives the ratio of the interface conduction to stream convection for a given temperature drop in the counterflow-configuration. The conductive interface material of the present study is disk-shaped, thus λ must be modified to λ' , given as

$$\lambda' = \frac{2\pi k_{mat} t}{\ln \left(\frac{2\pi R \beta^m}{n_0 w_m} \right) (\dot{m} C_p)_{min}} \quad (4.8)$$

For λ' , values of $\mathcal{O}(1)$ or greater, the longitudinal interface conduction is not negligible. For a fractal-like disk with a water flow rate of 10 g/min, a disk radius of 20mm, thickness of 500 μm , λ' , is approximately 0.2 and 6 for stainless steel and copper interfaces, respectively. Since the aforementioned conditions and geometries are typical for this type of device, the longitudinal conduction should not be neglected in the counter-flow configuration.

By summing the fluxes around the control volumes in each region and collecting terms, expressions for the control volume temperature in terms of the neighboring temperatures are obtained. The cold stream energy balance becomes

$$\begin{aligned} 0 &= T_{c,i-1} \left(\dot{m}_{c,k} C_{p,c} + \frac{k_{f,c} w_k H}{\Delta x} \right) \\ &+ T_{c,i} \left(-\dot{m}_{c,k} C_{p,c} - \frac{2k_{f,c} w_k H}{\Delta x} - \frac{1}{\sum R_{c,i}} \right) \\ &+ T_{c,i+1} \left(\frac{k_{f,c} w_k H}{\Delta x} \right) + T_{s,i} \left(\frac{1}{\sum R_{c,i}} \right) \end{aligned} \quad (4.9)$$

With the solid interface energy balance being

$$\begin{aligned}
0 = & T_{s,i-1} \left(\frac{k_s A_{xs,i-1/2}}{\Delta x} \right) \\
& + T_{s,i} \left(-k_s (A_{xs,i-1/2} + A_{xs,i+1/2}) - \frac{1}{\sum R_{c,i}} - \frac{1}{\sum R_{h,i}} \right) \\
& + T_{s,i+1} \left(\frac{k_s A_{xs,i+1/2}}{\Delta x} \right) + T_{c,i} \left(\frac{1}{\sum R_{c,i}} \right) + T_{h,i} \left(\frac{1}{\sum R_{h,i}} \right)
\end{aligned} \tag{4.10}$$

The area of the solid interface changes as a function of the radial distance from the inlet of the 0^{th} level channel, x and determined from

$$A_{s,i} = \left(\frac{n_0 w_m}{2\pi \beta^m} + x_i \right) 2\pi H \tag{4.11}$$

The hot stream energy balance is expressed in a form similar to that of the cold stream

$$\begin{aligned}
0 = & T_{h,i-1} \left(\dot{m}_{h,k} C_{p,h} + \frac{k_{f,h} w_k H}{\Delta x} \right) \\
& + T_{h,i} \left(-\dot{m}_{h,k} C_{p,h} - \frac{2k_{f,h} w_k H}{\Delta x} - \frac{1}{R_{h,i}} \right) \\
& + T_{h,i+1} \left(\frac{k_{f,h} w_k H}{\Delta x} \right) + T_{s,i} \left(\frac{1}{R_{h,i}} \right)
\end{aligned} \tag{4.12}$$

In order to maintain numerical stability when solving the system of equations, the advection fluxes are approximated using the upwind differencing scheme. Advection heat transfer leaving the control volume is approximated at $T_{h,i}$ and the advection entering the control volume is approximated at $T_{h,i-1}$. The temperatures

utilized in Eqn. 4.12 for the advection terms need only to be changed when the flow changes direction, i.e. in a counter flow heat exchanger. The $T_{h,i-1}$ temperature would no longer have a $\dot{m}C_p$ term, instead the flow is moving from right to left in the hot stream of Fig. 4.2, therefore the upstream temperature is now at the $i + 1$ index. Concerning boundary conditions, the conduction fluxes at the boundaries were estimated by assuming a piecewise linear temperature distribution between the temperature nodes. The boundary conditions of the 1-D model were chosen for convenience and to mimic the conditions a designer would face; a known uniform inlet temperature distributions as well as zero conduction at the inlet and outlet of the device.

All thermo-physical properties are set constant. With the inclusion of the longitudinal conduction term and stream coupling terms, the coefficient matrix for temperature is no longer strictly lower triangular; it becomes a banded matrix and cannot be solved for with forward substitution, instead the native banded matrix solver in MATLAB is used.

4.2.4 Nusselt Number Determination

One of the important aspects of the branching channel geometry is the re-development of the thermal and hydrodynamic boundary layers following each bifurcation. For a uniform heat flux boundary condition, the heat transfer coefficient can be determined from the non-dimensional Nusselt number, Nu from Shah and London's [7] correlation for simultaneously developing thermal and hydrodynamic boundary

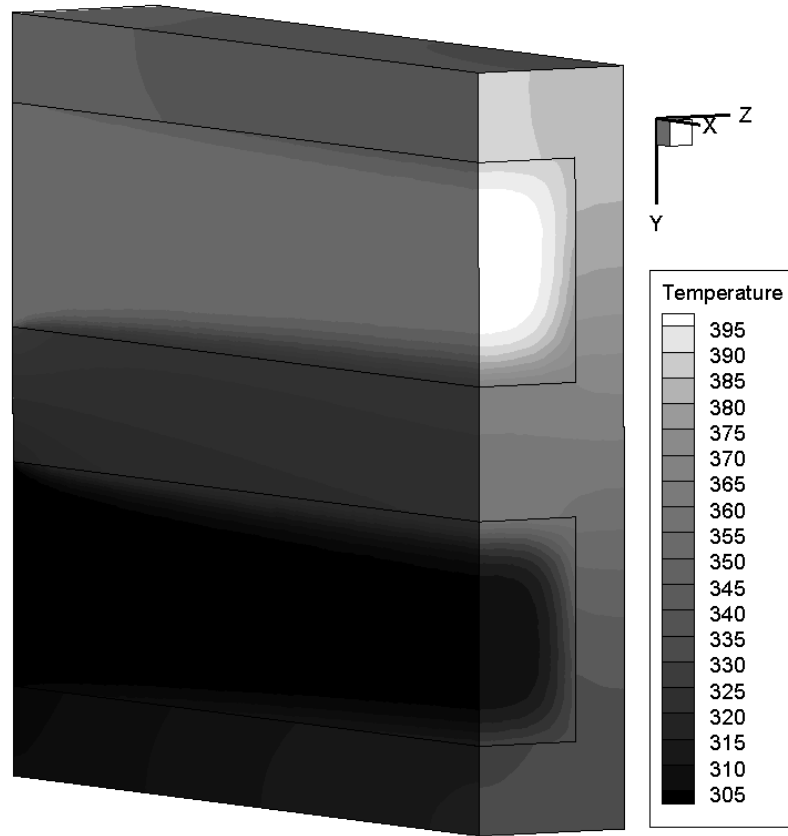


Figure 4.3: Example of temperature distribution obtained from the 3-D CFD model

layers. As mentioned previously, under only special circumstances is a uniform heat flux or uniform boundary condition present at the walls of the heat exchanger. Therefore, the convenient Nusselt number correlation of Shah and London [7] should not be employed without first verifying the validity under the boundary conditions of the heat exchanger.

To determine the Nusselt number as a function of the inverse Graetz number, x^* , a the fluid flow and heat transfer of a simple two-stream heat exchanger was

modeled in three dimensions using the commercial CFD package, StarCCM+. An illustration of the mesh, geometry and temperature distributions from the solver is illustrated in Fig. 4.3. Two different interface materials, copper and stainless steel were used to characterize the importance of conduction through the fins (i.e., walls) of the microchannel heat exchanger. Four flow rates between 0.24-2.37 m/s were chosen to ensure that less than 50% of the channel occupied the developing flow regime, defined by

$$x^* = Gz^{-1} = \frac{x}{D_h Re Pr} < 0.001 \quad (4.13)$$

The remaining geometrical parameters are; the height, H was 250 μm , and interface thickness was 150 μm . The fin thickness was chosen to be 1/2 of the channel width, $W = 250 \mu m$. Other widths were used to ascertain the effects of changing aspect ratio, however the change in the solution was negligible compared to the spread of the data. A 50 μm grid size was used and found to yield a grid-refined solution. A symmetry boundary condition was applied at the mid-plane of the channel to reduce the number of control volumes by one-half. Both a uniform temperature and velocity inlet boundary condition was enforced at the channel entrances, with a zero-diffusion, constant pressure outlet condition was applied to both the hot and cold streams. The time to solve a single case with approximately 10,000 control volumes was approximately 15 minutes running StarCCM+ on a desktop computer, utilizing all four cores of an Intel vPro processor with 4GB of RAM. The heat transfer coefficient can be determined from the temperature and

velocity data obtained from the CFD analysis by

$$h(x) = \frac{q''(x)}{T_w(x) - T_b(x)} \quad (4.14)$$

Where the heat flux at the boundary, q''_x and the wall temperature, $T_{w,x}$ are readily known quantities, averaged over the circumference of the channel wall. The bulk fluid temperature, essentially a mass-flow weighted average temperature, $T_{b,x}$ is estimated from

$$\begin{aligned} T_b(x) &= \frac{\int_{y,z} T(x, y, z) \dot{m}(x, y, z)}{\int_{y,z} \dot{m}(x, y, z)} \\ &\approx \frac{1}{\dot{m}_{tot}} \sum_{y,z} T(x, y, z) \dot{m}(x, y, z) \end{aligned} \quad (4.15)$$

The Nusselt number is computed using the hydraulic diameter, D_h of the channel, and the thermal conductivity, k_f of the fluid for both hot and cold streams of the heat exchanger. The results are presented for both stainless steel and copper interface materials in Fig. 4.4.

The most obvious feature of Fig. 4.4 is the difference in Nusselt numbers between the stainless steel and copper interface materials. The deviation comes from the manner in which the heat transfer coefficient is determined in Eqn. 4.14, assuming that the wall temperature and heat flux are uniform around the periphery of the channel. To mitigate this, an expression accounting for the non-uniformity of the temperature distribution through the walls, i.e., fins of the heat exchanger may be employed. This is known as the fin efficiency, η .

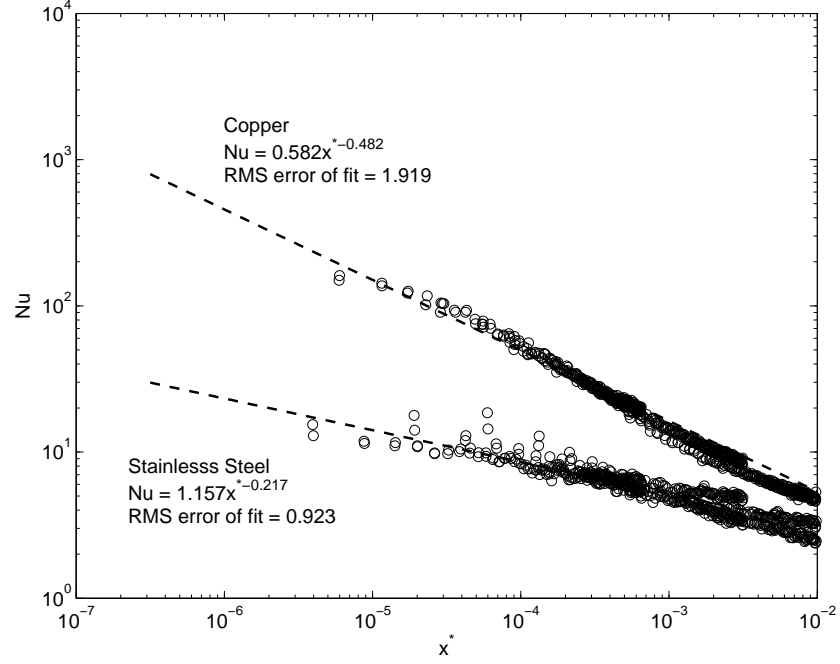


Figure 4.4: Nusselt number as a function of x^* for copper and stainless steel interface materials determined from 3-D CFD model, circles (\circ) are data, the dashed line (- -) is the least squares fit

4.2.5 Fin Efficiency

The fin efficiency can be found in [40], and since the outer boundary of the wall is insulated the adiabatic tip boundary condition is used. In order to compare the temperature distributions between the CFD data and the analytic solution, the relation

$$\frac{\theta}{\theta_{y=0}} = \frac{\cosh\left(\sqrt{\frac{h}{k_{mat}t}}(H-y)\right)}{\cosh\left(\sqrt{\frac{h}{k_{mat}t}}H\right)} \quad (4.16)$$

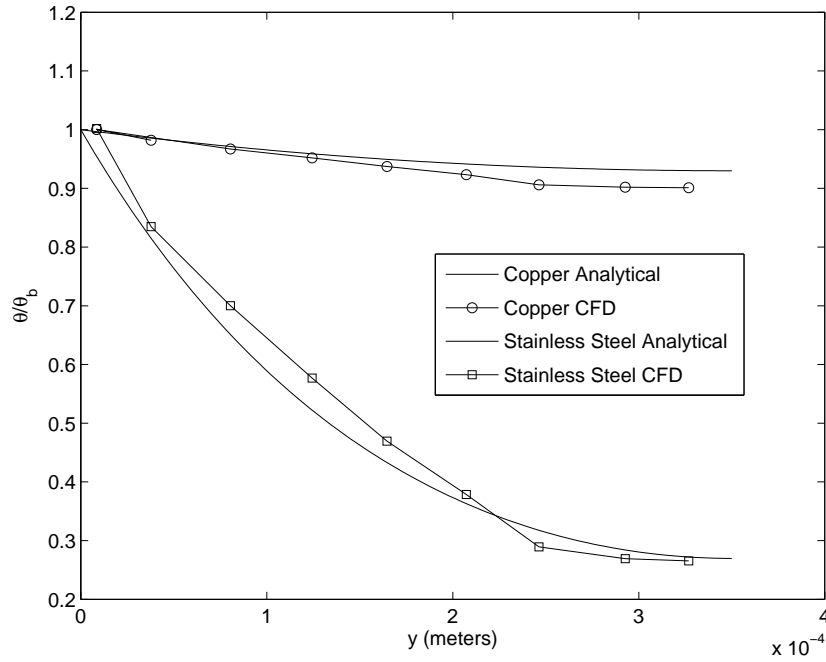


Figure 4.5: Comparison of analytical [40] and 3-D CFD model of fin temperature distributions through both copper and stainless steel

is used. Where, θ is the temperature difference, $T_{w,x,y} - T_{b,x}$ between the wall and the bulk fluid. The analytical function and CFD data for the temperature distribution ratio, $\theta/\theta_{y=0}$ in the fin at $x = 500\mu m$ for copper and stainless material is illustrated in Fig. 4.5. Both distributions of the CFD data agree well enough with the analytic solution to warrant employing the fin efficiency, described by the temperature distribution in the analysis.

The previous estimation of the heat transfer coefficient was ineffective, as it did not provide a result independent of material type. By including the fin efficiency in the analysis of the CFD data, the effects of the non-uniform temperature dis-

tribution due to conduction in the walls should be mitigated to provide a Nusselt number independent of material. The fin efficiency for the adiabatic tip boundary condition is

$$\eta = \frac{\tanh \sqrt{\frac{h}{k_{mat}t}} H}{\sqrt{\frac{h}{k_{mat}t}} H} = \frac{q''}{h\theta_b} \quad (4.17)$$

The true heat transfer coefficient is buried on the right hand side and in the hyperbolic tangents of Eqn. 4.17, it is therefore necessary to solve the non-linear equation in a numeric solver. This was done and the results are presented in Fig. 4.6. Notice that for $\eta = 1$ Eqn. 4.17 reduces to Eqn. 4.14, thus it should not be surprising that for copper with a high thermal conductivity there is less change between the Nusselt number distributions of Figs. 4.4 & 4.6. There is now little difference between the Nusselt numbers of the stainless steel and copper CFD datas, thus the independence is achieved through accounting for the fin efficiency when estimating the heat transfer coefficient. By including the fin efficiency in the model, results change by approximately 10% in a stainless steel fractal-like heat exchanger. The Nusselt number correlations in Figs. 4.4 and 4.6 were determined by the non-linear least squares fit of

$$Nu(x^*) = C_1 x^{*C_2} \quad (4.18)$$

to the data. The sensitivity of the coefficients in Fig. 4.6 to the number of significant digits was found to be small for the C_1 coefficient, but larger for the C_2 coefficient. For example, rounding C_1 to 0.50 and C_2 to 0.56 resulted in ≈ 1 %

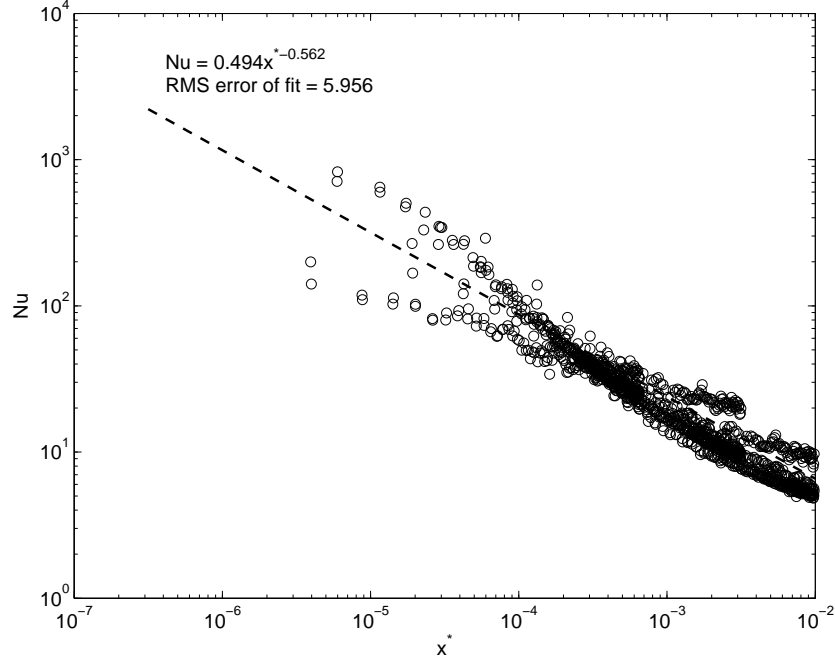


Figure 4.6: Nusselt number as a function of x^* for copper and stainless steel interface materials determined from 3-D CFD model, adjusted for fin efficiency, circles (\circ) are data, the dashed lines (- -) is the least squares fit

change in the correlation of Eqn. 4.18. The improvement in agreement between the copper and stainless steel CFD data after accounting for fin efficiency is somewhat devalued by the increase root-mean-squared (RMS) error of the fit between Figs. 4.4 and 4.6. However, the C_2 coefficient in Fig. 4.6 of -0.56 agrees with the $h \propto u_m^{1/2}$, with $x^* \propto 1/u_m$ dependence given by Shah and Sukulić [41]. Fig. 4.7 compares the correlation of the present study with the correlation given by Shah and London [7], which takes the form

$$Nu = \frac{1}{a_0 [1 - \exp(-a_1 x^{*a_2})]} \quad (4.19)$$

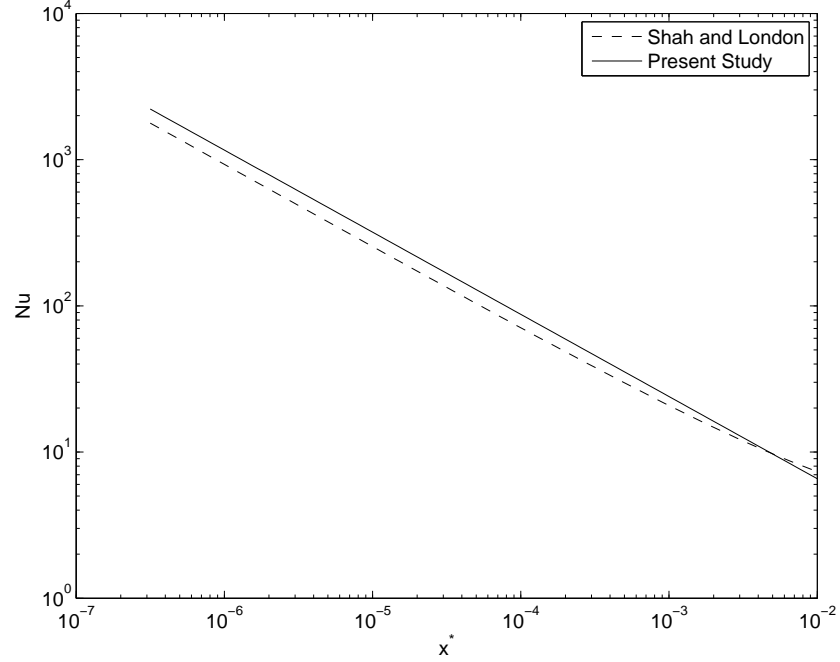


Figure 4.7: Nusselt number correlations from Shah and London [7] and present study from Eqn. 4.18 with $C_1 = 0.50$ and $C_2 = 0.56$ as a function of x^*

for simultaneous thermal and hydrodynamic boundary layer redevelopment under a uniform heat flux boundary condition. The coefficients, a_{0-2} are tabulated as a function of channel aspect ratio. The characteristic slopes of the two correlations are nearly identical, what separates them is the constant term for which x^* is multiplied by. However, it is clear that when comparing the difference in between the two correlations that there is little difference between the correlations compared to the magnitude of the error, especially at low values of x^* . For example the RMS error between the correlation of Shah and London [7] and the CFD data is approximately 3.7%. Moreover, the skill of the fit (commonly referred to as

the coefficient of determination, or R^2 value) in Fig. 4.6 is approximately 0.64, compared to the skill of the Shah and London [7] correlation to the CFD data of 0.63 illustrates that the fit of the data is not any more statistically significant than the [7] correlation. Although there was no significant improvement to the Nusselt number correlation studied in a heat exchanger, this does not mean that the CFD analysis was performed in vain, it illuminated the need for a fin efficiency. The Nusselt number correlation provided by Shah and London [7] will be used in the 1-D model because the dependency of aspect ratio is well defined.

4.3 Results

The bulk of the present study is the development and validation of the fractal-like branching channel heat exchanger model. Some preliminary comparisons of different fractal and parallel channel geometries are presented in this section to illuminate the need for further study and optimization of the fractal geometries.

4.3.1 Algorithm validation

A comparison of the ϵ -NTU model results to the analytical solutions of [40] was performed to determine if the algorithm is consistently solving the energy equation. The co and counter flow ϵ -NTU relationships given by

$$\epsilon = \frac{q}{(\dot{m}C_p)_{min}(T_{h,i} - T_{c,i})} = \frac{1 - \exp[-NTU(1 + C^*)]}{1 + C^*} \quad (4.20)$$

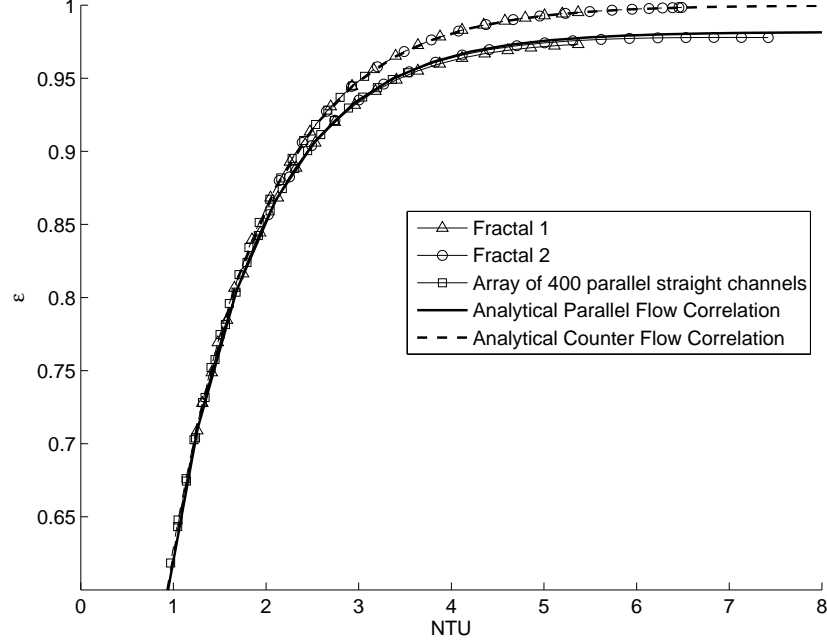


Figure 4.8: ε vs. NTU for $C^* = 0.019$ and $\dot{m}_c = 10 - 1000$ g/min obtained from the 1-D model, solid line (—) corresponds to co-flow, dashed line (---) corresponds to counter flow

and

$$\epsilon = \frac{1 - \exp[-NTU(1 - C^*)]}{1 - C^* \exp[-NTU(1 - C^*)]} \quad (4.21)$$

respectively for a concentric tube are appropriate for the present model, despite the complex fractal geometries, because at each discrete control volume, there is one pass of two fluids flowing in either co or counterflow configurations. Fig. 4.8 illustrates that the algorithm solves the energy equation in a manner consistent with the ϵ -NTU method and compares two fractal-like heat exchangers with an array of straight parallel channels, which will be discussed in a later subsection.

To test the dependence of the solution to the grid size, three grid sizes, $\Delta x = 1 \text{ mm}$, $100 \text{ }\mu\text{m}$ and $10 \text{ }\mu\text{m}$. A grid sizes of $1 \text{ }\mu\text{m}$ resulted in a domain with 18,000 elemental volumes, which exceeded the memory limit of the desktop computer. The axial temperature profiles are illustrated in Fig. 4.9. The solution time in the legend of Fig. 4.9 is the time it takes to construct and solve the coefficient matrix for both pressure and temperature. Between the grid sizes of 1 mm and $100 \text{ }\mu\text{m}$, the solution time increases by a factor of 2, but the temperature between the two solutions changes by approximately $10 \text{ }^\circ\text{C}$. As the grid becomes even more refined, between the grid sizes of $100 \text{ }\mu\text{m}$ and $10 \text{ }\mu\text{m}$, the solution time increases by a factor of 325, but the temperature between the two solutions changes by approximately $1 \text{ }^\circ\text{C}$. The $100 \text{ }\mu\text{m}$ grid size offers a compromise between grid independence and computational time in order to keep the algorithm as efficient as possible.

4.3.2 Experimental validation

The results of the 1-D model were compared with experimental data obtained by Stull and Sabo [42] in a stainless steel fractal-like branching channel heat exchanger. The fractal geometry of the heat exchanger is similar to that of Fig. 1.1(b), with $\gamma = 1.41$, $\beta = 0.707$, $w_m = 93 \text{ }\mu\text{m}$, $H = 250 \text{ }\mu\text{m}$ and $t = 717 \text{ }\mu\text{m}$. The cold side fluid is distilled, deionized water, and the hot side fluid is Paratherm NF heat transfer fluid. The heat exchanger was operated in the co-flow configuration with a cold side mass flow rate of 60 g/min while varying the hot side flow rate from $200\text{-}400 \text{ g/min}$. Fig. 4.10 provides an exploded view of the test device, man-

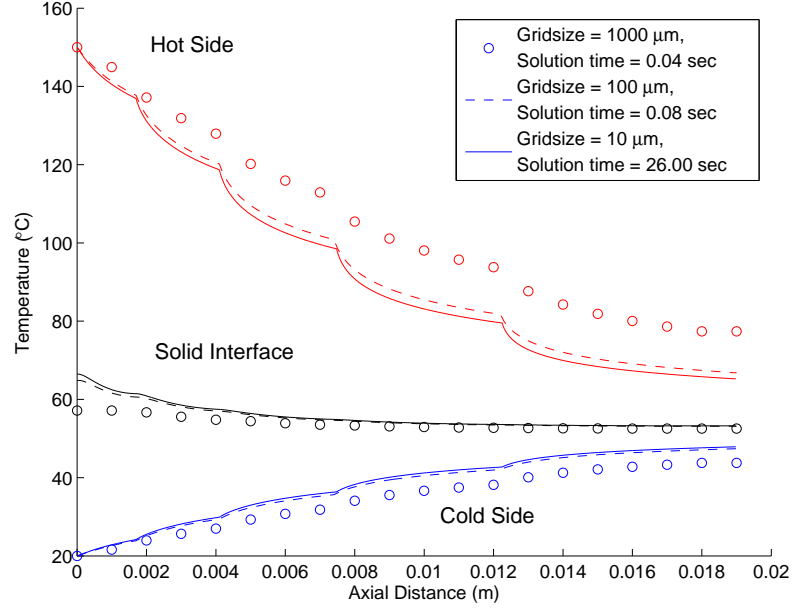


Figure 4.9: Bulk fluid and interface temperature profiles as a function of axial distance and grid size obtained from the 1-D model, the three sets of datas correspond to the hot bulk fluid temperature, interface temperature and cold bulk fluid temperature

ifold and measurement locations The inlet temperature of the hot side was set at 115 °C, and datas for three cold side inlet temperatures of 25, 40, and 60 °C were gathered.

Fig. 4.11 illustrates the comparison of the experimental ΔT_{lm} with the ΔT_{lm} computed from results obtained from the 1-D model under the same operating conditions. There is a negligible dependency of ΔT_{lm} to the hot side mass flow rate observed by Stull and Sabo [42], therefore the datas for each inlet temperature are clustered together, illustrated in Fig. 4.11. The highest cluster of ΔT_{lm}

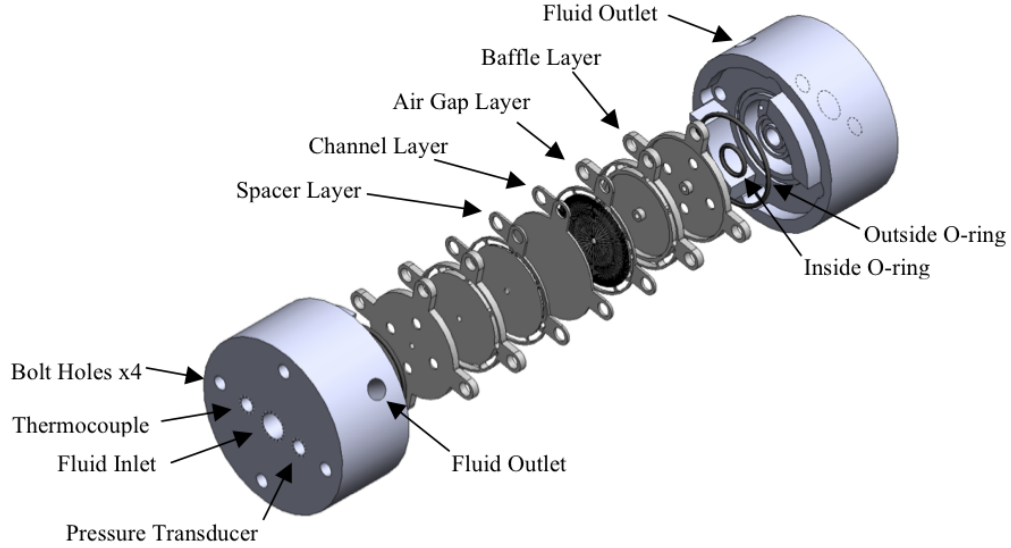


Figure 4.10: Illustration of the testing apparatus used to experimentally test the fractal-like branching channel heat exchanger [42], geometry is outlined in Table 4.1

corresponds to the lowest cold side inlet temperature and monotonically decreases as the inlet temperature increases, simply due to the fact that a higher cold side inlet temperature means a lower potential temperature difference. This relationship aside, the model predictions of ΔT_{lm} agree with the experimental data to an average deviation of 12%. The experimental uncertainty, determined from the Kline and McClintock method [43] is approximately 5°C , illustrated in Fig. 4.11 as an error bar. Moreover, the average error of the data around this deviation is approximately 1.1°C , pointing to the possibility that there may be some physical significance to the disagreement, since the numerical predictions of ΔT_{lm} are consistently higher than the experimental data. Further experimental study is warranted to ascertain the reason for the discrepancy.

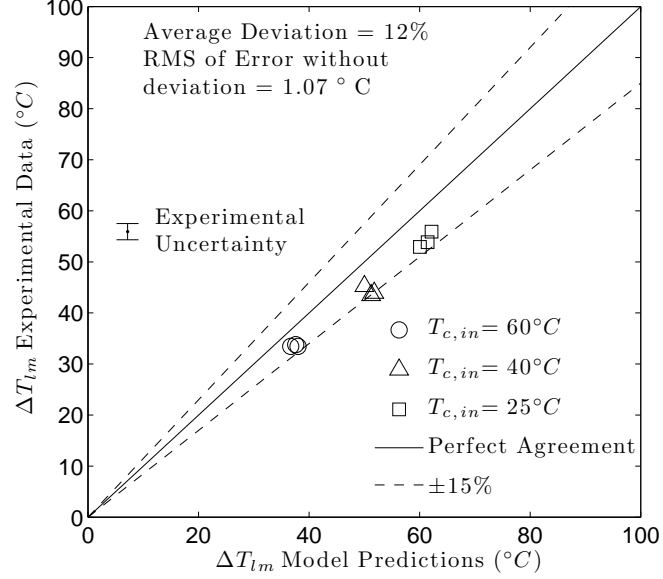


Figure 4.11: Experimental versus 1-D Numerical Model ΔT_{lm} for $\dot{m}_c = 60$ g/min, $\dot{m}_h = 200, 300$ and 400 g/min and three cold side inlet temperatures from [42]

4.3.3 Heat exchanger performance

Three geometries, two fractal-like ($m = 4$) and one array of straight parallel channels ($m = 0$) were considered for the following analyses, their geometric parameters are outlined in Table 4.1. The the three geometries were given the same global dimensions and subject to the same total mass flow rate and inlet temperatures in order to make a fair comparison of their performance. The overall size of the heat exchanger was made so that the planform extrema occupied the same space. For example, since the fractal-like heat exchangers are disk shaped, the total planform area taken up by the disk is a square of dimensions $2R \times 2R$, thus the array of

Table 4.1: Geometric parameters of the three heat exchanger geometries with $H = 250 \mu m$, $t = 717 \mu m$ and $R = 20 mm$

	n_0	m	$w_m(\mu m)$	β	γ
Fractal 1	16	4	100	0.707	1.41
Fractal 2	79	4	50	0.750	1.18
Straight Channels	400	0	50	-	-

straight parallel microchannels may also take up the dimension $2R \times 2R$, which gives an advantage to the array of straight channels in that the channels occupy the the entire area defined by the extrema.

The first fractal geometry in Table 4.1 (Fractal 1) is an equivalent of the test device fabricated and tested under adiabatic and diabatic boiling conditions in several studies [23, 44, 45]. It is included for comparison to an existing device and was also used by Stull and Sabo [42]. The second fractal geometry (fractal 2) was chosen as a result of the optimized single-phase microscale fractal-like branching channel heat sink geometry determined by Heymann et al. [21]. The third and final geometry is an array of straight, parallel microchannels given the same terminal channel width of $50 \mu m$ which is what allows 400 channels to be packed into the area $2R \times 2R = 40mm \times 40mm$. All geometries have a channel height, $H = 250 \mu m$ and interface thickness, $t = 717 \mu m$.

Inlet temperatures of $20^\circ C$ and $150^\circ C$ were set for cold and hot sides, respectively. A channel height, H of $250 \mu m$ and interface thickness, t of $717 \mu m$, finally the mass flow rates of the hot and cold sides were varied from 10 to 130 g/min. The heat transfer, q versus log-mean temperature difference (ΔT_{lm}) for all three

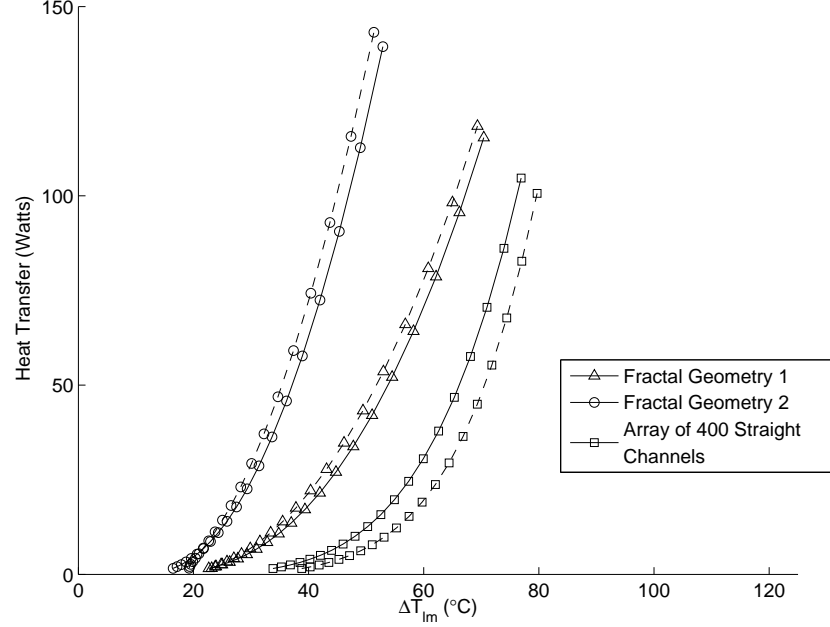


Figure 4.12: Heat transfer vs. log-mean temperature difference for $C^* = 0.019$, and $\dot{m}_c = 10 - 130$ g/min obtained from the 1-D model, solid line (—) corresponds to co-flow, dashed line (---) corresponds to counter flow

devices is illustrated in Fig. 4.12, while the performance parameter, ϵ_p , defined as

$$\epsilon_p = \frac{q}{Q_h \Delta P_h + Q_c \Delta P_c} \quad (4.22)$$

versus hot side and cold side mass flow rates is presented in Fig. 4.13.

The improvement in heat transfer performance between the fractal geometries and straight channels is not unexpected since the fractal geometry takes advantage of the the boundary layer redevelopment following each bifurcation, thus the heat transfer coefficient is higher and UA increases. The aforementioned flow conditions

of Fig. 4.12 are identical to that of Fig. 4.13. In order to draw a parallel between the two figures, it is helpful to note the increase in C^* , the ratio of heat capacity rates, accompanies an decrease in effectiveness and therefore a decrease in heat transfer. Thus, as C^* increases UA decreases, since UA is proportional to q and inversely proportional to ΔT_{lm} and the concavity of the curves in Fig. 4.12 is positive, hence q decreases at a rate higher than ΔT_{lm} decreases. With the UA dependency on q and ΔT_{lm} in mind Fig. 4.12 illustrates the monotonic increase of UA from the straight channel geometry, to Fractal 1, up to the optimized heat sink geometry, Fractal 2. These effects are also mirrored in the $\varepsilon - NTU$ curves of Fig. 4.8 where, for the same range of mass flow rates and value of C^* , the optimized design of Fractal 2 outperforms the other two. However, this relationship changes when the non-dimensional ratio of benefit to cost, ϵ_p is examined.

While the Fractal 2 geometry (optimized for the case of a heat sink in [21]) does result in higher heat transfer for a given mass flow rate, the overall flow power of the Fractal 2 geometry is higher compared to Fractal 1, evident by the difference in ϵ_p in Fig. 4.13. Higher flow power relative to the amount of heat that is transferred from the hot to the cold stream results in a lower ϵ_p . The geometry of Fractal 2 was generated for a heat sink a mass flow rate of ≈ 600 g/min and heat transfer of ≈ 6 kW subject to a maximum wall temperature constraint. The maximum wall temperature constraint of [21] requires that the mass flow rate increases as the heat flux increases. The lower performance of Fractal 2 compared to Fractal 1, evident in Fig. 4.13 and the incongruence of thermal constraints illustrates the need to optimize the heat exchanger geometry, since it is not subject to a maximum

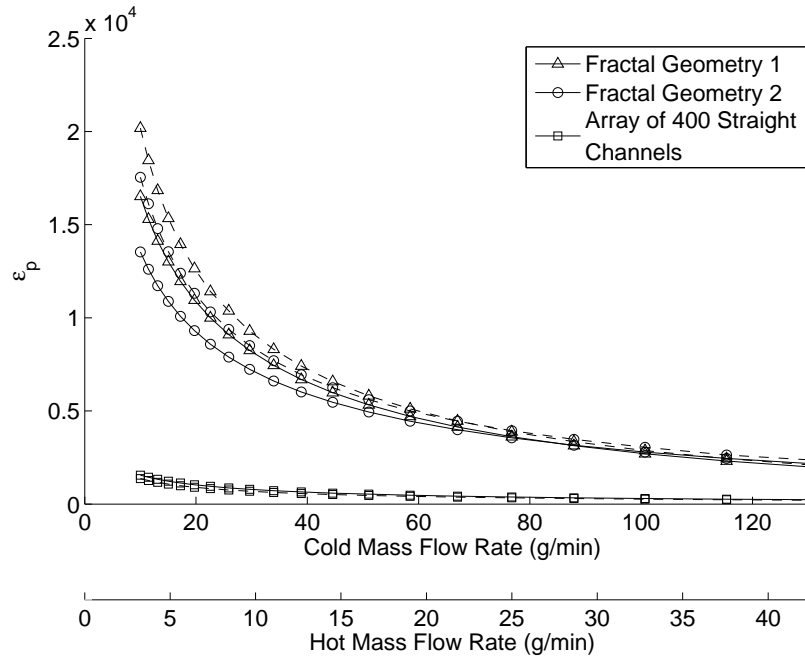


Figure 4.13: Performance parameter ϵ_p , versus mass flow rates of the hot and cold side of the heat exchanger obtained from the 1-D model for $C^* = 0.17$, and $\dot{m}_c = 10 - 130$ g/min obtained from the 1-D model, solid line (—) corresponds to co-flow, dashed line (---) corresponds to counter flow

wall temperature constraint, but may be subject to a fixed outlet temperature constraint.

4.3.4 Parametric variations subject to constrained mass flow rates

While it may be useful to maximize the heat transfer of a heat exchanger without regard to any constraints on the flow, such as the analysis of section 4.3.3, many applications require fixed cold side mass flow rates and temperature drops. For

Table 4.2: Parameters of the cost benefit analysis of Section 4.3.4

Parameter	Value	Unit
\dot{m}_c	1, 10, 100, 250, 500	g/min
ΔT_c	10, 20, 30, 50, 60	$^{\circ}C$
R	10-80 (20)	mm
w_m	50-150 (50)	μm
$error$	10^{-3}	$^{\circ}C$

example, say a manufacturing process needed to heat water from $25^{\circ}C$ up to a desired cold side exit temperature, $T_{c,e,set}$ of $65^{\circ}C$ at $10 g/min$, if the hot side inlet temperature is $150^{\circ}C$, there is a unique hot side mass flow rate for a given geometry which satisfies those requirements. A least-squares search was employed to find the unique \dot{m}_h that satisfies

$$|T_{c,e}(\dot{m}_h) - T_{c,e,set}| \leq error \quad (4.23)$$

Where $error$ is the minimum acceptable absolute difference between the desired cold side exit temperature, $T_{c,e,set}$ and the actual computed cold side exit temperature, $T_{c,e}$. The value $error$ was set at $10^{-3} ^{\circ}C$.

The aforementioned search for an \dot{m}_h that satisfies Eqn. 4.23 was performed under the conditions listed in Table 4.2. The parameters, \dot{m}_c and ΔT_c were varied under every possible combination with respect to each other. Conversely, w_m was varied in increments of $10 \mu m$ with respect to a constant disk radius, $R = 20mm$ and vice versa, with R varied in increments of $10 mm$ while $w_m = 50\mu m$.

Two geometries were employed in the study, one array of straight channels, one

square array of straight channels, and one fractal-like disk with the same geometric parameters listed in Table 4.1 for the “Straight Channel” and “Fractal 2” geometries, respectively. The purpose is to demonstrate any possible improvements in heat exchanger performance by utilizing the branching channels networks, opposed to an array of straight channels. In order to make a fair comparison of the performance between the two geometries, identical geometric extrema were given to the total size of each heat exchanger. For example, a fractal-like disk has a radius of R , thus the array of straight channels are packed into a square heat exchanger of dimensions $2R \times 2R$. This gives the array of straight channels the advantage of having a total planform area being $4/\pi$ times larger than the fractal-like disk.

Because the variation of the terminal channel width, w_m , and the disk radius, R , change the number of channels that may be packed into a fractal disk or array of straight channels, constraints were employed to increase the number of n_0 channels accordingly. The first constraint sets the number of n_0 channels according to the how the terminal channel width changes and is expressed as

$$n_{0,adj} = \left\lfloor \left(\frac{n_0 w_m}{w_{m,adj}} \right) \right\rfloor \quad (4.24)$$

Where the $\lfloor x \rfloor$ denotes the *floor* of the x , i.e. round x to the nearest integer towards negative infinity, while the *adj* subscript denotes the variable that is adjusted from its original baseline value in Table 4.1. Using these similar rules, the

adjusted n_0 value as R is changed may be formulated as

$$n_{0,adj} = \left\lfloor \left(\frac{n_0 R_{adj}}{R} \right) \right\rfloor \quad (4.25)$$

From the expressions in Eqns. 4.24 and 4.25, it is clear how as w_m decreases, and conversely as R increases, the packing of n_0 channels, or trees increases, the effects of which are illustrated and discussed in detail in Section 4.3.4.1. The remaining parameters were kept constant throughout the parametric analysis, a hot side inlet temperature, $T_{h,i} = 150^\circ C$, a cold side inlet temperature, $T_{c,i} = 25^\circ C$, an interface thickness, $t = 717 \mu m$ and channel height, $H = 250 \mu m$. The complete results from the parametric study are found in Appendix B, a small sub-sample of the results are presented in Sections 4.3.4.1 - 4.3.4.3 in order to highlight the salient findings of the parametric study.

4.3.4.1 Fractal versus Straight Channel Performance

The differences in performance between the Fractal 2 and Straight Channel geometries of Table 4.1 are illustrated in Figs. 4.14-4.18 as a function of cold side mass flow rate for both the counter and co-flow arrangements. What is clear in the figures, is that the fractal-like channels consistently outperform the straight channels in terms of ϵ_p (see Fig. 4.14) and ϵ (see Fig. 4.18) for a given cold side mass flow rate and temperature drop. Before further analysis is presented, it is important to call attention to the differences between the performance parameters,

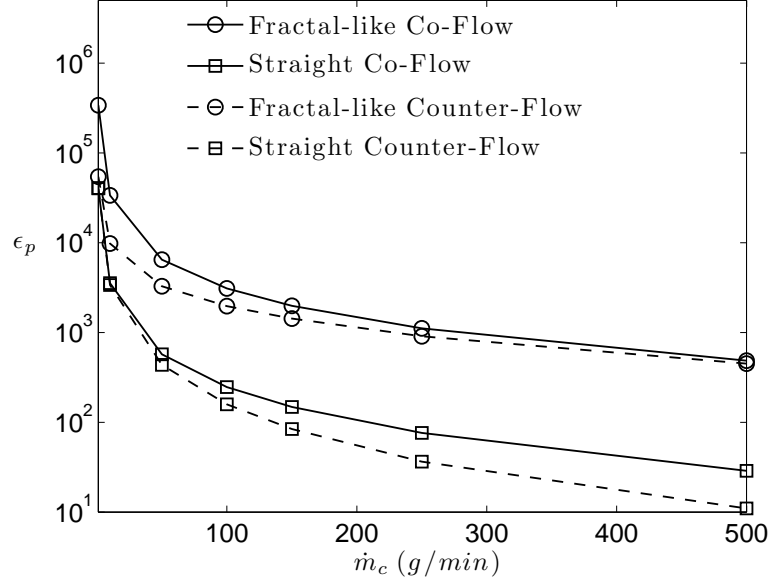


Figure 4.14: Benefit to cost ratio, ϵ_p versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ C$, $R = 20mm$ and $w_m = 50\mu m$

ϵ_p and ε .

The unique nature of the present analysis is that the fixed cold side mass flow rate and temperature drop yield a fixed heat exchanger heat transfer, thus ϵ_p (see Eqn. 4.22) is only a function of the combined flow power of the hot and cold sides. Conversely, ε (see Eqn. 4.20) is only a function of the minimum heat capacity rate, $C_{min} = (\dot{m}C_p)_{min}$, because the heat transfer, q and $(T_{hi} - T_{ci})$ are fixed.

The most universal trend of Figs. 4.14 and 4.18 is that both ϵ_p and ε are inversely proportional to the cold side mass flow rate, \dot{m}_c . This is simply due to the fact that as \dot{m}_c , \dot{m}_h must increase accordingly to meet the heat transfer

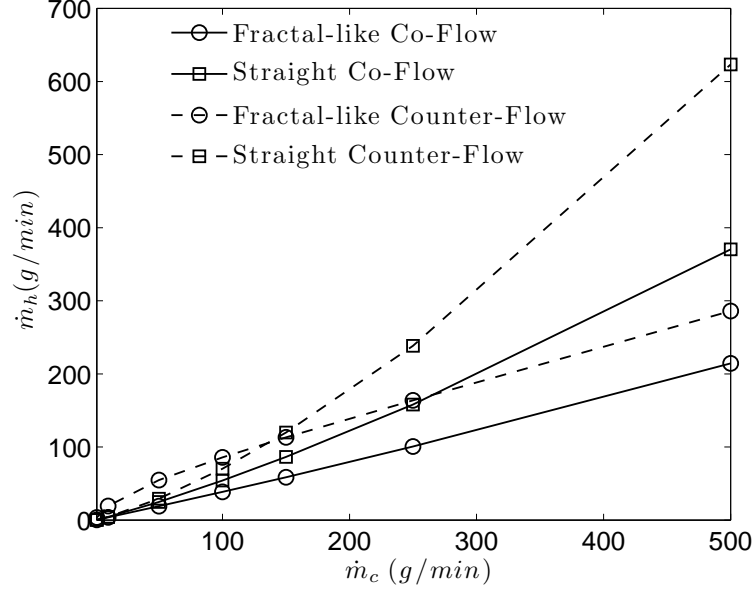


Figure 4.15: Heat exchanger hot side mass flow rate, \dot{m}_h versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ\text{C}$, $R = 20\text{mm}$ and $w_m = 50\mu\text{m}$

requirement. Higher mass flow rates mean higher flow power, reducing ϵ_p , and correspondingly, $(\dot{m}C_p)_{\min}$ increases, decreasing ϵ , see Eqns. 4.22 and 4.26. Improvements in ϵ_p by utilizing the fractal-like channels, compared to the array of straight channels are realized by the increase in cross-sectional flow area as each channel bifurcates, reducing the flow velocity and thereby reducing the pressure drop, as was reported by Pence [5]. In addition, lower hot side mass flow rates are achieved in fractal-like branching channel networks because of the greater than two fold increase in UA found in fractal-like geometries that make use of the redevelopment of the thermal boundary layers. A device with an increased UA does

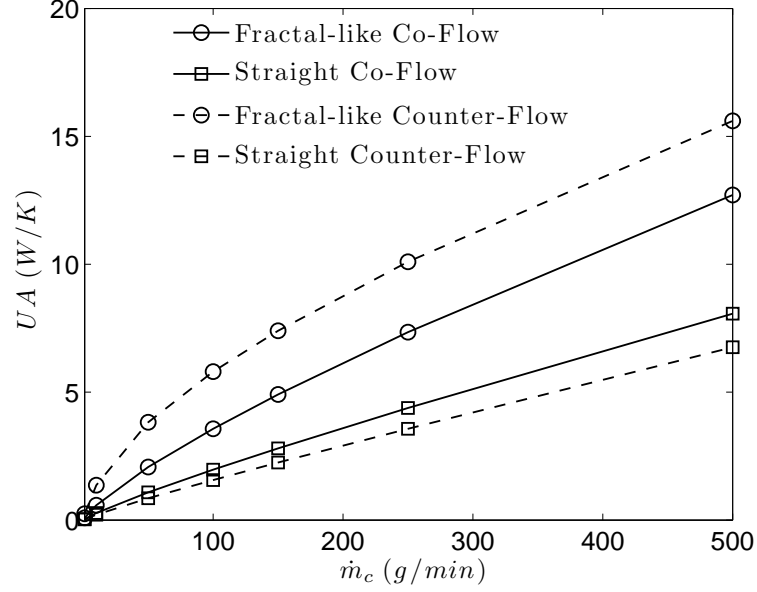


Figure 4.16: Heat exchanger UA versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ\text{C}$, $R = 20\text{mm}$ and $w_m = 50\mu\text{m}$

not need as high of a hot side mass flow rate to achieve the same heat transfer. Fig. 4.15 illustrates the differences in \dot{m}_h between the fractal and straight channel geometries for both counter and co-flow. Note that the hot side mass flow rate in the fractal geometry is consistently lower only in the parallel flow configuration. In the counter-flow configuration, for $\dot{m}_c \leq 150\text{g/min}$ the decrease in pressure drop is primarily what improves the ϵ_p performance of the fractal geometry. Furthermore, the higher \dot{m}_h in the counter-flow fractal networks, compared to the straight channels at $\dot{m}_c \leq 150\text{g/min}$ is explained by the sudden decrease in UA (as seen in Fig. 4.16) as the cold side mass flow rate decreases below 150 g/min . However, overall the fractal-like branching channel networks outperform the straight

channels, simply because straight channels of uniform cross-sectional area do not have the advantage of a decreasing flow velocity while simultaneously increasing the convective area. In addition, the UA of the straight channels is much lower than that of the fractal-like channels, evidenced in Fig. 4.16, simply because there is only one thermally developing region in a straight, non-branching channel.

The last thing to take note of in Fig. 4.16 is the difference in UA between the counter and co-flow configurations of both the fractal-like and straight channel geometries. The fractal-like counter-flow UA values are consistently higher than their co-flow counterparts, however, the opposite is true for the straight geometries, the result is a higher straight channel co-flow ε than counter-flow for the same mass flow rate. For both fractal geometries in Fig. 4.18, as expected, the counter-flow configuration results in higher heat transfers for a given \dot{m}_c compared to the parallel-flow configuration, indicative of higher UA values in the counter-flow case, confirmed in Fig. 4.16. However the opposite is true for the straight channel geometry, the parallel-flow configuration outperforms the counter-flow configuration, evidence of higher UA values in the parallel-flow case compared with the counter-flow arrangement for identical mass flow rates, channel geometry and inlet temperatures. Because the algorithm performs in accordance with the expectations of Eqns. 4.20 and 4.21, evidenced in Fig. 4.8, any computational error can be ruled out. The incongruous relationship between parallel and counter flow in straight channels may be made more clear by looking at the relationship between ε and UA , given as

$$\varepsilon \propto \frac{UA}{C_{min}} \quad (4.26)$$

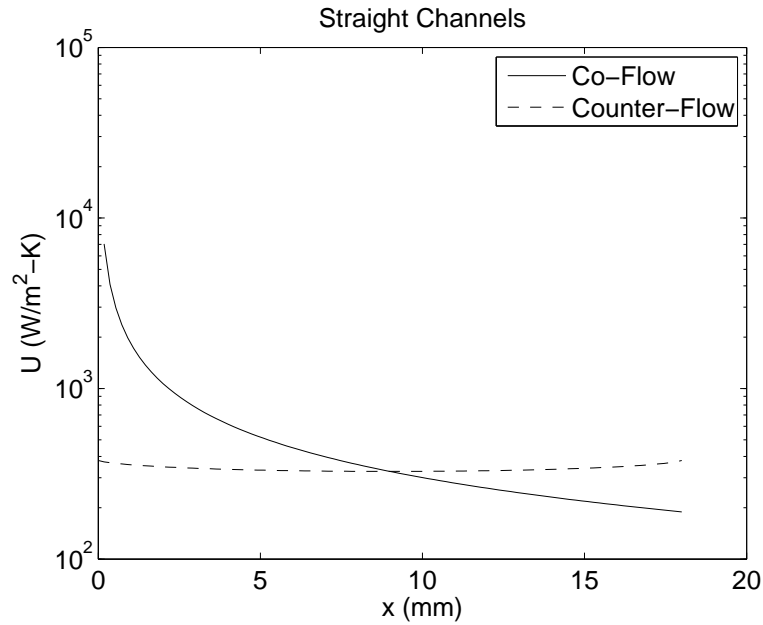


Figure 4.17: Overall heat transfer coefficient per unit area (U) versus axial distance of the straight channel geometry for $C^* = 0.17$, $T_{h,in} = 150^\circ\text{C}$ and $T_{c,in} = 25^\circ\text{C}$

where $C_{min} = (\dot{m}C_p)_{min}$. Figs. 4.15 and 4.16 illustrate that, for a fixed cold side mass flow rate, that an increase in UA results in a decrease in \dot{m}_h , which results in a higher ϵ because, for these cases, $C_{min} = (\dot{m}C_p)_h$. Under thermally developing flow conditions, the U value at any x location is dependent on both the hot and cold heat transfer coefficients, i.e. thermal resistances, R . The value of UA in Eqn. 4.26 is diminished with one high value of thermal resistance on either the hot or cold side. This is evidenced in Fig. 4.17, where in the counterflow case, the developing regions are at opposing ends of the channel, thus the U values are attenuated at each extrema because either the hot or cold thermal resistance is very high, leaving a low overall U at both inlets in the counter-flow arrangement.

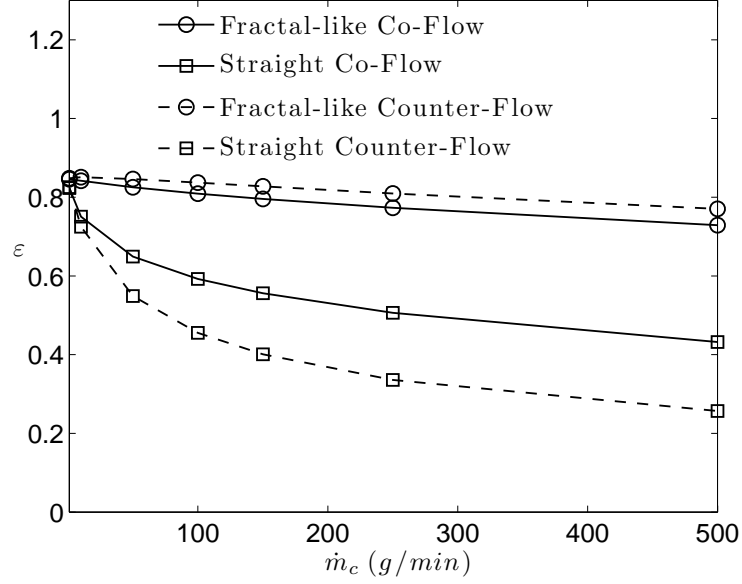
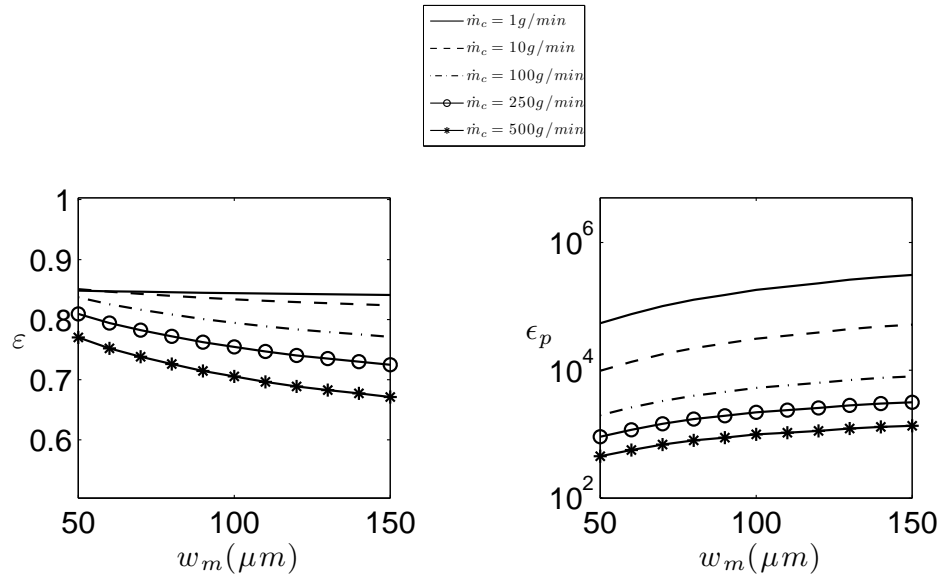


Figure 4.18: Heat exchanger effectiveness, ε versus cold side mass flow rate \dot{m}_c for the Fractal 2 and Straight Channel geometries, with $\Delta T_c = 20^\circ\text{C}$, $R = 20\text{mm}$ and $w_m = 50\mu\text{m}$

In the parallel-flow orientation, the developing regions of the hot and cold sides are alongside on another, complementing one another, resulting in a beneficial arrangement of thermal resistances, where the high thermal resistance at the outlet is negligible compared to the disproportionately low thermal resistance at the inlet of the parallel-flow configuration. For some perspective on the difference in U values between the parallel and counter configurations, the average counter-flow U in Fig. 4.17 is $340\text{ (W/m}^2\text{K)}$, while the average parallel flow U is $600\text{ (W/m}^2\text{K)}$. What is more interesting, is that the extreme attenuation of the counter-flow U is only present in the straight channel geometry. The fractal-like geometries utilize

shorter channel lengths, thus the thermally developing region takes up a higher percentage of the total channel length, meaning lower thermal resistances throughout each branch level resulting in higher overall UA values, seen in Fig. 4.16. Thus the disadvantage of the developing region at opposing ends of the channel is diminished in the fractal-like geometries. Other than the influence of the counter and co-flow arrangements on the ε values, the results of Fig. 4.18 mirror those of Fig. 4.14 with the exception that the values of ε do not improve by an order of magnitude as the values of ϵ_p do between the fractal and straight channel geometries.



(b) ε versus w_m for fractal-like channels in counter-flow configuration (c) ϵ_p versus w_m for fractal-like channels in co-flow configuration

Figure 4.19: Heat exchanger effectiveness, ε and benefit to cost ratio, ϵ_p for both the fractal and straight geometries versus terminal channel width, w_m for a $\Delta T_c = 20^\circ C$, and $R = 20 mm$ in the counter-flow configuration

4.3.4.2 Parametric Variation of the Terminal Channel Width, w_m

The variation of the terminal channel width was performed according to the conditions outlined in Section 4.3.4. The results of which are presented in the current section and should highlight the tradeoffs in ϵ_p and/or ε associated with increasing or decreasing the terminal channel width. Both ε and ϵ_p versus w_m are presented in Fig. 4.19 for the geometry Fractal 2 in Table 4.1 with $R = 20mm$ and $\Delta T_c = 20^\circ C$. The results of varying w_m in Fig. 4.19 are consistent with the universal trend of Figs. 4.14 and 4.18 in that both ε and ϵ_p decrease with increasing \dot{m}_c , see the beginning of Section 4.3.4.1 for an explanation.

The relationships between ϵ_p , ε as a function of w_m is not universal, increases in the terminal channel width result in a higher benefit to cost ratio, ϵ_p , but a lower value of effectiveness, ε . To explain how ϵ_p varies with w_m , the relationship between ΔP and w_m is presented in mathematical form as

$$\Delta P \propto \frac{1}{w_m D_h^2} \quad (4.27)$$

and is based on the analytical fully developed flow equation in [46] for a constant volumetric flow rate, Q . When the proportionality of Eqn. 4.27 is combined with Eqn. 4.22, a relationship in the form of

$$\epsilon_p \propto w_m D_h^2 \quad (4.28)$$

illustrates the relationship exhibited in Fig. 4.19(c); that as the terminal channel

increases, so should the benefit to cost ratio, ϵ_p .

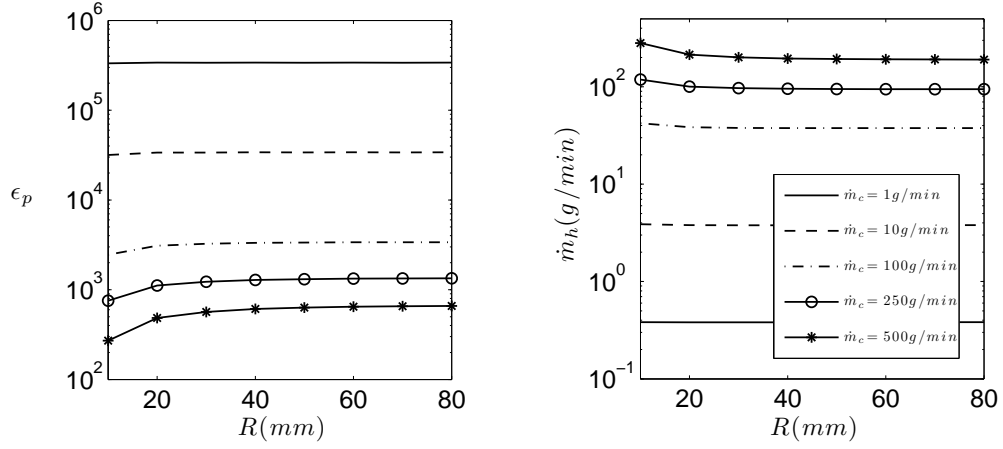
Conversely, the relationship between ε and w_m in Fig. 4.19(b) is monotonically decreasing, i.e. smaller channel widths yield greater values of heat exchanger effectiveness. The reason for this has to do with how ε varies with UA and UA varies with w_m . First, from Fig. 4.8, we know that $\varepsilon \propto UA/C_{min}$, and from Figs. 4.15 and 4.16 that $C_{min} \propto 1/UA$. Thus, as UA increases, so should the heat exchanger effectiveness, ε . Moreover, because UA is simply the inverse of the sum of the thermal resistances between the hot and cold fluid bulk temperatures, $UA \propto 1/D_h w$, resulting in

$$\varepsilon \propto \frac{1}{D_h w_m} \quad (4.29)$$

which corroborates the results illustrated in Fig. 4.19(c), that as w_m decreases the heat exchanger effectiveness increases. The increase is primarily due to two effects; *first*, the increase in convective surface area by packing in a greater number of smaller channels into a fixed volume (see Eqn. 4.24), *second*, from Eqn. 4.19, the heat transfer coefficient increases as the hydraulic diameter increases.

4.3.4.3 Parametric Variation of the Disk Radius, R

The variation of the disk radius, R was performed according to the conditions outlined in Section 4.3.4. The results of which are presented in the current section and highlight the relationship between both R and the performance parameters; ϵ_p and ε . Both ε and ϵ_p versus R are presented in Fig. 4.19 for the geometry Fractal 2 in Table 4.1 with $w_m = 50\mu m$ and $\Delta T_c = 20^\circ C$. The results of varying R in



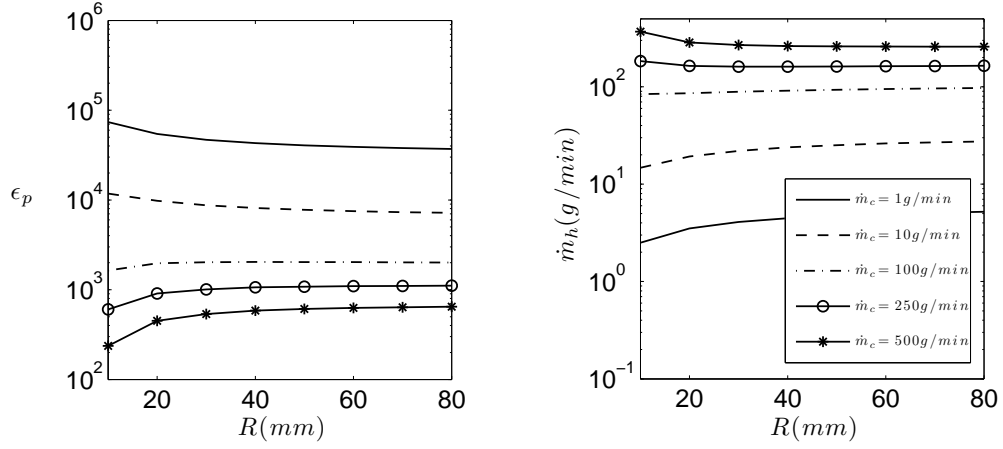
(a) Change in ϵ_p as a function of R in the co-flow configuration

(b) Change in \dot{m}_h as a function of R in the co-flow configuration

Figure 4.20: Change in ϵ_p and \dot{m}_h of Fractal 2 in co-flow as a function of R for $\Delta T_c = 20^\circ\text{C}$ and $w_m = 50\mu\text{m}$

Fig. 4.19 are consistent with the universal trend of Figs. 4.14 and 4.18 in that both ε and ϵ_p decrease with increasing \dot{m}_c , see the beginning of Section 4.3.4.1 for an explanation.

Changes in ϵ_p versus R for $\dot{m}_c = 1, 10, 100, 250$ and 500 g/min and $\Delta T_c = 20^\circ\text{C}$ are illustrated in Fig. 4.20(a), the accompanying variation in \dot{m}_h is also plotted in Fig. 4.20(b) for the same conditions. As the cold side mass flow rate is increased, the hot side mass flow rate also increases to achieve the required temperature drop in the manner outlined in Section 4.3.4. Thus, with both increasing mass flow rates, the flow power cost increases, decreasing the benefit to cost ratio in Figs. 4.20(a) and 4.21(a). Aside from changes in ϵ_p with respect to \dot{m}_c , it is interesting to point out the fact that there no noticeable change in ϵ_p or \dot{m}_c as a function of R for the co-flow configuration, illustrated in Figs. 4.20(a) and 4.20(b), for cold side mass



(a) Change in ϵ_p as a function of R in the counter-flow configuration (b) Change in \dot{m}_h as a function of R in the counter-flow configuration

Figure 4.21: Change in ϵ_p and \dot{m}_h of Fractal 2 in counter-flow as a function of R for $\Delta T_c = 20^\circ\text{C}$ and $w_m = 50\mu\text{m}$

flow rates of 1 and 10 g/min. This lack of trend can be attributed to the fact that at low flow rate ($\dot{m}_c = 1$ and 10 g/min), the flow is moving slow enough that there is no benefit from any increase in the total flow length provided by the increase in R . Conversely, at high flow rates, $\dot{m}_c \geq 100$ g/min, increased values of ϵ_p are seen by increasing the disk radius, R because of the accompanying decrease in \dot{m}_h , illustrated in Fig. 4.20(b). The decrease in \dot{m}_h is accomplished by increasing the hot side residence time and total convective surface area needed for values of $\dot{m}_c \geq 100$ g/min in the co-flow configuration.

The changes in ϵ_p and ε with R are not as clear-cut as the changes with respect to w_m in Section 4.3.4.2, especially for the counter-flow configuration. Figs. 4.21(a) and 4.21(b) show the same dependence of ϵ_p with R for high flow rates, but have an opposite trend for $\dot{m}_c \leq 10$ g/min. The opposing trend being for $R \leq 40$

mm, there is noticeable improvement in the benefit to cost ratio, ϵ_p , illustrated in Fig. 4.21(a). The reason this trend is only seen for low flow rates in the counter-flow configuration is because there is an additional benefit in the counter-flow configuration by reducing the total flow length, R . When $C_h > C_c$, as is the case when $\dot{m}_c \leq 10$ g/min, there is no benefit from increasing the residence time in the hot side, simply because the residence time in the cold side is much higher. Thus, if the time the fluid spends flowing through the hot side (i.e. residence time) is fixed, decreases in the length, result in a decreased mass flow rate, improving the benefit to cost ratio, ϵ_p , illustrated in Fig 4.21. The benefit from decreasing the mass flow rate is not possible in the co-flow configuration because there exists no high ΔT_h potential, whereas a ΔT_h potential of 130°C exists in the counter-flow configuration.

Heat exchanger effectiveness, ε , was also examined as a function of disk radius, R . The trends are more clear-cut than those of Figs. 4.20 and 4.21, ε consistently improves with increasing disk radius, simply due to the increase in convective surface area because of the increase in R , illustrated in Figs. 4.22 and 4.23. What is interesting to point out, is that there seems to be a maximum effectiveness under the co-flow configuration, $\varepsilon_{max,co}$, and is evident in Fig. 4.22. This maximum is the result of the fixed temperature drop, ΔT_c applied to the cold side. The variable, $\varepsilon_{max,co}$ can be found from the Eqn. 4.20 for ε , under the conditions where $C_h = C_{min}$ and is expressed as

$$\varepsilon_{max,co} = \frac{(T_{h,in} - T_{c,e})}{(T_{h,in} - T_{c,in})} \quad (4.30)$$

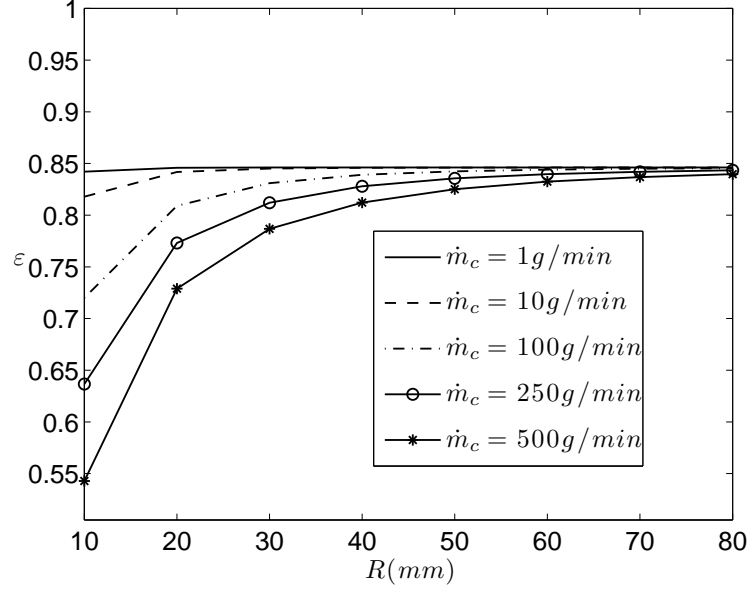


Figure 4.22: Heat exchanger effectiveness, ε versus cold side mass flow rate, \dot{m}_c for Fractal 2 in the co-flow configuration with $w_m = 50 \mu m$ and $\Delta T_c = 20^\circ C$

For the conditions in Fig. 4.22, $\varepsilon_{max,co}$ is calculated as 0.846, which matches the asymptote in Fig. 4.22. For counter flow, the maximum effectiveness, is 1.0, since the cold side inlet temperature, T_{ci} constrains the hot side exit temperature, T_{he} when $C_{min} = C_h$. Thus, no asymptote is present in the results of ε versus R in Fig. 4.23, clearly indicating counter-flow outperforms co-flow configurations as far as ε is concerned.

Conversely, there exists a minimum effectiveness, ε_{min} when $C_{min} = C_c$, and is the same for co and counter-flow simply due to the fact that a fixed cold side temperature drop is applied. The form of the equation is similar to that of Eqn. 4.30

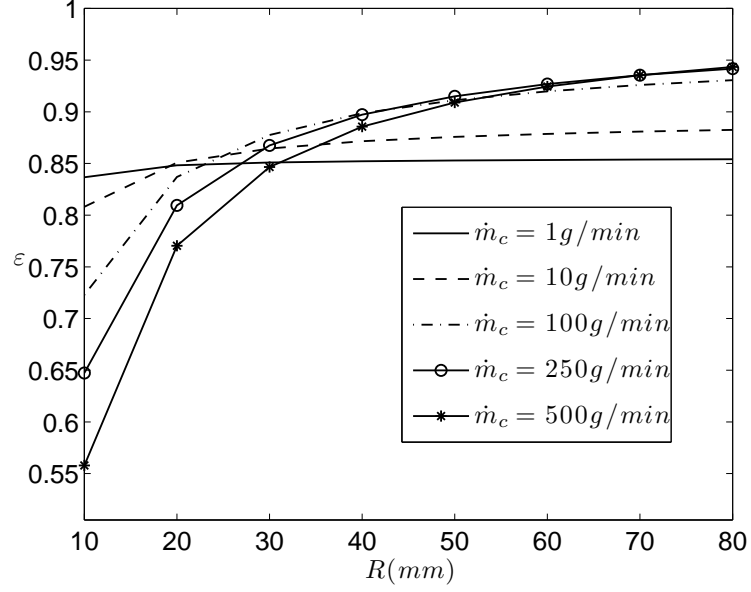


Figure 4.23: Heat exchanger effectiveness, ε versus cold side mass flow rate, \dot{m}_c for Fractal 2 in the counter-flow configuration with $w_m = 50 \mu\text{m}$ and $\Delta T_c = 20^\circ\text{C}$

and is expressed as

$$\varepsilon_{min} = \frac{T_{c,e} - T_{c,in}}{T_{h,i} - T_{c,in}} \quad (4.31)$$

For the conditions of Figs. 4.22 and 4.23, $\varepsilon_{min} = 0.133$, by examining Figs. 4.22 and 4.23, it is clear that the values presented in this section are well above $\varepsilon_{min} = 0.133$.

However, the minimum limitation can be seen in the figures of Appendix B.

4.4 Summary and Conclusions

A one-dimensional model of a fractal-like branching channel heat exchanger was developed to determine the pressure and bulk fluid temperature distributions. The

model was constructed to be both reasonable in terms of quantifying heat transfer and efficient in that there is little computational effort to solve the system of equations. The heat transfer coefficient is employed in lieu of solving any discrete y distribution of temperature in the energy equation. The validity of a heat transfer coefficient correlation applied to a heat exchanger was determined by analysis of the results of a simple two stream heat exchanger modeled by a three dimensional computational fluid dynamics package. Shah and London's [7] correlation proved to describe the CFD data to a degree no more statistically significant than a power fit to the data. Moreover, the fin efficiency was found to be important in capturing the difference in heat transfer between a material of high versus low thermal conductivity.

While the fin efficiency of the channel walls in the heat exchanger was found to be important for the particular geometry, a more detailed analysis is necessary to determine under what conditions it can be neglected, i.e., for what combination(s) of fin height, material type, etc will result in a enough heat transfer to ignore the fin temperature distribution effects. Additionally, an analysis of the heat exchanger under a fixed cold side mass flow rate and fixed inlet and exit temperatures was performed to determine the change in performance of different heat exchanger designs under the process cooling constraints. It was found that decreases in the terminal channel width increased the effectiveness but decreased the benefit to cost ratio of the heat exchanger. Furthermore, increases in the disk radius, R , yielded increases in heat exchanger effectiveness but the relationship between R and benefit to cost was found to depend highly on the cold side mass flow rate as

well as whether the heat exchanger is operating in co or counter-flow configuration.

Bibliography

- [1] Tuckerman, D. B., and Pease, R. F. W., 1981, “High-performance heat sinking for vlsi”, *IEEE Electronic Devices Letters*, **EDL-2**, pp. 126–277.
- [2] Alharbi, A. Y., Pence, D. V., and Cullion, R. N., 2004, “Thermal characteristics of microscale fractal-like branching channels”, *Journal of Heat Transfer*, **126**(5), pp. 744–752.
- [3] Alharbi, A., Pence, D. V., and Cullion, R., 2003, “Fluid flow through microscale fractal-like branching channel networks”, *Journal of Fluids Engineering*, **125**, pp. 1051–1057.
- [4] Pence, D. V., 2000, “Improved thermal efficiency and temperature uniformity using fractal-like branching channel networks”, In Proceedings of the International Conference on Heat Transfer and Transport Phenomena in Microscale, Bergell House, pp. 142–148.
- [5] Pence, D. V., 2002, “Reduced pumping power and wall temperature in microchannel heat sinks with fractal-like branching channel networks”, *Microscale Thermophysical Engineering*, **6**, pp. 319–330.
- [6] White, F., 1991, *Viscous Fluid Flow*, 2nd ed., McGraw-Hill Inc.
- [7] Shah, R. K., and London, A. L., 1978, *Laminar Flow Forced Convection in Ducts: A Source Book for Compact Heat Exchanger Analytical Data*, Academic Pr.
- [8] Pence, D. V., Enfield, K., and Narayanan, V., 2005, “Optimization of single phase fractal-like branching flow heat sinks”, In ECI International Conference on Heat Transfer and Fluid Flow in Microscale.
- [9] Bejan, A., 1997, “Constructal tree network for fluid flow between a finite-size volume and one source or sink”, *Revue Générale de Thermique*, **36**, pp. 592–604.

- [10] Bejan, A., and Errera, M., 2000, “Convective trees of fluid channels for volumetric cooling”, *International Journal of Heat and Mass Transfer*, **43**, pp. 3105–3118.
- [11] Lorente, S., Wechsato, W., and Bejan, A., 2002, “Tree-shaped flow structures designed by minimizing path lengths”, *International Journal of Heat and Mass Transfer*, **45**, pp. 3299–3312.
- [12] Wechsato, W., Lorente, S., and Bejan, A., 2002, “Optimal tree-shaped networks for fluid flow in a disc-shaped body”, *International Journal of Heat and Mass Transfer*, **45**, pp. 3299–3312.
- [13] Wechsato, W., Lorente, S., and Bejan, A., 2003, “Dendritic heat convection on a disc”, *International Journal of Heat and Mass Transfer*, **46**, pp. 4381–4391.
- [14] Gonzales, M., Jelisavcic, N., Moral, R., Saho, D., Dulikravich, G., and Martin, T., 2007, “Multi-objective design optimization of topology and performance of branching networks of cooling passages”, *International Journal of Thermal Sciences*, **46**, pp. 1191–1202.
- [15] Chapra, S., and Canale, R., 2002, *Numerical Methods for Engineers*, McGraw-Hill Inc., New York.
- [16] Koza, J., 1992, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge.
- [17] Pence, D., 2010, “The simplicity of fractal-like flow networks for effective heat and mass transport”, *Experimental Thermal and Fluid Science*, **34**(4), pp. 474 – 486, ECI International Conference on Heat Transfer and Fluid Flow in Microscale.
- [18] Bowers, M., and Mudawar, I., 1994, “High flux boiling in low flow rate, low pressure drop mini-channels and micro-channel heat sinks”, *International Journal of Heat and Mass Transfer*, **37**(2), pp. 321–332.
- [19] Bowers, M., and Mudawar, I., 1994, “Two-phase electronic cooling using mini-channel and micro-channel heat sinks: Part 2 - flow rate and pressure drop constraints”, *Journal of Electronic Packaging*, **116**(4), pp. 298–305.

- [20] Daniels, B. J., 2008, “A study of adiabatic and diabatic flow boiling in parallel microchannels and fractal-like branching microchannels”, PhD thesis, Oregon State University.
- [21] Heymann, D., Pence, D., and Narayanan, V., 2010, “Optimization of fractal-like branching microchannel heat sinks for single-phase flows”, *International Journal of Thermal Sciences*, **49**(8), pp. 1383–1393.
- [22] Edward, L. M., 2008, “Characterization of flow boiling in a fractal-like branching microchannel network”, Master’s thesis, Oregon State University.
- [23] Cullion, R., Pence, D., Liburdy, J., and Narayanan, V., 2007, “Void fraction variations in a fractal-like branching microchannel network.”, *Heat Transfer Engineering*, **28**(10), pp. 806 – 816.
- [24] Kandlikar, S., 2004, “Heat transfer mechanisms during flow boiling in microchannels”, *Journal of Heat Transfer*, **126**, pp. 8–16.
- [25] Lee, H. J., Liu, D. Y., and chune Yao, S., 2010, “Flow instability of evaporative micro-channels”, *International Journal of Heat and Mass Transfer*, **53**(9-10), April, pp. 1740–1749.
- [26] Qu, W., and Mudawar, I., 2004, “Measurement and correlation of critical heat flux in two-phase micro-channel heat sinks”, *International Journal of Heat and Mass Transfer*, **47**(10-11), pp. 2045 – 2059.
- [27] Revellin, R., and Thome, J. R., 2008, “A theoretical model for the prediction of the critical heat flux in heated microchannels”, *International Journal of Heat and Mass Transfer*, **51**(5-6), pp. 1216 – 1225.
- [28] Revellin, R., Thome, J. R., Bejan, A., and Bonjour, J., 2009, “Constructal tree-shaped microchannel networks for maximizing the saturated critical heat flux”, *International Journal of Thermal Sciences*, **48**(2), pp. 342 – 352, Nano Micro Mini Channels and Computational Heat Transfer.
- [29] Mauro, A., Thome, J., Toto, D., and Vanoli, G., 2010, “Saturated critical heat flux in a multi-microchannel heat sink fed by a split flow system”, *Experimental Thermal and Fluid Science*, **34**, pp. 81–92.
- [30] Lockhart, R., and Martinelli, R., 1949, “Proposed correlation of data for isothermal two-phase, two-component flow in pipes”, *Chemical Engineering Progress*, **45**, pp. 39–48.

- [31] Zivi, S., 1964, “Estimation of steady-state steam void-fraction by means of the principle of minimum entropy production”, *Journal of Heat Transfer*, **86**, pp. 247–252.
- [32] Guo, J., Cheng, L., and Xu, M., 2010, “Multi-objective optimization of heat exchanger design by entropy generation minimization”, *Journal of Heat Transfer*, **132**(8), p. 081801.
- [33] Ishikawa, H., and Hobson, P. A., 1996, “Optimization of heat exchanger design in a thermoacoustic engine using a second law analysis”, *International Communications in Heat and Mass Transfer*, **23**(3), pp. 325–334.
- [34] Shiba, T., and Bejan, A., 2001, “Thermodynamic optimization of geometric structure in the counterflow heat exchanger for an environmental control system”, *Energy*, **26**(5), pp. 493–512.
- [35] Rogiers, F., and Baelmans, M., 2009, “Towards maximal heat transfer rate densities for small-scale high effectiveness parallel-plate heat exchangers”, *International Journal of Heat and Mass Transfer*, **53**(4), pp. 605–614.
- [36] Bejan, A., 2002, “Dendritic constructal heat exchanger with small-scale cross-flows and larger-scales counterflows”, *International Journal of Heat and Mass Transfer*, **45**(23), pp. 4607–4620.
- [37] da Silva, A. K., Lorente, S., and Bejan, A., 2004, “Constructal multi-scale tree-shaped heat exchangers”, *Journal of Applied Physics*, **96**(3), pp. 1709–1718.
- [38] Zimparaov, V. D., da Silva, A. K., and Bejan, A., 2006, “Constructal tree-shaped parallel flow heat exchangers”, *International Journal of Heat and Mass Transfer*, **49**(23-24), pp. 4558–4566.
- [39] Chong, S. H., Ooi, K. T., and Wong, T. N., 2002, “Optimisation of single and double layer counter flow microchannel heat sinks”, *Applied Thermal Engineering*, **22**(14), October, pp. 1569–1585.
- [40] Incropera, F. P., and DeWitt, D. P., 1981, *Fundamentals of Heat Transfer*, Wiley.
- [41] Shah, R. K., and Sekulic, D. P., 2003, *Fundamentals of Heat Exchanger Design*, John Wiley and Sons.

- [42] Stull, C., and Sabo, M., 2008, Design of a microscale branching channel heat exchanger, Senior project report, Oregon State University, September.
- [43] Kline, S., and McClintock, F., 1953, “Describing uncertainties in single-sample experiments”, *Mechanical Engineering*, **75**, pp. 3–9.
- [44] Kwak, Y., Pence, D., Liburdy, J., and Narayanan, V., 2009, “Gas-liquid flows in a microscale fractal-like branching flow network”, *International Journal of Heat and Fluid Flow*, **30**(5), pp. 868 – 876, The 3rd International Conference on Heat Transfer and Fluid Flow in Microscale, The 3rd International Conference on Heat Transfer and Fluid Flow in Microscale.
- [45] Krebs, D., Narayanan, V., Liburdy, J., and Pence, D., 2010, “Spatially resolved wall temperature measurements during flow boiling in microchannels”, *Experimental Thermal and Fluid Science*, **34**(4), pp. 434 – 445, ECI International Conference on Heat Transfer and Fluid Flow in Microscale.
- [46] White, F., 2003, *Fluid Mechanics*, 5th ed., McGraw-Hill Inc.

APPENDICES

Appendix A – MATLAB Code

A.1 Single-Phase Heat Sink Optimization Code

A.1.1 One-Dimensional Single Phase Heat Sink Model

```

% clear all

clear AC          L          SD          cp
      mdot        xLehplus    %h
clear AR          Mdot       SDtable     d          h_plot
      mtrue       xLet
clear ARs         Nuz        Tb          dP         hfluid      mu
      xLetstar
clear ARtab       Nuzplot    Tbplot      hs          n
      z           % data_store
clear ARtemp      P          Tin         den         htc
      zend        % no
clear Ac          Pexit      Tw          dhdz        i          ns
      zeta
clear Apstot      Power      Twmax       dratio      k          nu
      zplot
clear Aptot       Pplot      Twplot      ds          kfl
      rho         zs
clear Aptotf      Pr         U           kfluid      ri
      zsend       % dx
clear Astot       Prplot     V           dxs         kset        ro
      zstar
clear Atot        Psat       alpha        dxtrue      ktemp       sd
clear Atotb       Psatplot   alphas      energy       steps
      %ktrue
clear CfpRe       Pstart     alphaset    factor      l
      totalsteps
clear CompTitle   Ptemp      ans          %w
      flux lengthf
clear D           Q          fluxf        ls
      %wN beta
clear            c          fluxs        lstemp      ws
      %Qtot K

```

```

clear Kinf          Re          m          xLeh %gamma
count

% close all
% home
%%setting global values to determine the Re through a straight
    channel to get same pressure drop as through fractal network
% global ARs ws hs ds ls ns rho nu Pnew power
% global k Mdot h fluxf alpha Q

%%see paper by Pence (2000) for nomenclature definitions...
%%assumes hydraulic diameter is sufficient for computations...
    change if otherwise (line 15??)
%%assumes water at this point

%% Fractal channel input
n=2;                %number of branches following each fractal
    bifurcation
% no=1;              %number of initial, 0th order, branches
% ktrue=4+1;         %number of branching levels, k, plus '1' for the 0
    th branch
% w(ktrue)=100;      %terminal branch width in microns
% lengthf=16.3;      %total fractal branch length in mm
% h=250;             %constant channel depth in microns
% flux=350;          %Watts per centimeter squared for the fractal
    channel only
% Qtot=10;           %total fractal flow rate in ml/s
%TEMPORARY ONLY
%Qtot=2;

%% straight channel input %either square or rectangular - match the
    terminal hydraulic diameter
%ws=w(ktrue); %terminal branch width of straight channel in microns
%hs=h; %Input constant channel depth in microns
ws=4*h*w(ktrue)/(2*h+2*w(ktrue));
hs=ws;

%% comparison information
dratio=1;           % (0 ==> hydraulic diameter ratio is equal to beta -
    NOT RECOMMENDED )
                    % (1 ==> width ratio is equal to beta - RECOMMENDED
                        BASED ON OPTIMIZATION)
                    % (2 ==> area ratio is equal to beta^2, which is
                        equal to the width ratio equal to beta^2 for a
                        constant channel height - NOT RECOMMENDED)

```



```

lstep=0;          % to make channel length identical to fractal channel
                  length enter zero (0)
                  % to make channel length fit a square plate foot
                  print, select (-1)
                  % to set a channel length in mm - simply specify the
                  number
alphaset=2;       %plate surface to convective surface area ratio
                  % (1 ==> As_s/As_f=1 & Ap_s/Ap_f=1) == plate area
                  % assumed to be equal to total convective area
                  % (0 ==> As_s/As_f=1 & Ap_s/Ap_f≠1) == channel
                  % surface areas identical - plate areas differ
                  % (2 ==> Ap_s/Ap_f=1 & As_s/As_f≠1) == plate areas
                  % identical - channel surface areas differ
energy=1;         % to make total energy input equal between the
                  straight and fractal devices, set equal to (1)
                  % NOTE: when running alphaset=1 or alphaset=2 this
                  % will be the default (i.e. energy=1)...
                  % to make flux equal between devices, set equal to
                  (0)
AC=2;             % flow optimization factor AC = 2 for width ratio -
                  RECOMMENDED- area conserving for hydraulic diameter ratio - see
                  notes from 12-09-02
                  % flow optimization factor AC = 3 for hydraulic
                  diameter

%filename for output
filename=['Q',num2str(Qtot),'alpha',num2str(alphaset),'ls',num2str(
lstep),'old'];
% filename=['h',num2str(h),'L',num2str(lengthf),'k',num2str(ktrue),'
wt',num2str(w(ktrue)),'Q',num2str(Qtot),'q',num2str(flux),'ls',
num2str(lstep),'as',num2str(alphaset),'es',num2str(energy),'AC',
num2str(AC),'.mat'];

%% additional input items - may change
% dx=0.01;        %initial spatial increment for pressure drop and
                  Nusselt number calculations (changes with l)
Pexit=0;%101300;   %exit pressure at exit plenum [Pa]

%%geometry restrictions that won't change
D=2;              %Euclidean dimension

%%keep items below - integrate into the optimization program
Q(1)=Qtot/no;     % flow rate through each initial branch
fluxf=flux*10000; % conversion of flux to Watts per meter squared (
                  assigned to fractal network)

```

```

Tin=22+273;          % inlet plenum temperature in Kelvin for branching
                      channel network
%Tbs(1)=Tb(1);       % inlet plenum temperature in Kelvin for
                      straight channels

%%assumes water as fluid with constant properties evaluated at inlet
  T of 22 deg Celcius
rho=998.2;           %kg/m-m-m
mu=0.001002;         %kg/m-s
nu=mu/rho;           %m-m/s
cp=4183;             %specific heat in J/kg-K at 295 K
kfl=0.603;           %thermal conductivity W/m-K at 295 K
Pr=(rho*cp*nu)/kfl;  %calculation of Prandtl number

%%computing mass flow rate, branching dimensions, surface area, and
  entrance lengths for the fractal array
mdot=rho*Q(1)/1000000; %total mass flow rate kg/
                      s through a single branch leaving plenum
% beta=n^(-1/AC);      %diameter ratio
% gamma=n^(-1/D);      %length scale ratio
d(ktrue)=4*(w(ktrue)*h)/(2*w(ktrue)+2*h); %hydraulic diameter in
                      microns
k=ktrue;              %assign an increment to
                      retain true number of branchings, ktrue
kset=[1:1:ktrue];     %set up vector for
                      computing terminal branch length
l(ktrue)=lengthf/sum(1./gamma.^(kset-1)); %determination of
                      terminal branch length
Q(ktrue)=Q(1)/(n^(ktrue-1)); %evaluation of flow rate
                      through terminal branch

while k>1
    if dratio==0;
        w(k-1)=w(k)*h/(beta*(w(k)+h)-w(k)); %determination of lower
                      order branching widths - based on hydraulic diameter ratio
                      equal to beta
    elseif dratio==1;
        w(k-1)=w(k)/beta; %width ratio equal to
                      beta
    elseif dratio==2;
        w(k-1)=w(k)/beta^2; %actual area ratio equal
                      to beta
    end
    l(k-1)=l(k)/gamma; %determination of lower
                      order branching lengths

```

```

%      d(k-1)=d(k)/beta;                                %determination of lower
order hydraulic diameters
d(k-1)=2*h*w(k-1)/(h+w(k-1));
Q(k-1)=Q(k)*n;                                           %evaluation of flow rate
      through each branch
k=k-1;                                                   %increment for k
end % end of while

V=Q./(w./10000*h/10000);                                %velocity in cm/s
      through each branch
Re=V/100.*d/1000000/nu;                                %calculation of
      Reynolds number for each branch
xLeh=d./1000.*(0.6./(1+0.035.*Re)+0.056.*Re);           %calculation of
      hydrodynamic entrance length - mm
xLet=d./1000.*(0.6./(1+0.035.*Re*Pr)+0.056.*Re*Pr);     %calculation of
      thermal entrance length - mm
xLehplus=xLeh.*1000./(d.*Re);                           %dimensionless
      hydrodynamic entrance length
xLetstar=xLet.*1000./(d.*Re*Pr);                         %dimensionless
      thermal entrance length
ktemp=[1:1:ktrue];                                       %temporary vector
      width for branching levels
Atotb=(n.^(ktemp-1).*1.*(w+h)*2/1000);                  %calculation of
      the total convective surface area per branch - mm^2
Atot=sum(Atotb)*no;                                       %calculation of
      the total convective surface area for an entire branching flow
      network - mm^2

%determine plenum size
ri=no*(w(1)/1000)/(2*pi)*1.05; %inside diameter of inlet plenum - mm
ro=ri+sum(l); %outside diameter of plate - mm
Aptotf=pi*(ro^2-ri^2); %fractal plate surface area in mm^2

if alphaset==0; %plate surface area to be included in
      analysis - but channel surface area of straight array is
      identical to that of the fractal array
      Aptot=Aptotf; %fractal plate surface area in mm^2
      alpha=Aptot/Atot; %fractal plate to convective surface
      area ratios
elseif alphaset==1; %all energy is provided through walls
      of channels (plate area is neglected)
      Aptot=Atot; %total plate area is a nonreal
      quantity
      alpha=alphaset; %set alphas to unity for case when
      plate surface area is identical to convective surface area

```

```

elseif alphaset==2;           %plate surface area to be included in
    analysis - but plate contact area of straight array is identical
    to that of the fractal array
    Aptot=Aptotf;             %fractal plate surface area in mm^2
    alpha=Aptot/Atot;         %fractal plate to convective surface
    area ratios
end % end of if

if lstemp==0
    ls=sum(l);                %total straight channel length
elseif lstemp==-1
    ls=sqrt(Aptot);           %determine the channel length necessary to
    achieve a square foot print for the plate area
else
    ls=lstemp;                %maintains assigned straight channel length
end % end of if

if alphaset==0;               %plate surface area
    to be included in analysis - but channel surface area (convective)
    of straight array is identical to that of the fractal array
    ns=round(Atot/(ls*2*(ws+hs)/1000)); %determine the number
    of channels that will give the same convective area as the
    fractal
    Apstot=(ns*(2*ws/1000)+ws/1000)*ls; %determine the plate
    surface area for a straight channel array with identical
    convective surface area as the fractal array
    Astot=ns*2*(ws+hs)/1000*ls; %determine the total
    convective surface area based on the number of channels - mm^2
    alphas=Apstot/Astot; %straight channel
    plate to convective surface area ratios
elseif alphaset==1;           %all energy is
    provided through walls of channels (plate area is neglected)
    ns=round(Atot/(2*ls*(ws+hs)/1000)); %determine the number
    of straight channels for equal convective surface as a
    fractal network
    Astot=ns*2*(ws+hs)/1000*ls; %determine the total
    convective surface area based on the number of channels - mm^2
    Apstot=Astot; %set plate area equal
    to convective surface area (i.e. neglect plate area)
    alphas=alphaset; %set alphas to unity
    for case when plate surface area is identical to convective
    surface area
elseif alphaset==2;           %plate surface area
    to be included in analysis - but plate contact area of straight
    array is identical to that of the fractal array

```

```

ns=round((Aptot/ls-ws/1000)/(2*ws/1000)); %determine the number
    of straight channels to equal convective surface area of a
    single branching channel network
Astot=ns*2*(ws+hs)/1000*ls; %determine the total
    convective surface area based on the number of channels - mm2
Apstot=(ns*(2*ws/1000)+ws/1000)*ls; %determine plate
    surface area based on the number of parallel channels - mm2
alphas=Apstot/Astot; %straight channel
    plate to convective surface area ratios
end % end of if

%Straight channel bulk fluid and wall surface temperatures
if energy==0; %computations based on
    the assumption that the heat flux between the two devices are
    equal
    fluxs=fluxf; %flux for the straight
    array is set equal to the fractal array
elseif energy==1; %based on the assumption
    that the total heat input to the straight array is the same as the
    fractal array
    fluxs=fluxf*Aptot/Apstot; %determine the straight
    flux from the total energy to the fractal array
end % end of if

steps=1./dx; %determination of number of
    increments of dx in length of channel branch
mtrue=round(steps); %actual integer number of steps per
    branch
totalsteps=sum(mtrue); %total number of steps from inlet to
    exit of heat sink
m=totalsteps; %initialization for reverse ordering
    pressure
dxtrue=1./mtrue; %true spatial increment to
    accommodate full length of each branch

load sd_data_table; %load coefficient data - already
    determined from sd_tdata_table_formulation.m
ARtab=SD_table(:,1); %reassignment of data
SD(:,:)=SD_table(:,2:4);

% THIS IS THE BEGINNING OF KENT ENFIELD'S VARIABLE PROPERTY SECTION
L = 1; % to avoid confusion
    between the number "1" and the letter "l".
Mdot = Q*rho/1e6; % [kg/s] With changing
    density, mass flow rate is constant. Volumetric flow rate is not.

```

```

dhdz      = fluxf*alpha*(2*(h+w))./Mdot/1e6; % Heat load per channel
length [J/m].
Ac         = h*w; % Cross-channel area [
microns^2].

zend = round(L./dxtrue + 1);
for k = 1:ktrue
    z(k,1:zend(k)) = 0:dxtrue(k):L(k);
end

% enthalpy of flow [J/kg]
hfluid(1,:) = dhdz(1)*z(1,:)/1e3 + hf_lof_T(Tin-273);
for k = 2:ktrue
    hfluid(k,1:zend(k)) = dhdz(k)*z(k,1:zend(k))/1e3+hfluid(k-1,zend(
        k-1));
end

Tb         = Tofh(hfluid); % Bulk temperature [deg C] based on
enthalpy.
den        = 1./vf(PofT(Tb)); % Density of liquid [kg/m^3].
nu         = nuf(PofT(Tb)); % Kinematic viscosity [m^2/s].
Pr         = PrfofT(Tb); % Prandtl number
kfluid     = kf(PofT(Tb)); % Thermal conductivity [W/m-K].
Pstart     = 0; % Initial guess for inlet pressure [Pa].

for k = 1:ktrue
    U(k,1:zend(k)) = Mdot(k)./den(k,1:zend(k))/Ac(k)*1e12;
    % average velocity [m/s]
    Re(k,1:zend(k)) = U(k,1:zend(k))./nu(k,1:zend(k))*d(k)/1e6;
    % local Re based on diameter
    zeta(k,1:zend(k)) = z(k,1:zend(k))/1e3./Re(k,1:zend(k))/d(k)*1e6;
    % zeta parameter
    zstar(k,1:zend(k)) = zeta(k,1:zend(k))./Pr(k,1:zend(k));
    % non-dimensional length for Nux
    zeta(k,1) = eps;
    zstar(k,1) = eps;
    AR = min(h,w(k))/max(h,w(k));
    [CfpRe,Kinf,c]=Pconstant(AR);
    ARtemp=max((round(AR*100))/100,0.25);
    sd(1)=interp1(ARTab,SD(:,1),ARtemp);
    sd(2)=interp1(ARTab,SD(:,2),ARtemp);
    sd(3)=interp1(ARTab,SD(:,3),ARtemp);
    Nuz(k,1:zend(k)) = 1./(sd(1)*(1-exp(sd(2)*zstar(k,1:zend(k)).^sd
        (3))));
    htc(k,1:zend(k)) = Nuz(k,1:zend(k)).*kfluid(k,1:zend(k))./(d(k)*1

```

```

        e-6);
Tw(k,1:zend(k)) = Tb(k,1:zend(k)) + fluxf*alpha./htc(k,1:zend(k))
;
Psat(k,1:zend(k)) = PofT(Tb(k,1:zend(k)))*1e2;      % Saturation
               pressure in kPa.
if k==1
    P(k,1) = Pstart;
else
    P(k,1) = P(k-1,zend(k-1));
end
for i = 2:zend(k)
    P(k,i) = P(k,i-1) + 0.25*(den(k,i)+den(k,i-1))*(U(k,i)+U(k,i-
        1))^2*...
        ( zeta(k,i-1)*(3.44/sqrt(zeta(k,i-1)) + (CfpRe + Kinf/(4*
            zeta(k,i-1)) - 3.44/sqrt(zeta(k,i-1)))/(1+c/zeta(k,i-
            1)^2)) ...
        - zeta(k,i)*(3.44/sqrt(zeta(k,i)) + (CfpRe + Kinf/(4*zeta
            (k,i)) - 3.44/sqrt(zeta(k,i)))/(1+c/zeta(k,i)^2)) );
    % Pressure in Pa
end
end

P = (P-P(ktrue,zend(ktrue))) + Pexit;

Pplot = P(1,1:zend(1));
Tbplot = Tb(1,1:zend(1));
zplot = z(1,1:zend(1));
Nuzplot = Nuz(1,1:zend(1));
Twplot = Tw(1,1:zend(1));
Psatplot = Psat(1,1:zend(1));
Prplot = Pr(1,1:zend(1));
h_plot = htc(1,1:zend(1));
Re_plot = Re(1,1:zend(1));
Uplot = U(1,1:zend(1));
for k = 2:ktrue
    Pplot = [Pplot P(k,1:zend(k))];
    Tbplot = [Tbplot Tb(k,1:zend(k))];
    zplot = [zplot z(k,1:zend(k))+zplot(end)];
    Nuzplot = [Nuzplot Nuz(k,1:zend(k))];
    Twplot = [Twplot Tw(k,1:zend(k))];
    Psatplot = [Psatplot Psat(k,1:zend(k))];
    Prplot = [Prplot Pr(k,1:zend(k))];
    h_plot = [h_plot htc(k,1:zend(k))];
    Re_plot = [Re_plot Re(k,1:zend(k))];
    Uplot = [Uplot U(k,1:zend(k))];

```

```

end

dP = Pplot(1)-Pplot(end);
Q = Mdot(1)*no/den(1);
Pow = dP*Q;
Twmax = max(Twplot);

ARs=min(hs,ws)/max(hs,ws); %set straight channel dimensions the
    same as the terminal channel in the fractal network
ds=4*(ws*hs)/(2*ws+2*hs); %calculation of hydraulic diameter of
    straight channel
dxs=ls/(round(ls/dx)); %updated node spacing
zs=[0:dxs:ls]; %new x vector for Pressure
    calculations
zsend = length(zs);

```

A.1.2 Optimization Algorithm

A.1.2.1 $T_{w,max}$ Constraint

```

function f_out = frac4(geometry)
% function f_out = frac4(geometry)
% Function that returns dP, Pow, Twmax, Qtot for fractal with given
    geometry
% specified by above beta, gamma, and wN
% designed to run with FMINCON in frac_opt.m

global flux_p Radius h m TWMAX a bmin d Qt nf n0_min n0_fix wm_max
global Twmax Qtot no
% disp(geometry)

beta = geometry(1);
gamma = geometry(2);
wN = geometry(3)*wm_max;

N = m;
R = Radius/1000; % [convert to mm for fractal_vp-2]

Qtot = Qt;

no = n0_fix;

count_break = 0;

```



```

if no < n0_min
    f_out = NaN;
    Twmax = NaN;
    Qtot = NaN;
else

    ktrue = N+1;
    w(ktrue) = wN;
    lengthf = R - no*(wN/1000)/(beta^N)/(2*pi);    % [mm]
    w_temp = wN./beta.^[N:-1:0];    % [microns]
    L_temp = lengthf*(gamma.^[0:N])/sum(gamma.^[0:N]);    % [mm]
    Area_wall_f = sum(2*L_temp.*(w_temp+h)/1e3.*(2.^[0:N]))*no; % [mm
        ^2]
    Area_plate = pi*R^2;    % [mm^2]
    load = flux.p*1e4*Area_plate*1e-6;    % [W]
    flux = flux.p*Area_plate/Area_wall_f;    % [W/cm^2]

    %    Qtot = 10;    % [ml/s]
    dx = 1;    % [microns]

    if lengthf <= 0
        f_out = NaN;
        Twmax = NaN;
        return
    end

    fractal_vp_2b
    count = 1;

    Twmax_old = Twmax;
    dP_old = dP;
    Pow_old = Pow;
    Qtot_old = Qtot;

    count = 1;
    TOL = 1e-4;
    QERR = 1;
    TTOL = 1e-4;    % relative tolerance on Twmax
    omega = 1.50;    % relaxation factor on Qtot

    while QERR > 0

        Qtot = ((Twmax-Tin+273)/(TWMAX-Tin+273))^omega*Qtot_old;
        %    disp(Qtot)
        ERR = 1;

```

```

dx = 1;

while ERR > TOL

    dx = dx/2;
    fractal_vp_2b
    count = count+1;
    ERR = max(abs([(Twmax-Twmax_old)/Twmax_old, (dP-dP_old)/
        dP_old, (Pow-Pow_old)/Pow_old]));
    Twmax_old = Twmax;
%    disp(Twmax)
    dP_old = dP;
    Pow_old = Pow;
    if count ≥ 50000
        disp('max iterations reached')
        count_break = 1;
        break
    end
end
Qtot_old = Qtot;
QERR = ((Twmax > TWMAX*(1+TTOL)) | (Twmax < (1-TTOL)*TWMAX));
%    disp([Qtot, ((Twmax-Tin+273)/(TWMAX-Tin+273))^omega])
if count_break
    break
end
end

if nf == 1
    f_out = dP;
else
    f_out = Pow;
end

end

Qt = Qtot;

% disp(f_out)

```

A.1.2.2 Gradient Based Search

```

% runs multiple iterations of frac_opt_v3_f
% for P = 0:1
%     for q = [10:5:30,50:25:100,150]%[200,300,500,1000]

```

```

%         frac_opt_v3_f([20,q,70,150],[0.5,1.0,0.5,1.5,50,150],[0.1,1
%         .5,2.5,1.5],P);
%         pack
%     end
% end
% for P = 0:1
%     %     if P == 0
%     %         Rrange = 50;
%     %     else
%     Rrange = 10:10:30;%40:10:50;
%     %     end
%     for R = Rrange
%         frac_opt_v3_f([R,200,70,150],[0.5,1.0,0.5,1.5,50,150],[0.1
%         ,1.5,2.5,1.5],P);
%         pack
%     end
% end
clear;clc;close all
global data_store max_store
tic
for P = 0
    T = 70;
    for q_r = 10;
        frac_opt_v3_f([20,q_r,T,150],[0.5,1.0,0.5,1.5,50,150],[0.1,1
        .5,2.5,1.5],P);
        pack
    end
end
toc
disp(data_store);
disp(max_store);
save(['results-on-' datestr(now,30) '.mat'],'data_store','max_store')
;

%for P = 0:1
%    %for b = [1.5,2,3,4,5]
%    % frac_opt_v3_f([20,200,70,150],[0.5,1.0,0.5,2.5,50,150],[0.1,
%    % b,b+2,1.5],P);
%    % pack
%    % end
%end
% for P = 0:1
%     for w = [10,20,50]
%         frac_opt_v3_f([20,200,70,150],[0.5,1.0,0.5,1.5,w,150],[0.1
%         ,1.5,2.5,1.5],P);

```

```

%         pack
%     end
% end
%for P = 0:1
%    for H = [75:25:175]
%        frac_opt_v3_f([20,200,70,H],[0.5,1.0,0.5,1.5,50,150],[0.1,1
%.5,2.5,1.5],P);
%    pack
%    end
%end
%for P = 0:1
%    for A = [0.05:0.05:0.25]
%        frac_opt_v3_f([20,200,70,150],[0.5,1.0,0.5,1.5,50,150],[A,1
%.5,2.5,1.5],P);
%    pack
%    end
%end

```

```

function f_output = frac_opt_v3_f(HeatSink,SearchRange,ConParam,
    dP_or_Pow)
% Master m-file for testing optimization programming in function form.
% Inputs are HeatSink, SearchRange, ConParam, and dP_or_Pow
% HeatSink is a 4-element vector of the form:
%    [Radius [mm], heat flux [W/cm^2], Max wall temp [deg C], channel
    depth
%    [microns]]
% SearchRange is a 6 element vector of the form:
%    [beta_min, beta_max, gamma_min, gamma_max, wm_min, wm_max]
%    wm_min and wm_max are in microns
% ConParam is a 4 element vector in the form:
%    [a, bmin, bmax, d]
% dP_or_Pow is a scalar
%    1 for pressure drop
%    anything else for pumping power

```

```

warning off MATLAB:divideByZero

```

```

global a bmin bmax d wm_max
global flux_p Radius h m TWMAX Qt nf n0_min n0_fix g0
global data_store max_store
global Twmax Qtot
tic

```

```

stor_dir = 'OutputRun02/';

```

```

nf = dP_or_Pow;           % 1 for pressure drop
                           % other for pumping power

Radius = HeatSink(1);    % [mm]
flux_p = HeatSink(2);    % [W/cm^2]
TWMAX  = HeatSink(3);    % [deg C]
h      = HeatSink(4);    % [microns]

beta_min = SearchRange(1);
beta_max = SearchRange(2);
gamma_min = SearchRange(3);
gamma_max = SearchRange(4);
wm_min    = SearchRange(5);
wm_max    = SearchRange(6);

a      = ConParam(1);
bmin   = ConParam(2);
bmax   = ConParam(3);
d      = ConParam(4);

Radius = Radius*1000; % [microns]
Qt = 100;           % [ml/s, initial guess]

n0_min = 3;

lower = [beta_min, gamma_min, wm_min/wm_max];
upper = [beta_max, gamma_max, 1]; % NOTE wm_max/wm_max = 1

g0 = [beta_min, gamma_max, wm_min/wm_max];

options = optimset('LargeScale','off','Display','off','TolFun',1e-6,'
    TolCon',1e-6,'DiffMinChange',10e-4,'TypicalX',[0.5,0.5,0.5]);%, '
    PlotFcns' ,{ @optimplotfval @optimplotstepsize });%, 'MaxFunEvals
    ',10^1);
m_min = ceil(-log(a*bmax)/log(2*beta_max));
m_min = max(1,m_min);
m_max = floor(log(2*pi*Radius/(bmin*n0_min*wm_min))/log(2));

max_store = [];

% m loop
wait_id = waitbar(0,'Starting . . . ');

k=0;

```

```

% n0_aa=[251;144;79;42;22;11;5;3]; % 1000 W/cm^2 case
% n0_aa=[5;3];%[251;144;79;42;22;11;5;3];
% n0_aa=[251;143;79;42;22;11;5;3]; %
% n0_aa=[168;139;79;42;22;11;5;3]; % 50
n0_aa=[84;101;78;42;22;11;6;3]; % 10
for m =2:9;%m_min:m_max
    k=k+1;
    data_store = inf*ones(1,9);

    % finds upper and lower limits on n0
    n0_limits

    no_n0 = n0_u-n0_l+1;
    bar_top = ['m = ',num2str(m),' (from ',num2str(m_min),' to ',
        num2str(m_max),')'];
    bar_bot = [' (from ',num2str(n0_l),' to ',num2str(n0_u),')'];
    waitbar(0,wait_id,strvcat(bar_top,['n_0 = 0',bar_bot]));
    n0_a=[];
    for n0_fix = n0_aa(k) %n0_l:n0_u

        %         disp(' ')
        %         disp([m,n0_fix])
        %         disp(' ')

        %         if ((Qt == Inf)|(Qt == NaN))
        Qt = 10;
        %         end

        opt_method='genetic';
        switch opt_method
            % should put in a wait bar
            %         waitbar((n0_fix-n0_l+1)/no_n0,[wait_id,strvcat(
            %         bar_top,['n_0 = ',num2str(n0_fix),bar_bot]));

        case 'gradient'
            gstart_finder; %% Does this find the valid
                range, or a starting point?
            [g, fval, exitflag, output] = fmincon(@frac4,gstart
                ,[],[],[],[],lower,upper,@constraints,options);

        case 'genetic'
            options=gaoptimset('PopulationSize',10,'Generations'
                ,10^2,...
                'PopulationSize',200,
                ...

```

```

        'StallGenLimit',10^2,'StallTimeLimit',1200,...
        'PlotFcns',@gaplotbestindiv,'MigrationInterval'
        ,1,'display','iter');
    [g, fval, exitflag] = ga(@frac4,3,[],[],[],[],lower,
        upper,@constraints,options);
    %         ga(...)
    % results comparison and storage
end
if fval < data_store(6)
    data_store(1:5) = [m,n0_fix,g];
    f = frac4(g);
    data_store(6:8) = [f,Twmax,Qtot]; % Storing the results
    data_store(9)=exitflag;
else
    data_store(9)=exitflag;
end

end

if ~isempty(data_store)
    max_store(m-m_min+1,1:9) = data_store;
else
    max_store(m-M_min+1,1:9) = [m,NaN,NaN,NaN,NaN,NaN,NaN,NaN,NaN
        ];
end

end

close(wait_id)
%
% % toc
% % return
%
% warning on MATLAB:divideByZero
%
% output_name = [stor_dir,'frac_opt_out2_',datestr(now,30)];
% output_name(find(output_name==' '))='_';
%
% diary([output_name,'_setup.txt'])
% if dP_or_Pow == 1
%     disp(['dP or Pow = dP']);
% else
%     disp(['dP or Pow = Pow']);
% end
% disp(['flux_p=      ',num2str(flux_p)]);

```

```

% disp(['radius=      ',num2str(Radius)]);
% disp(['TWMAX=      ',num2str(TWMAX)]);
% disp(['h=          ',num2str(h)]);
% disp(['a=          ',num2str(a)]);
% disp(['bmin=       ',num2str(bmin)]);
% disp(['bmax=       ',num2str(bmax)]);
% disp(['d=          ',num2str(d)]);
% disp(['beta_min=   ',num2str(beta_min)]);
% disp(['beta_max=   ',num2str(beta_max)]);
% disp(['gamma_min=  ',num2str(gamma_min)]);
% disp(['gamma_max=  ',num2str(gamma_max)]);
% disp(['wm_min=     ',num2str(wm_min)]);
% disp(['wm_max=     ',num2str(wm_max)]);
% diary off
%
% csvwrite([output_name,'_results.txt'],max_store);
% % save(['frac_opt_out ',datestr(now),'.mat'],'flux_p','Radius','
%       TWMAX','h','a','bmin','bmax','d',...
%       'beta_min','beta_max','gamma_min','gamma_max','wm_min','
%       wm_max','max_store');
%
% disp(' ')
% toc
%
% [min_val,min_loc] = min(max_store(:,6)); % Returns the minimum flow
%       power or pressure drop
% % and its location
% f_output = max_store(min_loc,:);

```

A.2 Two-Phase Heat Sink Optimization Code

A.2.1 Critical Heat Flux Estimation

```

function mdot=OneDCHF(qflux,m,n0,beta,gamma,wm,L,H)
% Find the mass flow rate at critical heat flux of a fractal-like
% branching
% channel network by solving a system of one-dimensional ODEs, first
% proposed by
if any(isnan([qflux,m,n0,beta,gamma,wm,L,H]))
    return
end
% [1]      R. Revellin and J. R. Thome. A theoretical model for
%       the prediction
%       of the critical heat flux in heated microchannels.

```



```

International
%      Journal of Heat and Mass Transfer, 51(5-6):pp. 1216-1225,
2008.

% The conservation of void fraction and quality for computing the
film
% thickness and velocities, respectively was adapted for square
channels
% from

% [2]      R. Revellin, J. R. Thome, A. Bejan, and J. Bonjour.
Constructal
%      tree-shaped microchannel networks for maximizing the
saturated critical
%      heat flux. International Journal of Thermal Sciences, 48(2)
:342 ? 352,
%      2009. Nano Micro Mini Channels and Computational Heat
Transfer.

% Run solver here to find the mass flow rate
% mdot = fsolve(@FINDMdot,...)
xt=linspace(0,L,100);
[Wa,tau,Lf,tau,tau]=fractalGeometry(m,beta,gamma,wm,xt);
% Properties
T = 100+273.15;
sigma = 58.9e-3; %53.6e-3;% N/meter
vl=Vlw(T);%0.842e-3;%
vv=Vvw(T);%0.0265;%
hfg=hvw(T)-hlw(T);%172e3;%
% mul = 0.01905e-2;%muliq_w(T,1); muv = muvap_w(T,1);
% muv = 12.4e-6;
g = 9.81;
Y = 0;
% Solving for the mass flow rate
guess1 = 0.52*n0/16*n0/60000*qflux^2/(12.3*100^2)^2;
opmdot = optimset('display','none','tolx',0.01*n0/60000,'tolfun',1e-
6);
% mdot=fmincon(@FINDMdot,guess1,[],[],[],[],eps,1000*n0/60000,[],
opmdot);
key = 0;
mdot = fzero(@FINDMdot,guess1,opmdot);

% key = 1;
% FINDMdot(mdot);

```

```

% mdot = abs(mdot);
% mdot = FINDMdot(5000*wm/beta^m*H*n0);
% Ma=(.3522)*n0/60000;
% resa=0*Ma;count = 0;
% for mm=Ma
%     count = count + 1;
%     resa(count)=FINDMdot(mm);
% end
% % figure,plot(Ma,resa);
% keyboard
function residualm=FINDMdot(M)
    if M≤0
        residualm = (M);
        return
    end
%     M = abs(M);
% Run ODE here to find r, uv, ul, pv and pl
% with the Kelvin-Helmholtz stability criteria at the exit
% Initial Conditions
Dhi = 2*(Wa(1)*H)/(Wa(1)+H);
y0(1) = M/(n0*Wa(1)*H)*v1;
y0(2) = M/(n0*Wa(1)*H)*v1;
y0(3) = Psw(T)-sigma*2/Dhi;
y0(4) = Psw(T);
Dref = min([Wa(1) H]);
y0(5) = (Dref - 0.1*Dref)/2;
Dhe = 2*(Wa(end)*H)/(Wa(end)+H);
%     VL=[];xlt=[];Vv=[];dd=[];
Ya = [];
for k=0:m

    % Solving the ODE
    %     dyo=y0';
    W = wm/beta^(m-k);
    opode=odeset('RelTol',1e-3,'AbsTol',1e-6);%,'NonNegative', [1 2]);
    [x1,Y] = ode113(@ODES,[0 Lf(k+1)],y0,opode); %#ok<ASGLU>
    d = Y(end,5); ul = Y(end,1); uv = Y(end,2);
    Al = 2*d*(W+H-2*d); Av = W*H-2*d*H-2*d*W+4*d^2;

    W = wm/beta^(m-(k+1));

    % Liquid Thickness
    dl = (beta^m*((H^2*beta^(2*m) + 8*beta*beta^(2*m)*d^2 +
        beta^2*beta^(2*k)*wm^2 - 4*H*beta*beta^(2*m)*d - 4*

```

```

        beta*beta^k*beta^m*d*wm)/beta^(2*m))^(1/2) + H*beta^m
        + beta*beta^k*wm)/(4*beta^m);
d2 = (H*beta^m - beta^m*((H^2*beta^(2*m) + 8*beta*beta
    ^ (2*m)*d^2 + beta^2*beta^(2*k)*wm^2 - 4*H*beta*beta
    ^ (2*m)*d - 4*beta*beta^k*beta^m*d*wm)/beta^(2*m))
    ^ (1/2) + beta*beta^k*wm)/(4*beta^m);
if any([(H-2*d1) (wm/beta^(m-k-1)-2*d1)] <0)
    y0(5) = d2;
else
    y0(5) = d1;
end
% Pressures
y0(3) = Y(end,3); y0(4) = Y(end,4);
% Velocities
Aln = 2*y0(5)*(W+H-2*y0(5)); Avn = W*H-2*y0(5)*H-2*y0(5)*W
    +4*y0(5)^2;
y0(1) = 1/2*ul*Al/Aln; y0(2) = 1/2*uv*Av/Avn;

% variables for plotting
%         if k>0
%             xlt = [xlt (xl'+xlt(end))];
%         else
%             xlt = xl';
%         end
%         VL = [VL Y(:,1)'];
%         Vv = [Vv Y(:,2)'];
%         dd = [dd Y(:,5)'];
if key==1
    Ya = [Ya;Y(:,5)];

    end
end
if key==1
    xtemp = linspace(0,L,length(Ya));
    figure,plot(xtemp,Ya);

end

%         W = interp1(xt,Wa,xlt);
%         Alf = 2*dd.*(W+H-2*dd); Avf = W*H-2*dd*H-2*dd.*W+4*dd.^2;
%         xl=xlt;
%         figure,plot(xl,W/2-dd);
%         hold on;plot(xl,W/2,'k-','linewidth',2.0);
%         legend('Vapor Core Height','Wall');
%         patch([xl fliplr(xl)], [W/2-dd fliplr(W/2)], 'b')

```

```

%       xlabel('x (m)');ylabel('\Delta (m)');
%       figure,plot(xl,[VL' Vv']);
%       xlabel('x (m)');ylabel('u (m/s)');
%       legend('u_l','u_v');
%       figure,plot(xl,[Alf' Avf']);
%       xlabel('x (m)');ylabel('A (m^2)');
%       figure,plot(xl,[Alf'.*VL'/vl Avf'.*Vv'/vv Alf'.*VL'/vl+Avf'
%       .*Vv'/vv]);
%       xlabel('x (m)');ylabel('mdot (kg/s)');
%       legend('Liquid','Vapor','Total');
%       figure,plot(xl,Avf'.*Vv'./vv./(Alf'.*VL'/vl+Avf'.*Vv'/vv));
%       xlabel('x (m)');ylabel('Quality');
%       keyboard
% residualmdot = (R-r) - Eqn. (19)
ind = (Y(:,1)≤0);
Y(ind,end)=-Y(ind,end);
uv=Y(end,2);ul=Y(end,1);
uv = abs(uv);ul = abs(ul);

residualm = Y(end,end)-0*0.15*Dhe/2*(uv/ul)^(-3/7)...
    *((1/vl-1/vv)*g*0.25*Dhe^2/sigma)^(-1/7);
residualm = real(residualm);
function dy=ODES(x,y) %#ok<INUSL>
    % Constants
    ul=y(1); uv=y(2);
    %Pl=y(3); Pv=y(4);
    d=y(5);
    % find the fractal geometry at a specific x level
    W=wm/beta^(m-k);
    Al = 2*d*(W+H-2*d); dAl = 2*W+2*H-8*d;
    Av = W*H-2*d*H-2*d*W+4*d^2; dAv = -2*H-2*W+8*d;
    %           Pi = 2*(W-2*d)+2*(H-2*d);
    %           Plw = 2*(W+H);

    % Determine flow conditions and shear stresses
    % Vapor
    %           Dhv = 4*Av/(Pi); Rev = uv*Dhv*vv/muv;
    %           if Rev<2300;Cf = 16/Rev;else Cf = 0.078*Rev
    %           ^(-1/4);end
    %           taulv = 1/2*Cf/vv*uv^2;
    %           tauvl = -taulv;
    % Liquid
    %           Dh1 = 4*Al/(Plw); Rel = ul*Dh1*vl/mul;
    %           if Rel<2300;Cf = 16/Rel;else Cf = 0.078*Rel
    %           ^(-1/4);end

```

```

%          taulw = 1/2*Cf/vl*ul^2;

% Define and quantify the values and their derivatives
% here
% Solving the nonlinear system of derivatives
A= [Al,0,ul*dAl;
    0,Av,uv*dAv;
    Al/vl,Av/vv,ul*dAl/vl+uv*dAv/vv];%0,0,1,-1,sigma*r^(-
    2)];%
b=[-qflux*2*(W+H)*vl/hfg;
    qflux*2*(W+H)*vv/hfg;
    0];

%if any(isnan(A(:))) || any(A(:)==inf)
%   keyboard
%end
dy=pinv(A,eps)*b;
dy=[dy(1:2);0;0;dy(3)];
end

end

end

function [W_a L_a L_f xf k_array]=fractalGeometry(m,beta,gamma,wm,x)
% Calculates the fractal channel widths, individual channel lengths,
% cumulative channel length at each bifurcation and generates an
% array of
% the same size as x with each variable
if max(size(x))==1
    L_ch=x;
    x=linspace(0,L_ch,m+1);
else
    L_ch=x(end);
end
dx=mean(diff(x));
wo=wm/beta^m;
% Running Sum Channel Array
Lo=L_ch/sum(gamma.^(0:m)); % Branch 0 length in meters
L_a=zeros(1,m+1);L_f=L_a;wa=L_a;
for k=0:m
    % Array of channel lengths (m)
    % Cumulative channel length

```

```

        L_a(k+1)=Lo*sum(gamma.^(0:k));
        % Individual channel lengths
        L_f(k+1)=Lo*gamma.^k;
        % and widths
        wa(k+1)=wo*beta.^k;
    end
    % Bifurcation Indices
    bif_ind=zeros(1,m+1);
    try
        for k=1:length(L_a)
            bif_ind(k)=find(x<=L_a(k),1,'last');
        end
    catch
        save(['fractalGeometryError' datestr(now,30) '.mat']);
    end
    % These are arrays used for finding flow conditions in each branch
    level
    W_a=wo*ones(size(x));
    k_array=0*x;
    for k=1:m
        % Array of k level numbers
        k_array(bif_ind(k)+1:bif_ind(k+1))=k;
        % Channel Width
        W_a(bif_ind(k)+1:bif_ind(k+1))=wo*beta^k; % meters
    end
    if length(x)==m+1
        W_a=wa;
    end
    % Generating the individual channel lengths for computing the
    entrance
    % effects
    bif_ind=[0 bif_ind];
    for k=1:m+1
        xf(bif_ind(k)+1:bif_ind(k+1))=linspace(dx,L_f(k)...
            ,length(bif_ind(k)+1:bif_ind(k+1)));
    end
end

```

A.2.2 Optimization

A.2.2.1 Main Algorithm

```

function run_2p(choose_flux)
warning off all

```

```

%choose_flux=100;
% wm (um)
% beta
% gamma
% H (um)
% L (m)
% m-1
% n0
% T_in
% q_in - wall heat flux (W/cm^2))
% m_dot - inlet mass flow rate (g/min)
global g0 T_a Tw_a P_a quality_a VF_a Pe% for finding the starting
point
study_type='Parametric';
% study_type='Gradient';
% study_type='Direct';
% study_type='Genetic';
%% Geometry and Flow
H=150; % microns
R=0.020; % meters

%% Constraints
a=0.1;bmin=1.5;bmax=2.5;d=1.5;
wm_min=50; wm_max=150;
wm = wm_min;
beta_min=0.5; beta_max=2.0;
gamma_min=0.51; gamma_max=2.0; n0_min=3;
g0=[wm_min beta_min gamma_min];
lower=[beta_min gamma_min];
upper=[beta_max gamma_max];
%% Execution
switch study_type
case 'Parametric'
%
Pe_range=101.3; % kPa-absolute
% Setting Ranges
Tin_range=25;
qflux_range=choose_flux;%[10 50 100 150];
%     m_dot_range=[5 25 50 75];
%     wm_range=100;%[10 50 70 100];
%     beta_range=[0.5 0.7 0.9 1.0 1.1 1.3 1.5];
%     gamma_range=[0.5 0.7 0.9 1.0 1.1 1.3 1.5];
for Pchf=0.7;%[0.7 0.8 0.9];
    for Pe=Pe_range;

```

```
for qfluxd=qflux_range
    for Tin=Tin_range
        %                                     for wm=wm_range
        %                                     for beta=beta_range;
        %                                     for gamma=gamma_range
        for m=9:-1:2
            [n0_l n0_u]=n0_limits; % Finds acceptable limits for n0
            for n0=n0_l:1:n0_u
                disp([m n0])
                %x=[wm beta gamma];
                try
                    tic,
                    options_start=optimset('display','final','
                        MaxFunEvals',1e4,'MaxIter',1e4);%, 'Algorithm'...
                    %, 'interior-point');
                    options=optimset('display','iter','MaxFunEvals',1e4
                        , 'MaxIter',1e4,'tolfun',1e-4,'tolx',1e-3);
                    % Optimizing...
                    g0=[beta_min,gamma_max];
                    %                                     x00=[0.5*(
                        beta_max+beta_min)...
                    %                                     ,0.5*(
                        gamma_min+gamma_max)];
                    x00=[1*(beta_max) ...
                        ,1*(gamma_min)];
                    % Finding Starting Location:

                    x0=fmincon(@gstart,x00,[],[],[],[]...
                        ,lower,upper,@(x) constraints(x,wm,H*1e-6,R,m,n0,a
                            ,bmin,bmax,d,wm_min)...
                        ,options_start);

                    [xf fval exitflag]=fmincon(@opt_run_2p,x0
                        ,[],[],[],[],lower,upper...
                        ,@(x) constraints(x,wm,H*1e-6,R,m,n0,a,bmin,bmax,d
                            ,wm_min)...
                        ,options);
                    endt=toc;
                    % recording the solution data:
                    sol_name=['Opt_CHF_solution_data_StabConst_'
                        num2str(qfluxd) '.txt'];% num2str(R) '-'...'
                    %,num2str(qfluxd) '-' num2str(wm) '-' num2str(Tin)
                        '.txt'];
                    % Recording the temperature, pressure, quality and
```



```

        void fraction
    % profiles
    % Presssure
    P_a=P_a(1)-P_a(end);
    %dlmwrite([sol_name(1:end-4) '_P.txt'],P_a,'-append
        ')
    % Bulk Temperature
    T_a=max(T_a);
    %dlmwrite([sol_name(1:end-4) '_T.txt'],T_a,'-append
        ')
    % Wall Temperature
    Tw_a=max(Tw_a);
    %dlmwrite([sol_name(1:end-4) '_Tw.txt'],Tw_a,'-
        append')
    % Qualtiy
    W_a=max(quality_a);
    %dlmwrite([sol_name(1:end-4) '_quality.txt'],
        quality_a,'-append')
    % Void Fraction
    VF_a=max(VF_a);
    %dlmwrite([sol_name(1:end-4) '_VF.txt'],VF_a,'-
        append')
    sol_data=[m n0 xf P_a T_a Tw_a W_a VF_a m_dot qflux
        wm Tin Pe endt fval exitflag];
    dlmwrite(sol_name,sol_data,'-append');
catch
    disp(lasterr);

    %save(['Run2pErrorAt' datestr(now,30) '.mat']);
    continue
end
end % n0 loop
end % m loop
%
%                                     end % gamma loop
%                                     end % beta loop
%                                     end % wm loop

    end % inlet Temperature Loop
    end % Wall heat flux range
    end % Exit Vacuum Condition Range
    end % percent of CHF loop
end % switch study type

function Pow=opt_run_2p(x)
    %wm=x(1);

```

```

beta=x(1);gamma=x(2);
%disp([beta gamma])
if any(isnan([beta,gamma,wm*1e-6]))
    %             beta = x0(1);
    %             gamma = x0(2);
    %             x = x0;
    %wm = x0(1);

    disp('NaN Values Appeared');
    % NaN values are not allowed
end
Z=R-n0*wm*1e-6/(beta^m*2*pi);
Z = 10e-6*round(Z/10e-6);
% check constraints
%             c_a=constraints(x,H,R,m,n0,a,bmin,bmax,d,wm_min);
%                                     disp(c_a);

% Area generation for wall heat flux:
Lo=Z/sum(gamma.^(0:m));wo=wm/beta^m*1e-6;
Area_s=sum(n0.*2.^(0:m).*Lo.*gamma.^(0:m)...
    .*2.*(wo.*beta.^(0:m)+H*1e-6));
Area_d=pi*(R^2-(R-Z)^2);
qflux=qfluxd*Area_d/Area_s;
% Determine mdot from CHF condition

% Qu and Mudawar
m_dot_a=CHFmdot(qflux*100^2,H*1e-6,Z,m,beta,gamma,wm*1e-6,PCHF);
    % g/min, initial guess for Twmax constraint
m_dot=max(m_dot_a.*n0.*2.^(0:m))*60000;

% Revellin and Thome

m_dot = OneDCHF(1/0.5*qflux*100^2,m,n0,beta,gamma,wm*1e-6,Z,H*1e-
    6);
m_dot = m_dot*60000; % g/min

[DP m_dot T_a Tw_a P_a quality_a VF_a]=run_2p_code(x,wm,H,Z,m,n0,
    Tin,qflux,m_dot,1,[],Pe);

% Skipping to the single-phase algorithm
%             if quality_a(end)==0 || isnan(quality_a(end))==1
%                 % Run Pence's single phase code
%                 % with constraint for Twmax0-m_dot iterations
%

```

```

%           [xs_del Tb_del Ts_del P_del]...
%           =fractal(n0,m,m_dot*1000/60000,wm,gamma,beta,
bmin,qflux,Z,H,Pe,Tin);
%           DP=P_a(1)-P_a(end);
%
%
%
%
%           end
Pow=m_dot/1000/60000*DP; % Watts
end % Objective function for optimization

function res=gstart(x)

res=sum(sqrt((g0-x).^2));

end % Function for finding a feasible starting loc.
function [n0_l n0_u]=n0_limits
wm_min=wm_min*1e-6;wm_max=wm_max*1e-6;
for kk = 1:m
k2 = kk-1;
gamma_lim(2*kk-1) = 2*pi*R*beta_max^m/(wm_min*((2*beta_min)^k2
*d-1)*sum(gamma_min.^[0:m])/sum(gamma_min.^[0:k2])+1));
gamma_lim(2*kk) = 2*pi*R*beta_max^m/(wm_min*((2*beta_min)^k2
*d-1)*sum(gamma_max.^[0:m])/sum(gamma_max.^[0:k2])+1));
end
n0_u = min(floor([2*pi*R*a*beta_max^m/wm_min, ...
2*pi*R/(2^m*bmin*wm_min), ...
gamma_lim]));
n0_l = max(ceil([n0_min,2*pi*R/(2^m*bmax*wm_max)]));

wm_min=wm_min*1e6;wm_max=wm_max*1e6;
end

end % end the run_2p function

```

A.2.2.2 Sub Function

```

function [DP m_dot T_a Tw_a P_a quality_a VF_a]=run_2p_code(xfunv,...
wm,hcin_f,Z_f,Levs_f,No_f,Tin_tf,qsize,msize,rec_array,TWMAX,Pe)
%#ok<INUSD,INUSL>
% This is the objective function for which optimization takes place

```

```

wct_f=wm;beta_f=xfunv(1);gama_f=xfunv(2); %#ok<NASGU>
% qsize=200;
% msize=9.71571862068966; %#ok<NASGU>

% Initializing necessary accounting arrays
Cipsize=2/3*ones(size(qsize)); %#ok<NASGU>
casen=(1:1:length(qsize)); %#ok<NASGU>
% finding mass flow rate to avoid CHF

% Executing Function
if rec_array==1

    Psize=100000*ones(size(qsize)); %#ok<NASGU>
    StartPdrop_f;

    m_dot=Gin*No_f*wct_f/beta_f^Levs_f*hc_in_f*(1e-6)^2*60000; %
        Assigning the output

    T_a=real(T);
    Tw_a=real(Tw);
    P_a=real(P);
    quality_a=real(W);
    VF_a=real(VF);
    DP=P(1)-P(end);

end % recording array statement

end % function

```

A.2.3 Constraints

```

function [c ceq]=constraints(x,...
    wm,H,R,m,n0,a,bmin,bmax,d,wm_min)
% With unit corrections
beta = x(1);
gamma = x(2);
wm = wm*1e-6;

ceq = [];

c(1) = n0*wm/beta^m - 2*pi*R*a;

```

```

c(2) = -1;%2*pi*R/(2^m*bmax) - n0*wm;
c(3) = -1;%n0*wm - 2*pi*R/(2^m*bmin);
c(4:4+m-1)=zeros(1,m);
for k = 0:m-1
    c(k+4) = d*(2*beta)^(k+1) - 1....
        - ((2*pi*beta^m*R)/(n0*wm) - 1)*sum(gamma.^(0:k))/sum(gamma.
            ^ (0:m)); %#ok<AGROW>
end

c(k+4+1)=wm_min*1e-6-wm/beta^m;

c(k+4+2) =real (gamma-(beta-0.1)^( (3*H+wm)/(H+wm) ));

end

```

A.2.4 Post Processing

```

clear;clc;%close all
% Opt_CHF_solution_data_betaall_100
% With stability criterion, set markers solid
mark = {'-ko', '-ks', '-kd', '-k^', '-kv', '-k<', '-k>', '-kx', '-k*'};
facecolor = 'none';
handvis = 'on';
count = 0;
%
% [35 60 80 125 200 300] % W/cm^2 betaall
%% Properties
sigma = 58.9e-3; % N/m
rho = 1/1.044e-3; % kg/m^3
mu = 279e-6; % Pa-s
Pr = 1.76;
R = 0.020; % meters
H = 150e-6; % meters
%% Plotting

for flux = [10 20 80 125 150 200] % W/cm^2 StabConst
    count = count+1;
    % load the data
    data = importdata(['Opt_CHF_solution_data_NoStabConst_' num2str(
        flux) '.txt']);
    % find the limits of the data over which to loop over
    % [m n0 xf P_a T_a Tw_a W_a VF_a m.dot qflux wm Tin Pe endt fval
        exitflag]

```

```

m = data(:,1); n0 = data(:,2); gamma = data(:,4); beta = data(:,3);
wm = data(:,12)*1e-6; P = data(:,5); T = data(:,6); Tw = data(:,7);
    quality = data(:,8);
VF = data(:,9); mdot = data(:,10); qflux = data(:,11); endt = data
    (:,15);
Pow = data(:,16); FLG = data(:,17);
tempPow = zeros(1,length(min(m):max(m)));tempQuality = tempPow;
tempBeta = tempPow; tempGamma = tempPow; tempWm = tempPow;
tempN0 = tempPow;tempflg = tempPow;tempmdot = tempPow;
ReMax = tempPow; ReMin = tempPow;WeMax = tempPow; WeMin = tempPow;
LsMax = tempPow; LsMin = tempPow;
ii=0;

for k = min(m):max(m)
    ii=ii+1;
    [tempPow(ii) ind] = min(Pow((m==k & FLG>0)));
    tempQuality(ii) = quality(ind);
    tempBeta(ii) = beta(ind);
    tempGamma(ii) = gamma(ind);
    tempWm(ii) = wm(ind);
    tempN0(ii) = n0(ind);
    tempflg(ii) = FLG(ind);
    tempmdot(ii) = mdot(ind);
    % Renolds, Webber and Lstar parameters
    Re = fractalReynolds(k,tempN0(ii),tempBeta(ii),tempWm(ii),H,...
        tempmdot(ii)/60000,rho,mu);
    ReMax(ii) = max(Re);ReMin(ii) = min(Re);
    We = fractalWebber(k,tempN0(ii),tempBeta(ii),tempWm(ii),H,...
        tempmdot(ii)/60000,rho,sigma);
    WeMax(ii) = max(We);WeMin(ii) = min(We);
    Ls = fractalxstar(k,tempN0(ii),tempBeta(ii),tempGamma(ii),...
        tempWm(ii),H,R,tempmdot(ii)/60000,rho,mu,Pr);
    LsMax(ii) = max(Ls);LsMin(ii) = min(Ls);
end
% Power
figure(1);set(1,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),tempPow,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
% % flags
% figure(2);set(1,'position',[682 184 860 420]);
% hold on;plot(min(m):max(m),tempflg,mark{count},'markersize
    ',10,'displayname',...
% ['q' '=' num2str(flux) ' W/cm^2']);

```

```

% n0
figure(6);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),tempN0,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);

% Quality
figure(2);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),tempQuality,mark{count},'markersize',10,
    'displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
figure(7);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),ReMax,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
figure(8);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),WeMax,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
figure(9);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),LsMax,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
figure(10);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),ReMin,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
figure(11);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),WeMin,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
figure(12);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),LsMin,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
continue
% Beta

```

```

figure(3);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),tempBeta,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
% Gamma
figure(4);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),tempGamma,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);
% wm
figure(5);set(gcf,'position',[682 184 860 420]);
hold on;plot(min(m):max(m),tempWm,mark{count},'markersize',10,'
    displayname',...
    ['q' '=' num2str(flux) ' W/cm^2'],'markerfacecolor',facecolor,
    'handlevisibility',handvis);

end
for i=[1:6 7:12];figure(i);h=legend('toggle');set(h,'Location','
    EastOutside');end
figure(1);set(gca,'yscale','log')
ylabel('Flow Power (Watts)');
xlabel('m');
% figure(2);set(gca,'yscale','linear')
% ylabel('Convergence Flag');
% xlabel('m');
figure(7);
ylabel('Re-{max}');
xlabel('m');
set(gca,'yscale','log')
figure(8);
ylabel('We-{max}');
xlabel('m');
set(gca,'yscale','log')
figure(9);
ylabel('Ls-{max}');
set(gca,'yscale','log')

xlabel('m');
figure(10);
ylabel('Re-{min}');
xlabel('m');
set(gca,'yscale','log')
figure(11);

```



```

ylabel('We_{min}');
xlabel('m');
set(gca,'yscale','log')
figure(12);
ylabel('Ls_{min}');
xlabel('m');
set(gca,'yscale','log')

figure(2);
ylabel('Quality');
xlabel('m');
figure(3);
ylabel('\beta');
xlabel('m');
figure(4);
ylabel('\gamma');
xlabel('m');
figure(5);
ylabel('w_m (\mu m)');
xlabel('m');
figure(6);
ylabel('n_0');
xlabel('m');

function Fratio = fractalEvapMom(m,n0,beta,gamma,wm,R,H,qflux,mdot,
    rho1,rhov,hfg)

Ltot = R-n0*wm/beta^m/2/pi;
L0 = Ltot/sum(gamma.^(0:m));
L = zeros(1,m+1);
for k = 0:m
    i = k+1;
    L(i) = L0*gamma.^k;
end
Adisk = pi*(R^2-(R-Ltot)^2);

Awall = n0*L(1)*(sum(2.^((0:m)+1).*gamma.^(0:m).*(H+wm./beta.^(m-(0:m)
)))));
qfluxwall = qflux*Adisk/Awall;

Fratio = L*0;
for k = 0:m
    i = k+1;
    Fratio(i) = rho1/rhov*(n0*2^k/mdot*qfluxwall*L(i)*2*(wm/beta^(m-k)+

```

```
H)/hfg)^2;
```

```
end
```

```
end
```

```
function Re = fractalReynolds(m,n0,beta,wm,H,mdot,rho,mu)
% Determines the Reynolds number = VDh/nu at each k level
```

```
Re = zeros(1,m+1);
```

```
for i = 1:m+1
```

```
    k = i-1;
```

```
    wk = wm/beta^(m-k);
```

```
    Dh = 2*wk*H/(wk+H);
```

```
    V = mdot/(rho*n0*2^k*wk*H);
```

```
    Re(i) = rho*V*Dh/mu;
```

```
end
```

```
end
```

```
function We = fractalWebber(m,n0,beta,wm,H,mdot,rho,sigma)
```

```
% Determines the Webber number rhoV^2Dh/sigma at each k level
```

```
We = zeros(1,m+1);
```

```
for i = 1:m+1
```

```
    k = i-1;
```

```
    wk = wm/beta^(m-k);
```

```
    Dh = 2*wk*H/(wk+H);
```

```
    V = mdot/(rho*n0*2^k*wk*H);
```

```
    We(i) = rho*V^2*Dh/sigma;
```

```
end
```

```
function xs = fractalxstar(m,n0,beta,gamma,wm,H,R,mdot,rho,mu,Pr)
```

```
% determines the x^* = x/(DhRePr) value at each k level
```

```
Re = zeros(1,m+1);
```

```
xs = Re;
```

```
Ltot = R-n0*wm/beta^m/(2*pi);
```

```
sg = sum(gamma.^(0:m));
```

```
for i = 1:m+1
```

```
    k = i-1;
```

```
    wk = wm/beta^(m-k);
```

```
    Dh = 2*wk*H/(wk+H);
```

```
    V = mdot/(rho*n0*2^k*wk*H);
```

```
    Re(i) = rho*V*Dh/mu;
```

```
    L = Ltot/sg*gamma^k;
```

```

    xs(i) = L/(Dh*Re(i)*Pr);
end

```

```

end

```

A.3 Single-Phase Heat Exchanger Model Code

A.3.1 Model Function File

```

function [epsilon,DTlm,UA,NTU,FlowPower_h,FlowPower_c,Eout,Cstar,
    Thend,Tcend]=...
    fractalHXSolver(xvariable,m_dot_c,m_dot_h,Thi,Tci,m,n0,ploton,L)

% Solves for the temperature and pressure in one fractal tree
% clear;%clc
% clc;close all

beta = xvariable(1);
gamma = xvariable(2);
wm = xvariable(3);
% close all;
% set(0,'DefaultFigureWindowStyle','normal')
%% PROPERTIES
m_dot_c=m_dot_c/n0;
m_dot_h=m_dot_h/n0;
% m_dot_c=30/60000;%3.6704e-06;%0.30/60000/16; % kg/s
% m_dot_h=-3/60000;%3.6704e-06;%0.30/60000/16; % kg/s
Th_in=Thi;%115;%400-273.15; % Degrees C
Tc_in=Tci;%115-75;%300-273.15; % Degrees C
k_mat=400;%15.1; % W/mK
Pexit=101300; % Pascals
% Geometry
R=L;
H=250e-6; % meters
W=wm;%50e-6; % terminal channel width, meters
dx=100e-6;
t=717e-6;%500e-6; % Meters, the thickness of hot-cold interface
% n0=16;
mm=m;
% gamma=sqrt(2);
% beta=1/sqrt(2);
wo=W/beta^mm;wm=W;
if mm>0

```

```

    L_ch=R-n0*W/beta^mm/2/pi; % meters, length of the tree
elseif mm==0
    L_ch = R*2;
end

% round L_ch to the nearest dx
L_ch=dx*round(L_ch/dx);
% Controls:
% for the optimum
% to x^(-1/3)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% OUTPUT

% Tc_new = 1 x N array: Final bulk cold side temperature
% Th_new = 1 x N array: Final bulk hot side temperature
% q_in = Global energy balance (hot side)
% q_out = Global energy balance (cold side)
% run_time = Total time to attain solution to residual criterion
% m = number of iterations to attain solution
% x = 1 x N array: x locations
% hot_res = m x 2 array with residuals and their locations
% cold_res = m x 2 array with residuals and their locations

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Saturated liquid water at 300K
rho_c=997.56; % kg/m^3
Cp_c=4.1817e3; % J/kg K
mu_c=888.7e-6; % Ns/m^2
nu_c=mu_c/rho_c; % m^2/s
k_wat=620.3e-3; % W/mK
Pr_c=Cp_c*mu_c/k_wat; % Prandtl Number
% % Air at 300K
% rho_c=1.184; % kg/m^3
% Cp_c=1003; % J/kg K
% mu_c=1.855e-5; % Ns/m^2
% nu_c=mu_c/rho_c; % m^2/s
% k_wat=0.02603; % W/mK
% Pr_c=Cp_c*mu_c/k_wat; % Prandtl Number

N=round((L_ch/dx)+1);
u_m_c=m_dot_c/(rho_c*H*W/beta^mm); % meters/sec.

% % Engine Oil at 300K
% rho_h=884; % kg/m^3
% Cp_h=1.909e3; % J/kg K

```

```

% mu_h=48.6e-2; % Ns/m^2
% nu_h=mu_h/rho_h; % m^2/s
% k_oil=145e-3; % W/mK
% Pr_h=Cp_h*mu_h/k_oil; % Prandtl Number
% % Paratherm(R)
% rho_h=720; % kg/m^3
% Cp_h=1.909e3; % J/kg K
% mu_h=0.00072; % Ns/m^2
% nu_h=mu_h/rho_h; % m^2/s
% k_oil=116e-3; % W/mK
% Pr_h=Cp_h*mu_h/k_oil; % Prandtl Number
% % Paratherm NF
rho_h=863; % kg/m^3
Cp_h=2.009e3; % J/kg K
nu_h=11/1000^2; %mu_h/rho_h; % m^2/s
mu_h=nu_h*rho_h; % Ns/m^2
k_oil=104e-3; % W/mK
Pr_h=Cp_h*mu_h/k_oil; % Prandtl Number
% Saturated liquid water
% rho_h=997.56; % kg/m^3
% Cp_h=4.1817e3; % J/kg K
% mu_h=888.7e-6; % Ns/m^2
% nu_h=mu_c/rho_c; % m^2/s
% k_oil=620.3e-3; % W/mK
% Pr_h=Cp_h*mu_h/k_oil; % Prandtl Number
% % Air at 300K
% rho_h=1.184; % kg/m^3
% Cp_h=1003; % J/kg K
% mu_h=1.855e-5; % Ns/m^2
% nu_h=mu_c/rho_c; % m^2/s
% k_oil=0.02603; % W/mK
% Dhh=4*(W*H)/(2*(W+H)); % m, hydraulic diameter
% Pr_h=Cp_c*mu_c/k_wat; % Prandtl Number
kf_c=k_wat; kf_h=k_oil;
% k_l=k_wat; k_h=k_oil;
u_m_h=abs(m_dot_h/(rho_h*H*W/beta^mm));
% Initializing the streamwise distance array
x=0:dx:L_ch;
% building the matrix
%% SOLID GEOMETRY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Axial heat transfer area
if mm>0
    Axs=([0 x(1:end-1)+dx/2])+wo*n0/2/pi)*2*pi/n0*t; % m^2
else

```

```

    Axs=wm*2*t*ones(1,N);
end
%% COLD SIDE GEOMETRY AND FLOW
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Running Sum Channel Array
Lo=L_ch/sum(gamma.^(0:mm)); % Branch 0 length in meters
L_a=zeros(1,mm+1); L_f=L_a;
for k=0:mm
    % Array of channel lengths (m)
    L_a(k+1)=(1*Lo*sum(gamma.^(0:k)));
    L_f(k+1)=(1*Lo*gamma.^k);
end
% Bifurcation Indices
bif_ind=zeros(1,mm+1);

for k=1:length(L_a)
    bif_ind(k)=find(x<=L_a(k),1,'last');
end
bif_ind(end) = length(x); % ensure no roundoff errors
% These are arrays used for finding flow conditions in each branch
level
V_a_c=0*x; m_dot_c_a=0*x;
W_a=wo*ones(size(x));
V_a_c(1:bif_ind(1))=u_m_c;
m_dot_c_a(1:bif_ind(1))=m_dot_c;
k_array=0*x;
for k=1:mm
    k_array(bif_ind(k)+1:bif_ind(k+1))=k;
    % Velocity
    V_a_c(bif_ind(k)+1:bif_ind(k+1))=u_m_c/(2^k*beta^k); % m/s
    % Mass Flux
    m_dot_c_a(bif_ind(k)+1:bif_ind(k+1))=m_dot_c/(2^k); % kg/s/m^2
    % Channel Width
    W_a(bif_ind(k)+1:bif_ind(k+1))=wo*beta^k; % meters
end
% Fin width
r=(x+wm*n0/2/pi/beta^mm);
if mm>0
    Wf=(r*2*pi-W_a.*n0.*2.^k_array)./(n0*2.^k_array)+1e-6;
else
    Wf=wm;
end
Dh=4*W_a*H./(2*(W_a+H));
Re_c=V_a_c.*Dh/nu_c;

```

```

% Generating the individual channel lengths for computing the
    entrance
% effects
bif_ind=[0 bif_ind];
for k=1:mm+1
    xf(bif_ind(k)+1:bif_ind(k+1))=linspace(dx,L_f(k)...
        ,length(bif_ind(k)+1:bif_ind(k+1)));
end
% Generating Nusselt Number Correlations and thermal resistance
Nu_c=0.0949*(xf./(Dh.*Re_c*Pr_c)).^(-0.7875); h_c=Nu_c./Dh*k_wat;
mL=sqrt(2*dx*Nu_c./Dh*k_wat./(k_mat*Wf*dx)).*(H+Wf/2);
eta_c=tanh(mL)./mL;
R_c=1./(eta_c.*h_c.*2.*(W_a+H)*dx)+t./(2*(k_mat*W_a*dx));
%% HOT SIDE GEOMETRY AND FLOW
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Running Sum Channel Array
Lo=L_ch/sum(gamma.^(0:mm)); % Branch 0 length in meters
L_a=zeros(1,mm+1); L_f=L_a;
for k=0:mm
    % Array of channel lengths (m)
    L_a(k+1)=(1*Lo*sum(gamma.^(0:k)));
    L_f(k+1)=(1*Lo*gamma.^k);
end
x=0:dx:L_ch;
% Bifurcation Indices
bif_ind=zeros(1,mm+1);
for k=1:length(L_a)
    bif_ind(k)=find(x≤L_a(k),1,'last');
end
bif_ind(end) = length(x); % ensure no roundoff errors
% These are arrays used for finding flow conditions in each branch
    level
V_a_h=0*x;m_dot_h_a=0*x;
W_a=wo*ones(size(x));
V_a_h(1:bif_ind(1))=u_m_h;
m_dot_h_a(1:bif_ind(1))=m_dot_h;
k_array=0*x;
for k=1:mm
    k_array(bif_ind(k)+1:bif_ind(k+1))=k;
    % Velocity
    V_a_h(bif_ind(k)+1:bif_ind(k+1))=u_m_h/(2^k*beta^k); % m/s
    % Mass Flux
    m_dot_h_a(bif_ind(k)+1:bif_ind(k+1))=m_dot_h/(2^k); % kg/s/m^2
    % Channel Width

```

```

        W_a(bif_ind(k)+1:bif_ind(k+1))=wo*beta^k; % meters
    end
    Dh=4*W_a*H./(2*(W_a+H));
    Re_h=V_a_h.*Dh/nu_h;
    % Generating the individual channel lengths for computing the
    % entrance
    % effects
    bif_ind=[0 bif_ind];
    for k=1:mm+1
        switch sign(m_dot_h)
            case 1
                xfh(bif_ind(k)+1:bif_ind(k+1))=linspace(dx,L_f(k)...
                    ,length(bif_ind(k)+1:bif_ind(k+1)));
            case -1
                xfh(bif_ind(k)+1:bif_ind(k+1))=fliplr(linspace(dx,L_f(k)
                    ...
                    ,length(bif_ind(k)+1:bif_ind(k+1))));
        end
    end
    % Generating Nusselt Number Correlations and thermal resistances
    Nu_h=0.0949*(xfh./(Dh.*Re_h*Pr_h)).^(-0.7875);h_h=Nu_h./Dh*k_oil;
    mL=sqrt(2*dx*Nu_h./Dh*k_oil./(k_mat*Wf*dx)).*(H+Wf/2);
    eta_h=tanh(mL)./mL;
    R_h=1./(eta_h.*h_h*2.*(W_a+H)*dx)+t./(2*(k_mat*W_a*dx));

    %% PRESSURE DROP AND FLOW POWER CALCULATIONS
    AR=zeros(1,N);Ph=AR+Pexit;Pc=Ph;
    % Generate the coefficients for each x-location
    switch sign(m_dot_h)
        case 1
            Narray=N-1:-1:1;
        case -1
            Narray=2:N;
    end
    for i=Narray
        % Aspect Ratio
        AR(i)=min([H/W_a(i) W_a(i)/H]);
        [CfpRe Kinf CC]=Pconstant(AR(i));
        % Hot Side
        psi=xfh(i)/(Re_h(i)*Dh(i));
        Cfapp=(3.44/sqrt(psi)+...
            (CfpRe+(Kinf/4/psi)-3.44/sqrt(psi))...
            /(1+(CC/psi^2)))/Re_h(i);
        Ph(i)=Ph(i+sign(m_dot_h))+1/2*rho_h*(V_a_h(i))^2*Cfapp*4*dx/(Dh(i)
    );

```



```

end
% Cold Side
for i=N-1:-1:1
    [CfpRe Kinf CC]=Pconstant(AR(i));
    psi=xf(i)/(Re_c(i)*Dh(i));
    Cfapp=(3.44/sqrt(psi)+...
        (CfpRe+(Kinf/4/psi)-3.44/sqrt(psi))...
        /(1+(CC/psi^2)))/Re_c(i);
    Pc(i)=Pc(i+1)+1/2*rho_c*(V_a_c(i))^2*Cfapp*4*dx/(Dh(i));
end
% Flow Power
FlowPower_h=n0*(Ph(1)-Ph(end))*m_dot_h/rho_h;
FlowPower_c=n0*(Pc(1)-Pc(end))*m_dot_c/rho_c;
%% COLD SIDE HEAT TRANSFER MATRIX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % Cold matrix
% Cdiag= [1 -m_dot_c_a(2:end)*Cp_c-[2*kf_c*W_a(2:end-1) kf_c*W_a
    (end)]*H/dx-[1./R_c(2:end-1) 0]];
% Cdiagupper= [0 kf_c*W_a(2:end-1)*H/dx];
% Cdiaglower= m_dot_c_a(2:end)*Cp_c+kf_c*W_a(2:end)*H/dx;
%
% C=diag(Cdiaglower,-1)+diag(Cdiag,0)+diag(Cdiagupper,1);
% % Cold to solid matrix
% C_S=diag([0 1./R_c(2:end-1) 0],0);
switch sign(m_dot_c);
    case -1
        Tc_out=Tc_in;
        Tc_in=0;
        InletCoef=Cp_c*m_dot_c_a(2)-kf_c*W_a(1)*H/dx;
        InnerCoef=Cp_c*m_dot_c_a(3:end)-2*kf_c*W_a(2:end-1)*H/dx-1./
            R_c(2:end-1);
        OutletCoef=1;
        CCdiaglower=[kf_c*W_a(2:end-1)*H/dx 0];
        CCdiagupper=-Cp_c*m_dot_c_a(2:end)+kf_c*W_a(1:end-1)*H/dx;
    case 1
        Tc_out=0;
        InletCoef=1;
        InnerCoef=-Cp_c*m_dot_c_a(2:end-1)-2*kf_c*W_a(2:end-1)*H/dx-1
            ./R_c(2:end-1);
        OutletCoef=-Cp_c*m_dot_c_a(end)-kf_c*W_a(end)*H/dx;
        CCdiagupper=[0 kf_c*W_a(2:end-1)*H/dx];
        CCdiaglower=Cp_c*m_dot_c_a(2:end)+kf_c*W_a(2:end)*H/dx;
end
CCdiag=[InletCoef InnerCoef OutletCoef];

```

```

C=diag(CCdiaglower,-1)+diag(CCdiag,0)+diag(CCdiagupper,1);
C_S=diag([0 1./R_c(2:end-1) 0],0);

%% SOLID HEAT TRANSFER MATRIX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Sdiag=      [-k_mat*Axs(1)/dx -k_mat*Axs(1:end-2)/dx-k_mat*Axs(2:end-
1)/dx-1./R_c(2:end-1)-1./R_h(2:end-1) -k_mat*Axs(end-1)/dx];
Sdiagupper= k_mat*Axs(1:end-1)/dx;
Sdiaglower= k_mat*Axs(1:end-1)/dx;
S=diag(Sdiag,0)+diag(Sdiagupper,1)+diag(Sdiaglower,-1);
% Hot to solid matrix
HH_S=diag([0 1./R_h(2:end-1) 0],0);

%% HOT SIDE HEAT TRANSFER MATRIX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch sign(m_dot_h);
case -1
    Th_out=Th_in;
    Th_in=0;
    InletCoef=Cp_h*m_dot_h_a(2)-kf_h*W_a(1)*H/dx;
    InnerCoef=Cp_h*m_dot_h_a(3:end)-2*kf_h*W_a(2:end-1)*H/dx-1./
        R_h(2:end-1);
    OutletCoef=1;
    HHdiaglower=[kf_h*W_a(2:end-1)*H/dx 0];
    HHdiagupper=-Cp_h*m_dot_h_a(2:end)+kf_h*W_a(1:end-1)*H/dx;
case 1
    Th_out=0;
    InletCoef=1;
    InnerCoef=-Cp_h*m_dot_h_a(2:end-1)-2*kf_h*W_a(2:end-1)*H/dx-1
        ./R_h(2:end-1);
    OutletCoef=-Cp_h*m_dot_h_a(end)-kf_h*W_a(end)*H/dx;
    HHdiagupper=[0 kf_h*W_a(2:end-1)*H/dx];
    HHdiaglower=Cp_h*m_dot_h_a(2:end)+kf_h*W_a(2:end)*H/dx;
end
HHdiag=[InletCoef InnerCoef OutletCoef];
HH=diag(HHdiaglower,-1)+diag(HHdiag,0)+diag(HHdiagupper,1);

%% SOLVING THE HEAT TRANSFER MATRIX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Z=zeros(N);
A= [C C_S Z;C_S S HH_S;Z HH_S HH];
b=[Tc_in zeros(1,2*N-2) Tc_out Th_in zeros(1,N-2) Th_out]';
T=A\b; % Solving
Tc=T(1:N);Ts=T(N+1:2*N);Th=T(2*N+1:end);

```

```

%% OUTPUT PARAMETERS
% Ein=n0*m_dot_c*Cp_c*(Tc(end)-Tc(1));%+n0*kf_c*W_a(1)*H*(Tc(2)-Tc(1))
% /dx;
Tcend=max(Tc);
Thend=min(Th);
Eout=n0*m_dot_h*Cp_h*(Th(1)-Th(end));%-n0*kf_h*W_a(1)*H*(Th(2)-Th(1))
% /dx+...
% Eout=Ein;
% n0*kf_h*W_a(end)/dx*(Th(end)-Th(end-1))*2^mm;
% figure(4);hold on;plot(Ein,Eout,'ko');hold off;
Cstar=min(abs([m_dot_h*Cp_h/m_dot_c/Cp_c m_dot_c*Cp_c/m_dot_h/Cp_h]))
;
DTlm=((Th(1)-Tc(1))-(Th(end)-Tc(end)))/log((Th(1)-Tc(1))/(Th(end)-Tc(
end)));
UA=Eout/DTlm;
Cmin=n0*min(abs([m_dot_c*Cp_c m_dot_h*Cp_h]));
NTU=UA/Cmin;
epsilon=max([abs(Tc(1)-Tc(end)) abs(Th(1)-Th(end))])/(Thi-Tci);
% epsilon=Eout/Cmin/(Thi-Tci);
endt=toc;
if ploton==1;
    mark='-';
    figure(10),
    cla
    plot(x,Tc,['b' mark],'displayname',...
        ['Gridsize = ' num2str(1e6*dx,'%3.0f') ' \mu m,\newlineSolution
        time = ' num2str(endt,'%2.2f') ' sec']...
        , 'linewidth',1.5);hold on;
    plot(x,Ts,['k' mark],'handlevisibility','off','linewidth',1.5);
    plot(x,Th,['r' mark],'handlevisibility','off','linewidth',1.5);
    plot(x,Tc(1)+40+x*0,'k--','linewidth',1.5);
    hold off
    xlabel('$x\rightarrow$(meters)$','interpreter','latex');
    ylabel('$T\rightarrow(\circ C)$','interpreter','latex');
    drawnow;pause(0.1);
    Dprintpdf(gcf,['/Users/Doug/Desktop/OSUResearch/
        OptimizationAlgorithm/fractalHXmodel/ParametricStudy/mdot '
        num2str(m_dot_h*60000) '.pdf']);
% title(['m = ' num2str(mm)]);
% text(x(round(N/4)),Thi,{'\epsilon = ' num2str(epsilon,'%2.2f')
% },...
% ['C_c = ' num2str(m_dot_c*n0*Cp_c)],...
% ['C_h = ' num2str(m_dot_h*n0*Cp_h)]});
end

```

end

A.3.2 Parametric Search

```
function CounterFlowParametricFractalHX(beta,gamma,n0,m,wmo,nameid)
warning off all
% [epsilon,DTlm,UA,NTU,FlowPower_h,FlowPower_c,Eout,Cstar,endt,Tcend
]=...
% fractalHXsolver(xvariable,m_dot_c,m_dot_h,Thi,Tci,m,n0,ploton,L
);
%% Controls, Constants and Limits
n0o = n0;
ploton = 1;
beta_min = 0.5; beta_max = 2.0;
gamma_min = 0.5; gamma_max = 2.0;
wm_min = 50e-6; wm_max = 150e-6;
Thio = 150;
Thi_a = 60:40:200;
Tci = 20;
flag2 = NaN;
Ro = 0.020; % meters, disk radius
Ra = 0.01:0.01:0.080;
%wmo = 100e-6;
wm_a = (wm_min:10e-6:wm_max);
DT_a = 40;% (10:10:70); % degrees C temperature rise on the cold side
m_dot_c_a = 30/60000;%[1 10 50 100 150 250 500]/60000; % kg/s cold
side mass flow rate
d = 1.5; bmin = 1.5; bmax = 2.5;
a = 0.1; n0_min = 3;
options_mdot = optimset('display','none','tolx',eps,'tolfun',eps...
,'maxiter',10000,'maxfuneval',10000);
%% Vary wm
R = Ro; Thi = Thio;
savename = ['CounterFlowParametric_3geom_' nameid '.txt'];
for m_dot_c = m_dot_c_a
for DT = DT_a

for wm = wm_a % The independent variable loop
disp(wm);

n0 = floor(n0o*wmo/wm);
if n0<=0
```

```

        n0=1;
    end
    try
        tic
        xfinal = [beta gamma wm];
        if 1;%all(constraints(xfinal,R,m,n0,a,bmin,bmax,d,wm)≤0)
            % Running Model
            OptEpsilon_p = FindFractalHXmdot(xfinal);
            keyboard
            %
            [xfinal,OptEpsilon_p,flag2]=fmincon(
                @FindFractalHXmdot,x0,...
            %
                [],[],[],[],[beta_min,gamma_min,
                wm_min],[beta_max,gamma_max,wm_max]...
            %
                ,@(x) constraints(x,R,m,n0,a,bmin,bmax
                ,d,wm_min),opt_options);
            endt = toc;

            savedata = [m,n0,xfinal,R,epsilon,DTlm,UA,NTU,PowH,PowC,
                Eout,Cstar,Tce,Thi,The,m_dot_h,OptEpsilon_p,flag1,flag2,
                endt];
            dlmwrite(savename,savedata,'-append');
        end
        % Printing the error and continuing
    catch %#ok<CTCH>
        disp(lasterr); %#ok<LERR>
    end
end % the variable loop

end % the DT Loop
end % the m_dot_c loop
%% Vary R
Thi = Thio; wm = wmo;
for m_dot_c = m_dot_c_a
    for DT = DT_a

        for R = R_a % The independent variable loop
            disp(R);

            n0 = floor(n0o*R/Ro);
            if n0≤0
                n0=1;
            end
            try
                tic
                xfinal = [beta gamma wm];

```

```

if 1;%all (constraints (xfinal,R,m,n0,a,bmin,bmax,d,wm) ≤ 0)
% Running Model
OptEpsilon_p = FindFractalHXmdot (xfinal);

% [xfinal,OptEpsilon_p,flag2]=fmincon(
% @FindFractalHXmdot,x0,...
% [],[],[],[],[beta_min,gamma_min,
% wm_min],[beta_max,gamma_max,wm_max]...
% ,(x) constraints (x,R,m,n0,a,bmin,bmax
% ,d,wm_min),opt_options);
endt = toc;

savedata = [m,n0,xfinal,R,epsilon,DTlm,UA,NTU,PowH,PowC,
Eout,Cstar,Tce,Thi,The,m_dot_h,OptEpsilon_p,flag1,flag2,
endt];
dlmwrite(savename,savedata,'-append');
end
% Printing the error and continuing
catch %#ok<CTCH>
disp(lasterr); %#ok<LERR>
end
end % the variable loop

end % the DT Loop
end % the m_dot_c loop
%% Vary Thi
R = Ro; wm = wmo;
for m_dot_c = m_dot_c_a
for DT = DT_a

for Thi = Thi_a % The independent variable loop
disp(Thi);
try
tic
xfinal = [beta gamma wm];
if 1;%all (constraints (xfinal,R,m,n0,a,bmin,bmax,d,wm) ≤ 0)
% Running Model
OptEpsilon_p = FindFractalHXmdot (xfinal);

% [xfinal,OptEpsilon_p,flag2]=fmincon(
% @FindFractalHXmdot,x0,...
% [],[],[],[],[beta_min,gamma_min,
% wm_min],[beta_max,gamma_max,wm_max]...
% ,(x) constraints (x,R,m,n0,a,bmin,bmax
% ,d,wm_min),opt_options);

```

```

        endt = toc;

        savedata = [m,n0,xfinal,R,epsilon,DTlm,UA,NTU,PowH,PowC,
                    Eout,Cstar,Tce,Thi,The,m_dot_h,OptEpsilon_p,flag1,flag2,
                    endt];
        dlmwrite(savename,savedata,'-append');
    end
    % Printing the error and continuing
catch %#ok<CTCH>
    disp(lasterr); %#ok<LERR>
end
end % the variable loop

end % the DT Loop
end % the m_dot_c loop
%% Functions

% Function to find the mass flow rate
function Epsilon_p_inverse = FindFractalHXmdot(xvar)
    [m_dot_h,~,~,flag1]=lsqnonlin(@fractalsect,-m_dot_c,-inf,0,
        options_mdot);
    [epsilon,DTlm,UA,NTU,PowH,PowC,Eout,Cstar,The,Tce]...
        =fractalHXSolver...
        (xvar,m_dot_c,m_dot_h,Thi,Tci,m,n0,ploton,R);
    Epsilon_p_inverse = (PowH+PowC)/Eout;
function y=fractalsect(x)

    %           [epsilon(i,j),DTlm(i,j),UA(i,j),NTU(i,j),PowH(i,j),
    %           PowC(i,j),Eout(i,j),Cstar(i,j),endt,y]...
    [~,~,~,~,~,~,~,~,~,y]...
        =fractalHXSolver...
        (xvar,m_dot_c,x,Thi,Tci,m,n0,ploton,R);
    y=y-(Tci+DT);

end
end

% Function to find the n0 limits
function [n0_l n0_u]=n0_limits %#ok<DEFNU>
    gamma_lim = zeros(1,2*m);
    for kk = 1:m
        k2 = kk-1;
        gamma_lim(2*kk-1) = 2*pi*R*beta_max^m/(wm_min*(((2*beta_min)^k2
            *d-1)*sum(gamma_min.^(0:m))/sum(gamma_min.^(0:k2))+1));
        gamma_lim(2*kk) = 2*pi*R*beta_max^m/(wm_min*(((2*beta_min)^k2

```

```

        *d-1)*sum(gamma_max.^(0:m))/sum(gamma_max.^(0:k2))+1));
end
n0_u = min(floor([2*pi*R*a*beta_max^m/wm_min, ...
    2*pi*R/(2^m*bmin*wm_min), ...
    gamma_lim]));
n0_l = max(ceil([n0_min, 2*pi*R/(2^m*bmax*wm_max)]));

end

end

```

A.3.3 Post Processing

```

% PP4Geom.m
clear;clc;close all

%% Conditions and formatting
GEOM = [1/sqrt(2) sqrt(2) 16 4 100e-6
    1/sqrt(2) 1/sqrt(2) 16 4 100e-6
    0.75 1.18 79 4 50e-6
    1/sqrt(2) sqrt(2) 400 0 50e-6];
GeomName = {'frac1','frac2','frac3','straight'};
mdoth = [1 10 100 250 500]; % g/min
flowconf = {'CoFlow','CounterFlow'}; %
mdoth_leg = {'k-','k--','k-.','k-o','k-*'};
chilpos = [
    0.1848    0.1163    0.2799    0.2157
    0.6251    0.4423    0.2799    0.2157
    0.1848    0.4423    0.2799    0.2157
    0.6251    0.7696    0.2799    0.2157
    0.1848    0.7696    0.2799    0.2157]; % positions for the subplots
DTa = [10 20 30 50 60];
DTa = (10:10:70); % degrees C temperature rise on the cold side
m_dot_c_a = [1 10 50 100 150 250 500]/60000; % kg/s cold side mass
    flow rate
Ra = 0.01:0.01:0.08;
wmind = 5; % index for wm
yind = 18; % epsilon 7; % e_p 19 % mdoth 18
ylabelstr = '$\dot{m}_h$ (g/min)'; rotation = 90;
% ylabelstr = '$\varepsilon$'; rotation = 0;
% ylabelstr = '$\epsilon_p$'; rotation = 0;
% varname = 'epsilon';
% varname = 'ep';
% ylimshift = 4; % ep, eps

```



```

ylimshift = 0; % mdot
varname = 'mdot';
% setylim = [0.505 1]; % epsilon
% setylim = [1e2 1e6]; % e_p
setylim = [1e-1 500]; % mdoth
setyscale = 'log';
for j = 1:length(flowconf);
    for i = 1:size(GEOM,1)
        if i≠3
            continue
        end
        beta = GEOM(i,1); gamma = GEOM(i,2); n0 = GEOM(i,3); m = GEOM(i,4); % wmo = GEOM(i,5);
        newloadname = ['New' flowconf{j} 'Parametric_3geom_' GeomName{i} '.txt'];
        data = importdata(newloadname);

        % [m,n0,beta,gamma,wm,R,epsilon,DTlm,UA,NTU,PowH,PowC,Eout,Cstar,
            Tce,Thi,The,m_dot_h,OptEpsilon_p,flag1,flag2,endt,mdotc,DTc];
        % 1 2 3 4 5 6 7 8 9 10 11 12 13 14
            15 16 17 18 19 20 21 22 23 24%

        %% Filtering out bad data
        data = data(data(:,20)>0,:); % non-convergence
        data = data(data(:,13)>0,:); % bad energy
        data = data(data(:,7)≤1,:); % bad epsilon

        %% Plotting for Varying R
        varind = 6; % Radius
        wmo = GEOM(i,5);
        Tho = 150;

        iind = (data(:,wmind)==wmo & data(:,16)==Tho); %
        ydata = data(iind,yind);
        mdata = data(iind,23);
        DTdata = data(iind,24);
        vardata = data(iind,varind);
        % filter the x and y data, for best cases corresponding to a
            specific
        % disk radius
        figure(i+(j-1)*4); set(i+(j-1)*4, 'Position', [440 -17 637
            795], 'name'...
            , [varname GeomName{i} flowconf{j} 'R']);

        count = 0;

```

```

for DT = DTa
    if DT≠20 %&& DT≠60
        continue
    end
    count = count+1;
    ax=subplot(3,2,count);
    count2 = 0;
    for mdot = mdoth
        count2 = count2+1;
        ind = (mdata==mdot & DTdata==DT);
        tempdata=[vardata(ind) ydata(ind)];
        tempdata=sortrows(tempdata,1);
        % Plotting each subplot
        if yind==19 % e_p
            tempdata(:,2) = 1./tempdata(:,2);
        elseif yind==18 % mdoth
            tempdata(:,2) = 60000*abs(tempdata(:,2));
        end
        plot(tempdata(:,1)*1e3,tempdata(:,2),mdoth_leg{count2}...
            , 'linewidth',1.0, 'displayname', ['$\dot{m}_c = ' num2str(
                mdot, '%3.0f') 'g/min$']);hold on
        %
        plot(tempdata(:,1)*1e3, (1./tempdata(:,3))/0.5e6,
            mdoth_leg{count2}...
            , 'linewidth',1.0, 'handlevisibility','off');hold on
    end
    set(ax, 'ylim', setylim, 'xlim', [10 80], 'yscale', setyscale);
    yla = ylabel(ylabelstr, 'interpreter', 'latex', 'rotation',
        rotation);
    set(yla, 'position', get(yla, 'position')-[ylimshift 0 0]);
    xlabel('$R$ (mm)', 'interpreter', 'latex')
    % "a" begins with 97
    titlestring = ['$(' char(96+count) ') \Delta T_c = ' num2str(DT
        ) '^{\circ}C$'];

    %
    tt=title(titlestring, 'interpreter', 'latex');
    %
    set(tt, 'units', 'normalized');
    %
    postt = get(tt, 'Position');
    set(gca, 'position', chilpos(6-count, :));
    posax = get(gca, 'Position');
    an = annotation('textbox', [posax(1), posax(2)-0.0955, posax(3), 0
        .0332], ...
        'string', titlestring, 'interpreter', 'latex', '
        HorizontalAlignment', 'center'...
        , 'EdgeColor', 'none');
end

```

```

% Fixing the legend
plleg = legend('toggle','location','BestOutside');
set(plleg,'interpreter','latex','Position',[0.6001    0.0833    0
        .2834    0.1981])

%% Plotting for Varying wm
varind = 5; % w_m
varo = GEOM(i,5);
Ro = 0.020;

iind = (data(:,6)==Ro & data(:,16)==Tho); %
ydata = data(iind,yind);
mdata = data(iind,23);
DTdata = data(iind,24);
vardata = data(iind,varind);

% Plot the independent variable (parametrically varied) versus
the
% performance variable of interest
figure(i+(j-1)*4+8);set(i+(j-1)*4+8,'Position',[440    -17    637
        795],'name'...
,[varname GeomName{i} flowconf{j} 'Wm']);

count = 0;
for DT = DTa
    if DT≠20 %&& DT≠60
        continue
    end
    count = count+1;
    ax=subplot(3,2,count);
    count2 = 0;
    for mdot = mdoth
        count2 = count2+1;
        ind = (mdata==mdot & DTdata==DT); % inner indices
        tempdata=[vardata(ind) ydata(ind)];
        tempdata=sortrows(tempdata,1);
        % Plotting each subplot
        if yind==19
            tempdata(:,2) = 1./tempdata(:,2);
        elseif yind==18 % mdoth
            tempdata(:,2) = 60000*abs(tempdata(:,2));
        end
        plot(tempdata(:,1)*1e6,tempdata(:,2),mdoth_leg{count2}...

```

```

        , 'linewidth', 1.0, 'displayname', ['$\dot{m}_{-c} = ' num2str(
            mdot, '%3.0f') 'g/min$']); hold on
    %       plot(tempdata(:,1)*1e6, (1./tempdata(:,3))/0.5e6,
    mdoth_leg{count2}...
    %       , 'linewidth', 1.0, 'handlevisibility', 'off'); hold on

end
set(ax, 'ylim', setylim, 'xlim', [50 150], 'yscale', setyscale);
ylabel(ylabelstr, 'interpreter', 'latex', 'rotation', rotation)
xlabel('$w_m (\mu m)$', 'interpreter', 'latex')
% "a" begins with 97
titlestring = ['$\left(' char(96+count) '\right) \Delta T_c = '
    num2str(DT) '^{\circ}C$'];
set(gca, 'position', chilpos(6-count, :));
posax = get(gca, 'Position');
an = annotation('textbox', [posax(1), posax(2)-0.0955, posax(3), 0
    .0332], ...
    'string', titlestring, 'interpreter', 'latex', '
    HorizontalAlignment', 'center'...
    , 'EdgeColor', 'none');
end

% Fixing the legend
plleg = legend('toggle', 'location', 'BestOutside');
set(plleg, 'interpreter', 'latex', 'Position', [0.6001    0.0833    0
    .2834    0.1981])
%% Plotting for Varying Tin
end
end

```

Appendix B – Fractal-like Heat Exchanger Parametric Analysis Figures

Table B.1: Geometric parameters of the three heat exchanger geometries with $T_{c,in} = 20^\circ C$, $T_{h,in} = 150^\circ C$, $H = 250 \mu m$, $t = 717 \mu m$ and $R = 20 mm$

Geometry	n_0	m	$w_m(\mu m)$	β	γ
G1	16	4	100	0.707	1.41
G2	16	4	100	0.707	0.707
G3	79	4	50	0.750	1.18
G4	400	0	50	-	-

Table B.1 outlines the various geometries used to generate the following plots in this appendix. It should be mentioned that G4 in Table B.1 is not fractal-like, rather it is an array of parallel, straight channels arranged in a square geometry, not a disk shape. The results of this Appendix were generated by solving for the hot side mass flow rate that satisfied a fixed cold side mass flow rate and cold side temperature drop.

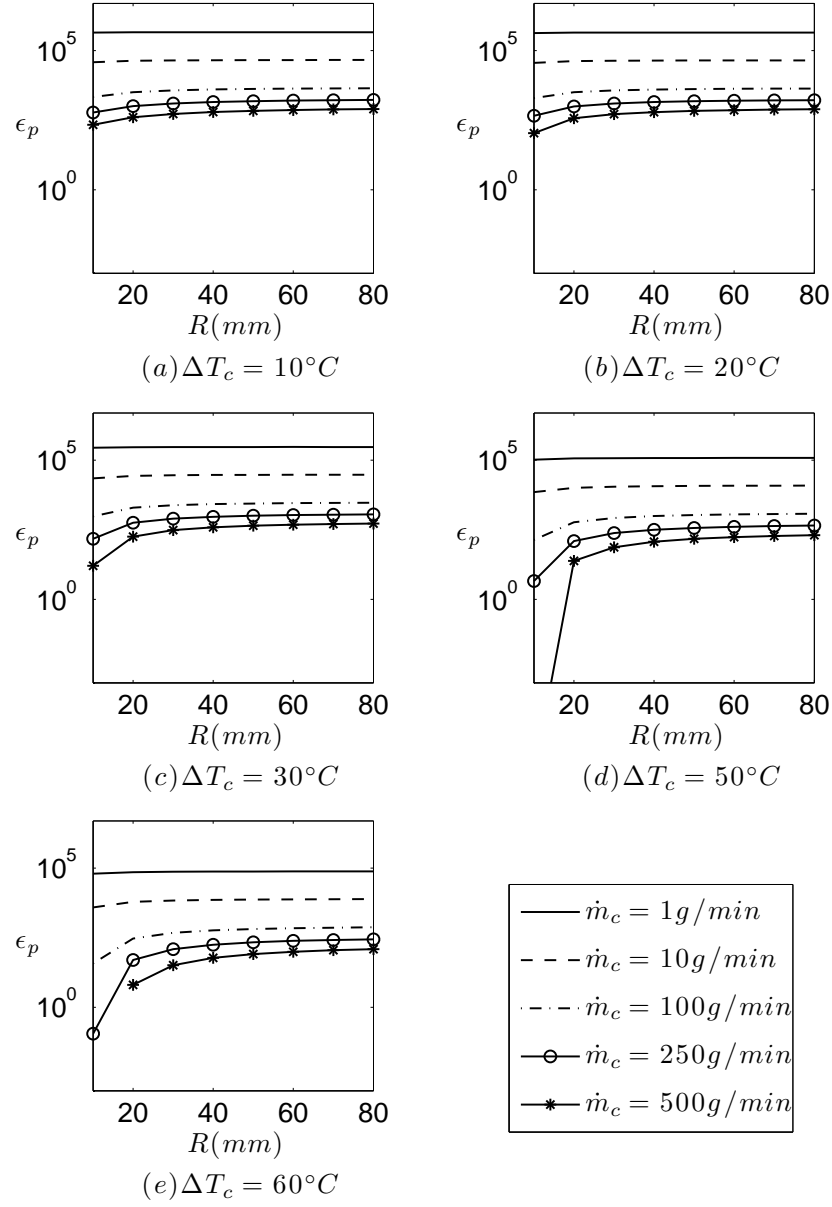


Figure B.1: Benefit to cost ratio of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

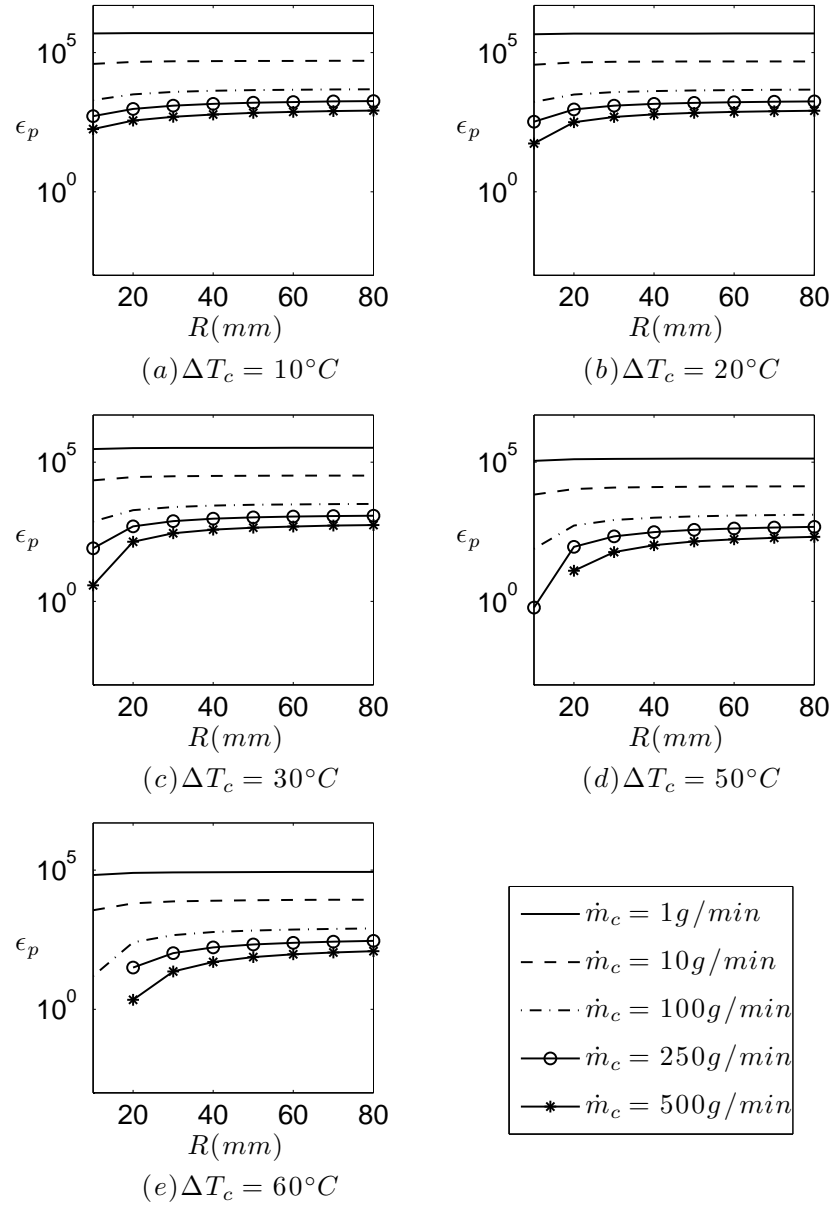


Figure B.2: Benefit to cost ratio of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

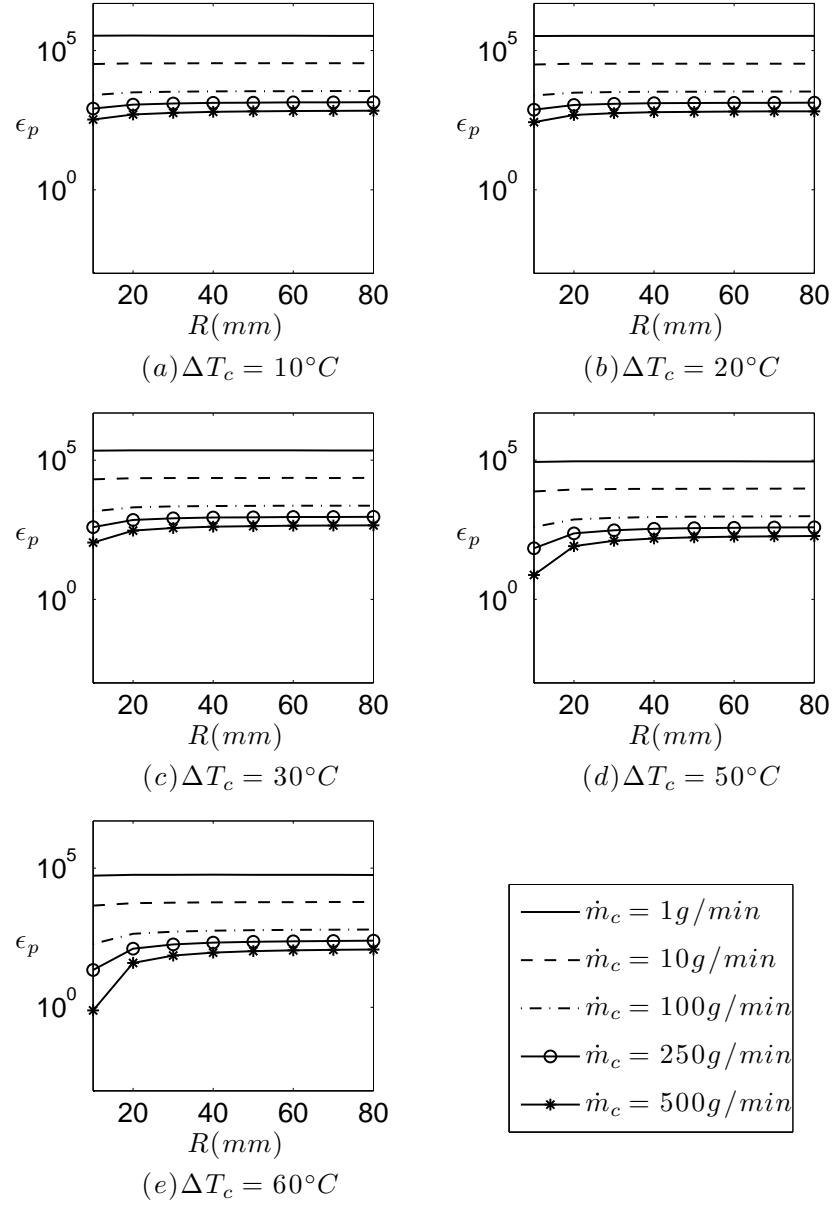


Figure B.3: Benefit to cost ratio of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

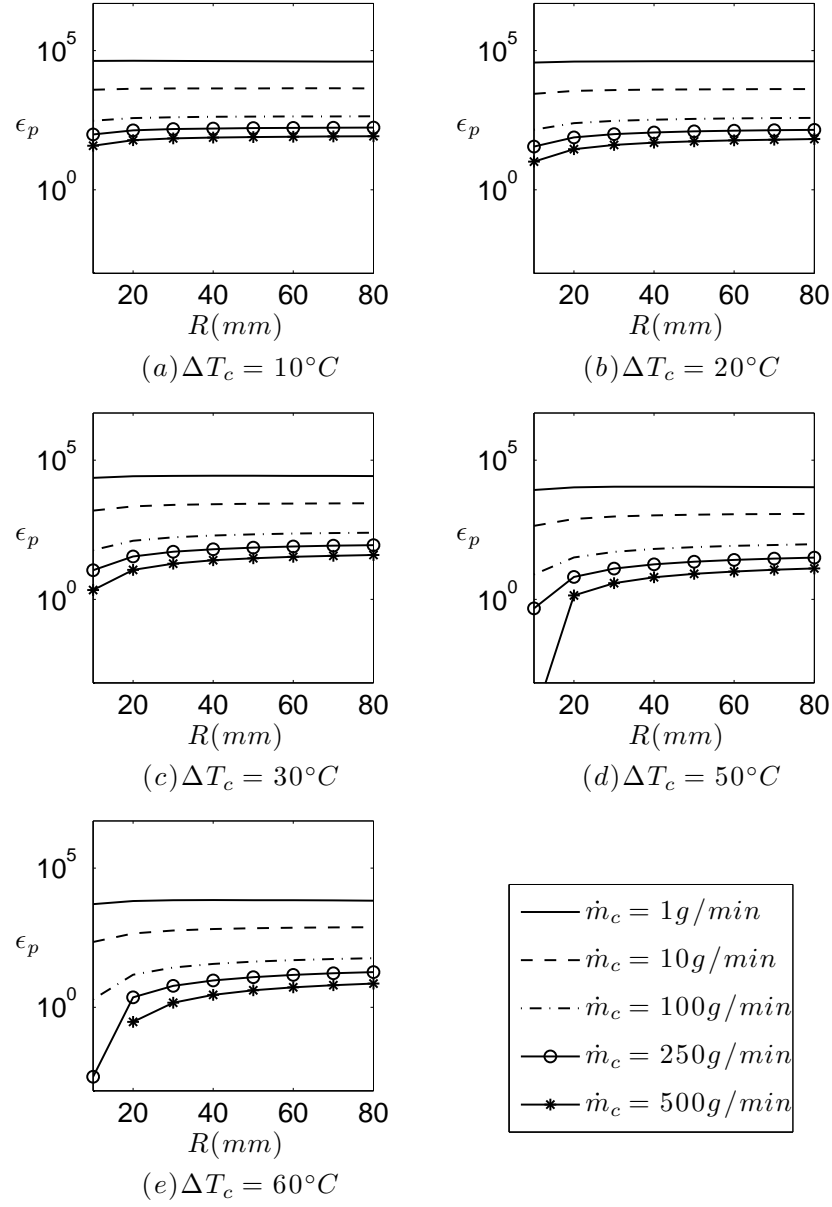


Figure B.4: Benefit to cost ratio of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

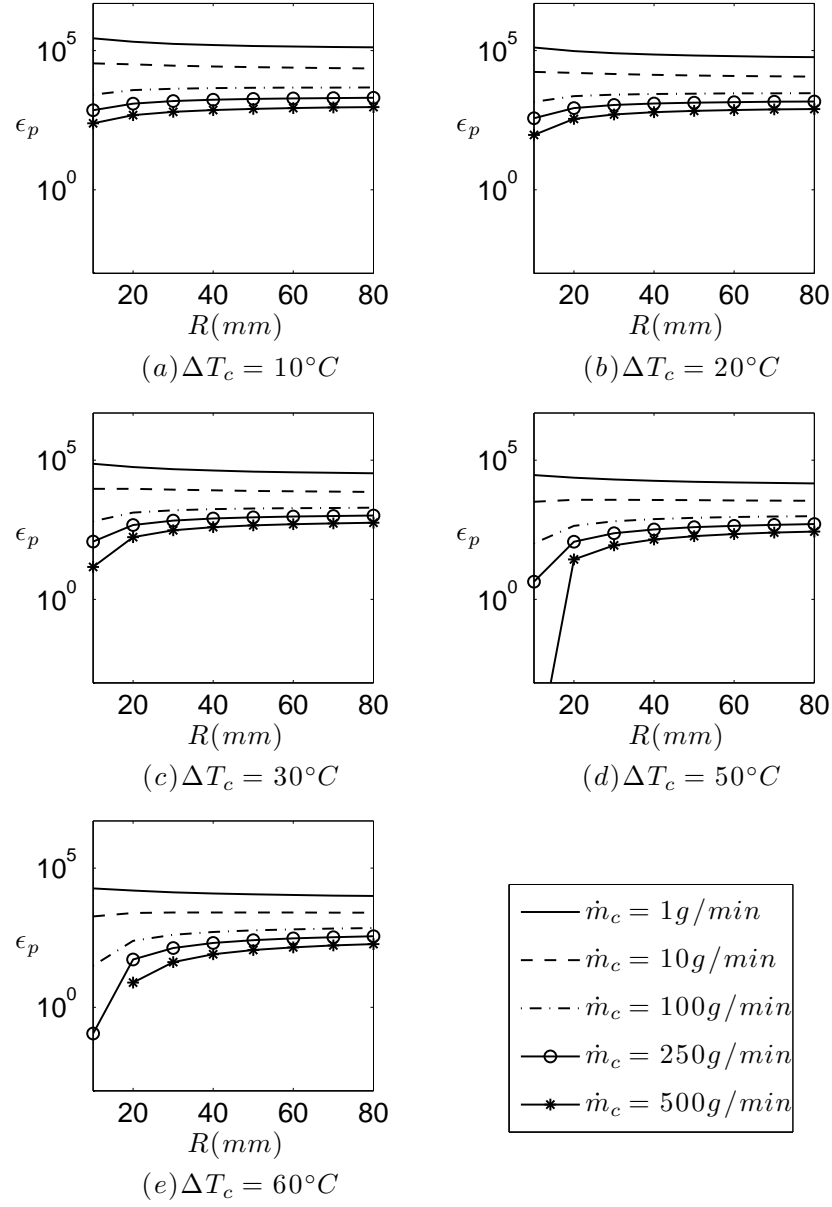


Figure B.5: Benefit to cost ratio of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

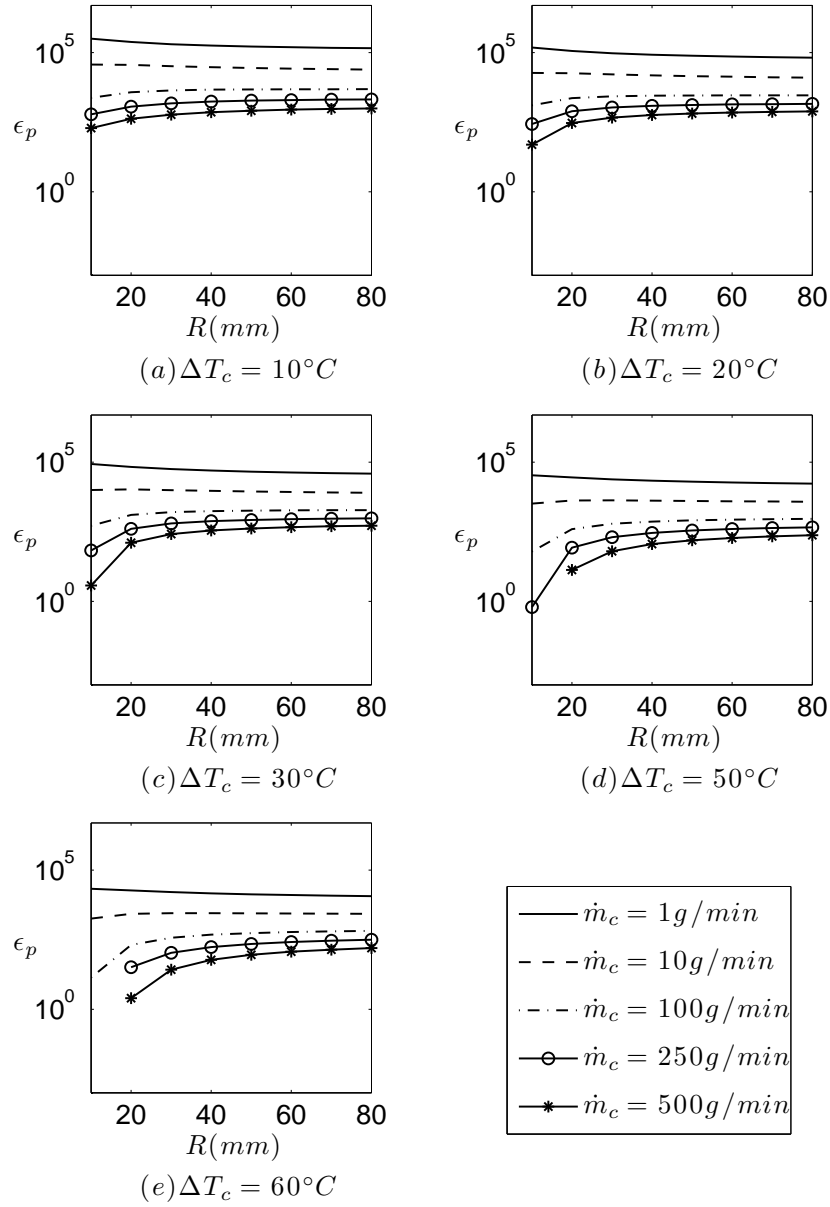


Figure B.6: Benefit to cost ratio of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

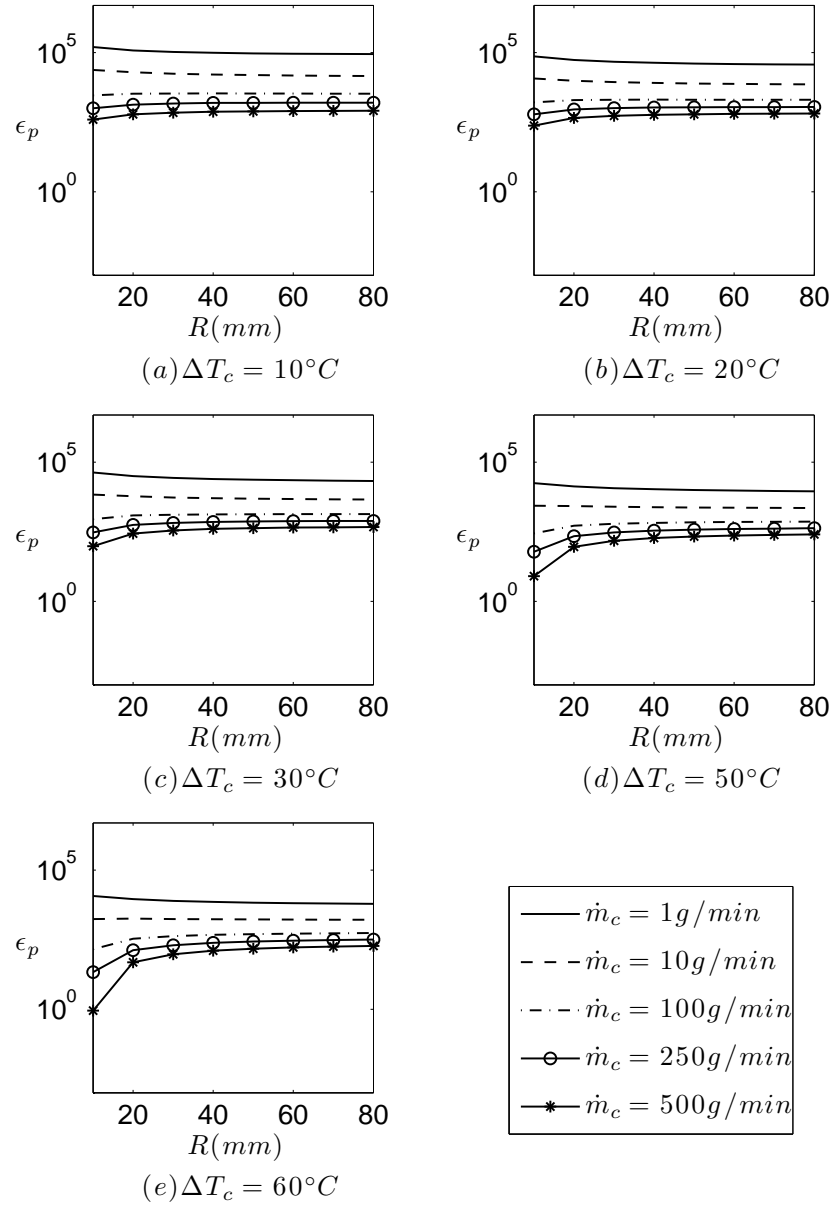


Figure B.7: Benefit to cost ratio of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

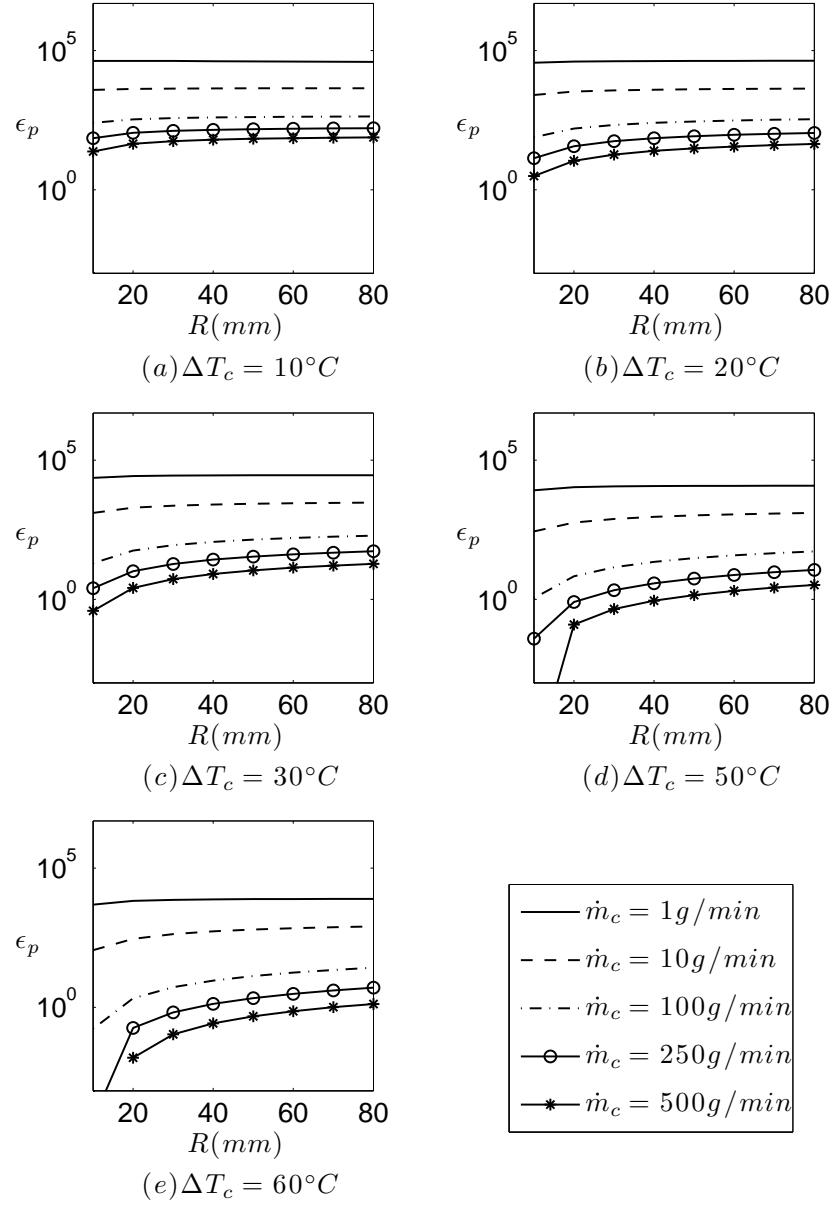


Figure B.8: Benefit to cost ratio of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

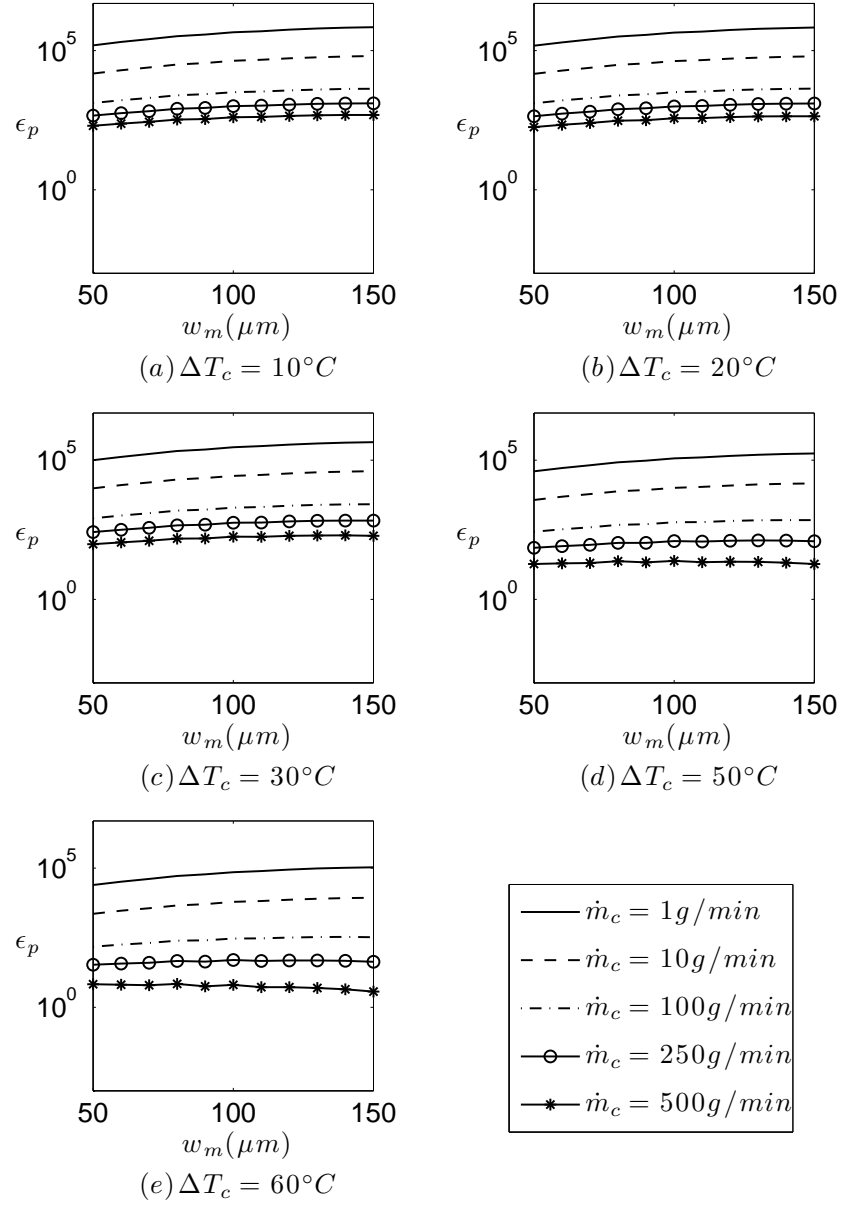


Figure B.9: Benefit to cost ratio of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

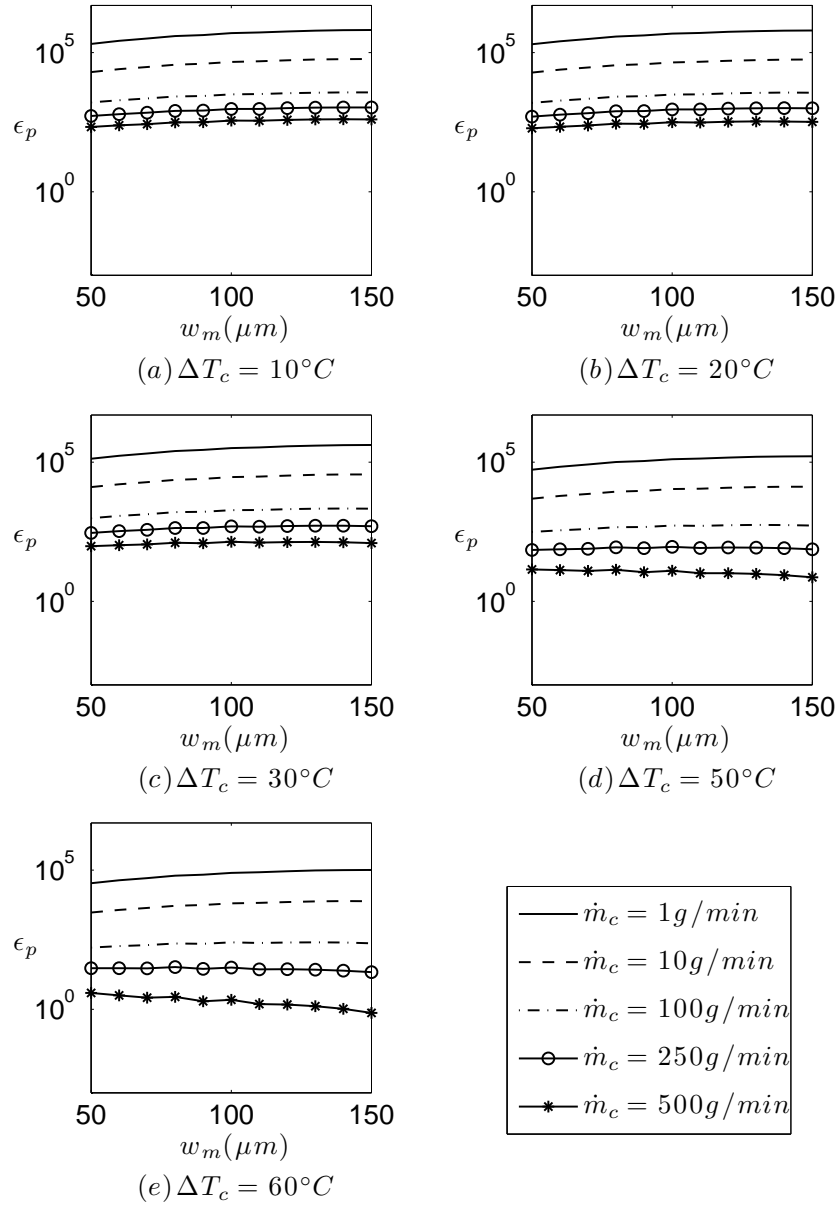


Figure B.10: Benefit to cost ratio of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

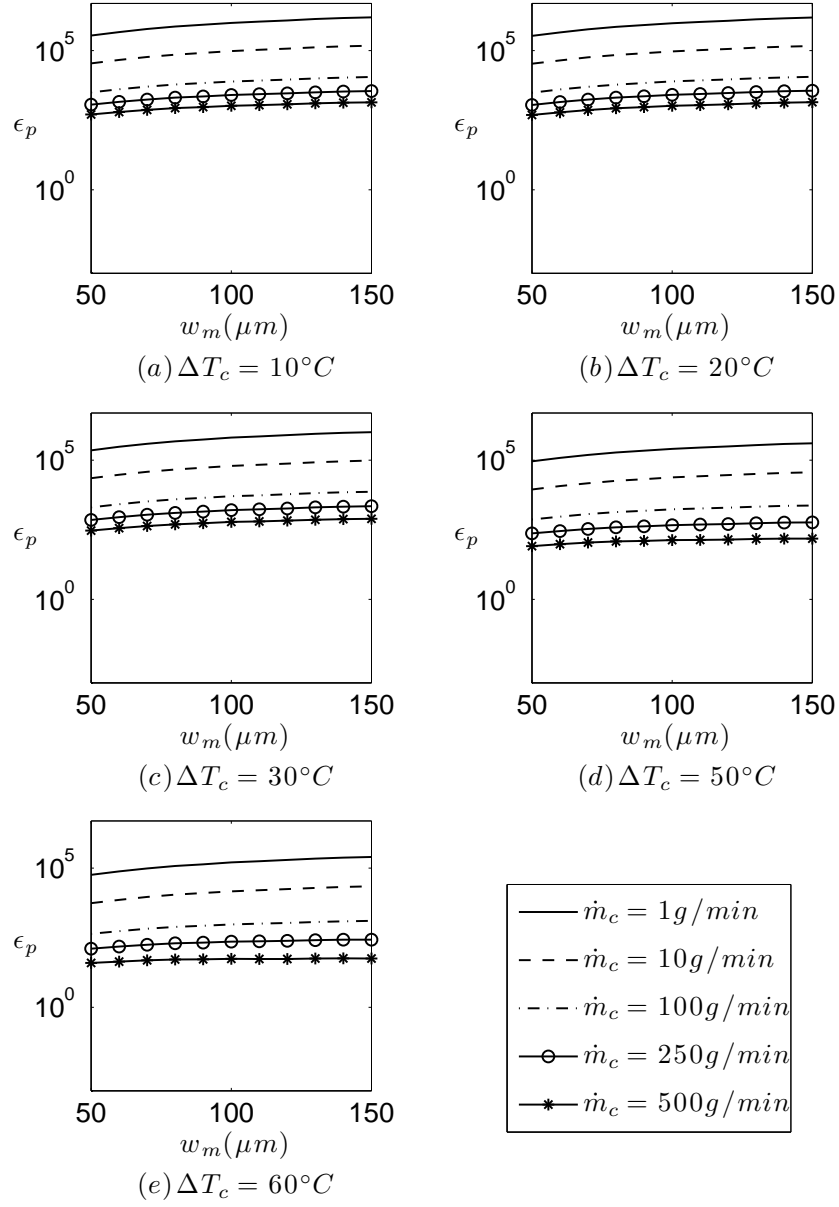


Figure B.11: Benefit to cost ratio of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

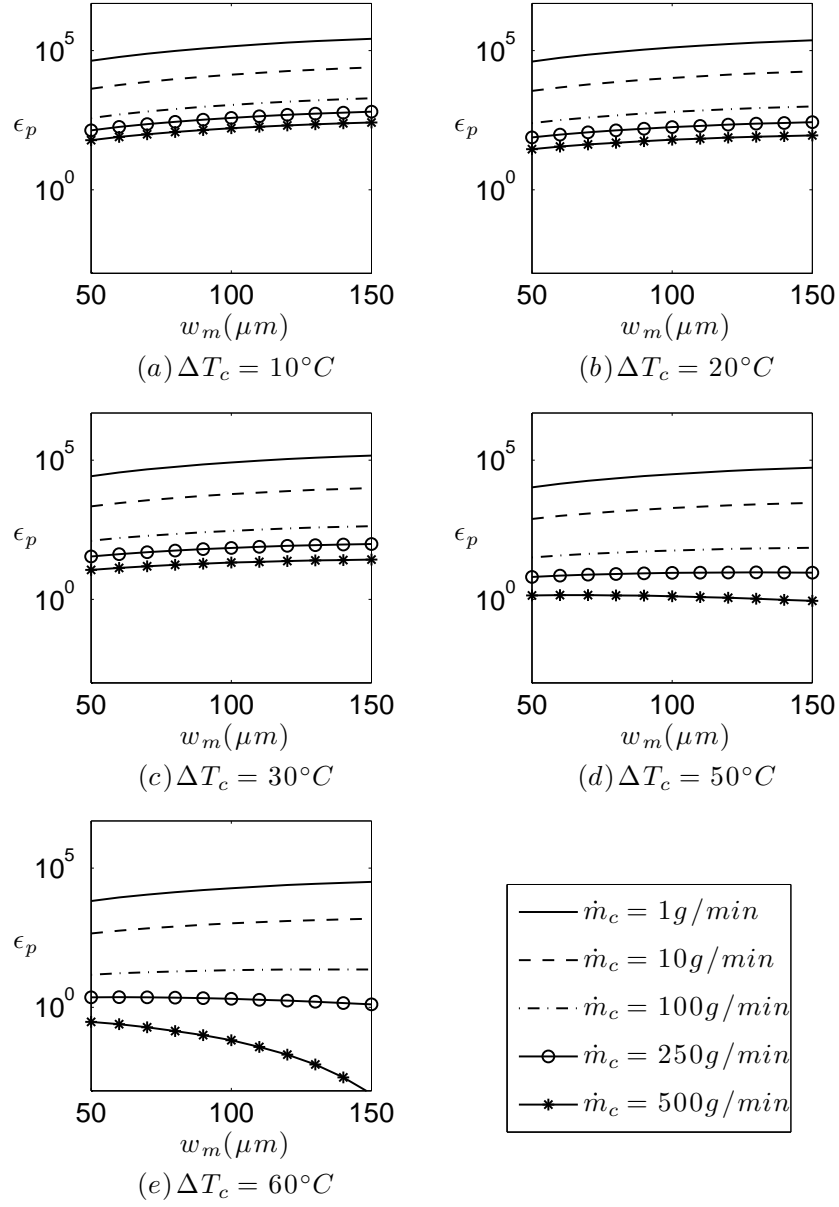


Figure B.12: Benefit to cost ratio of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

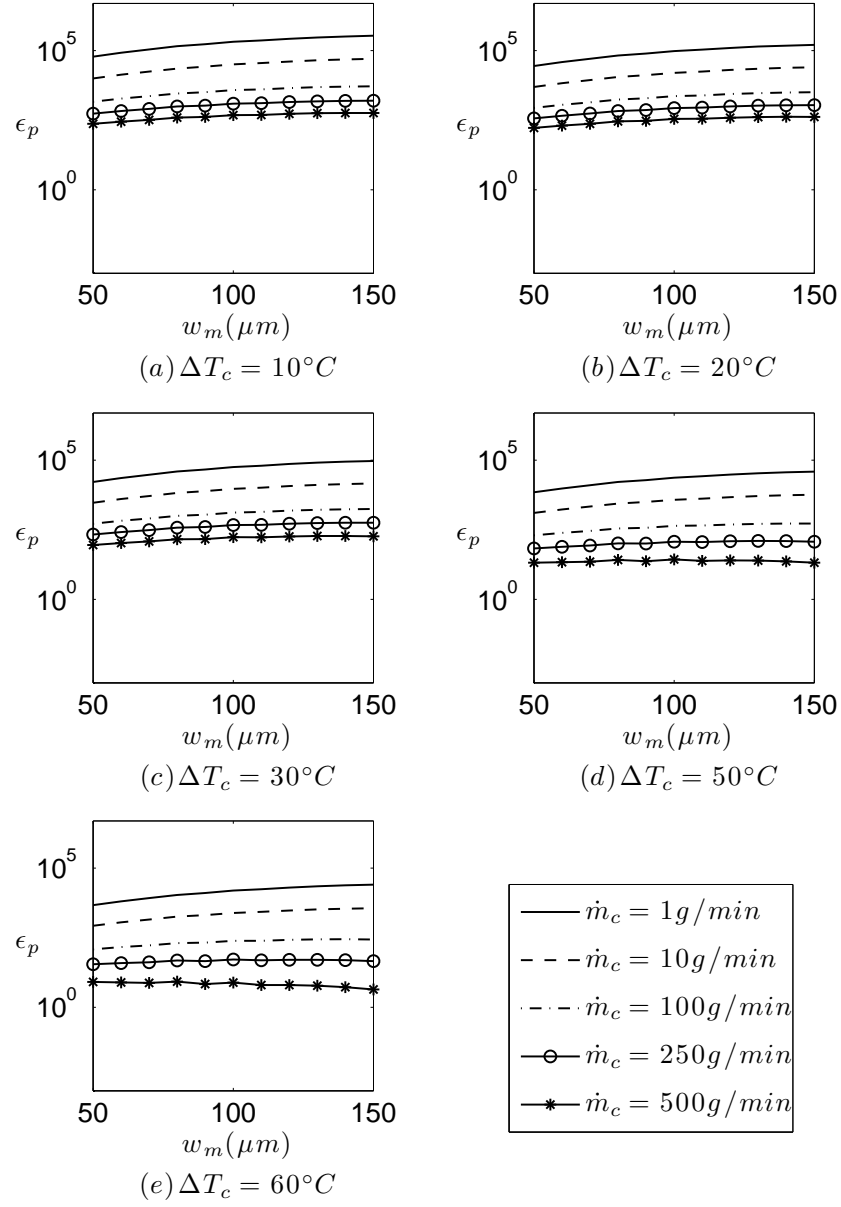


Figure B.13: Benefit to cost ratio of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

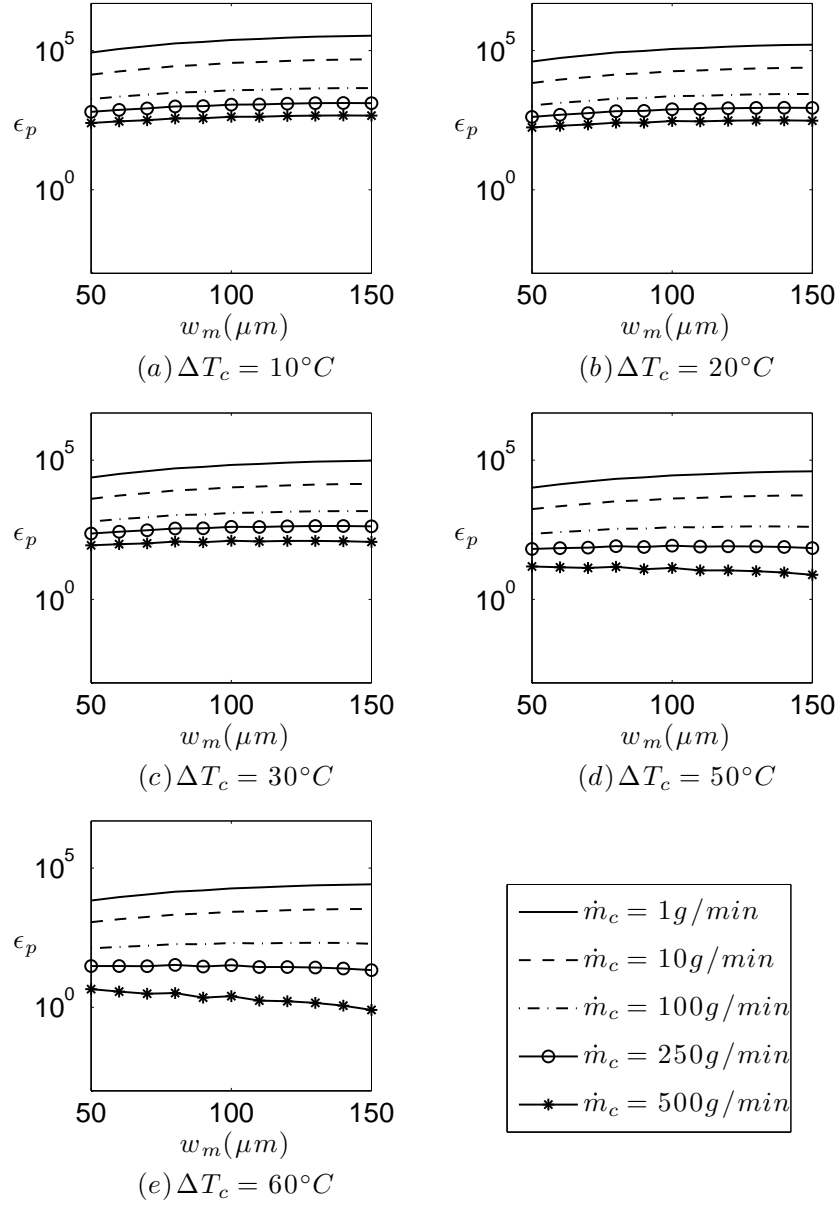


Figure B.14: Benefit to cost ratio of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

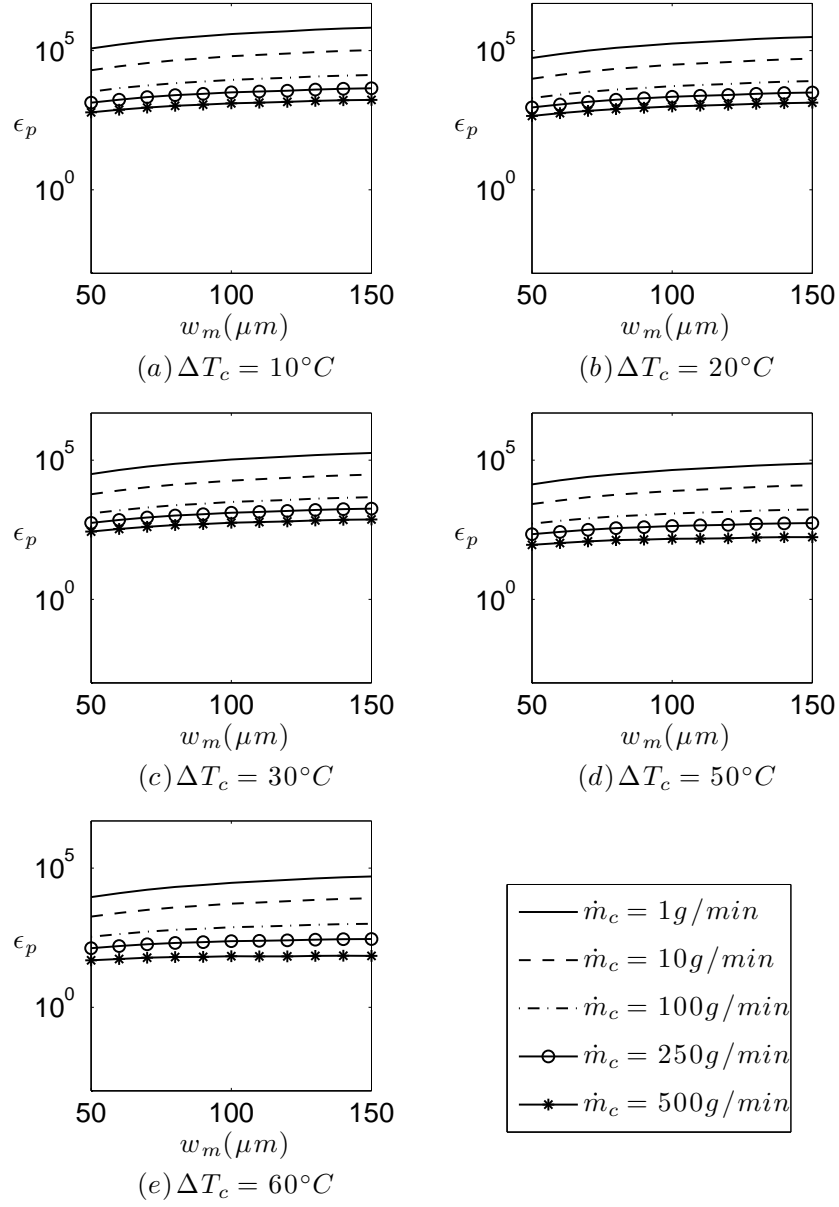


Figure B.15: Benefit to cost ratio of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

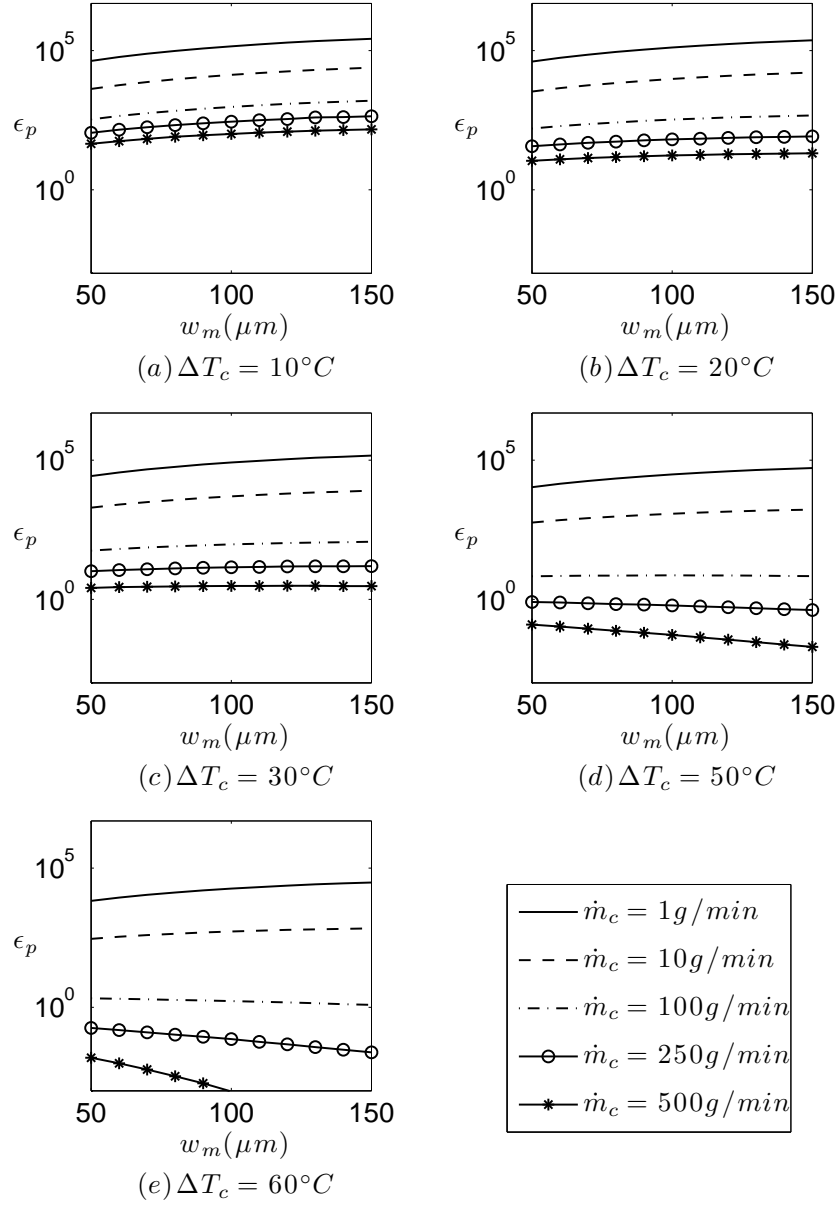


Figure B.16: Benefit to cost ratio of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

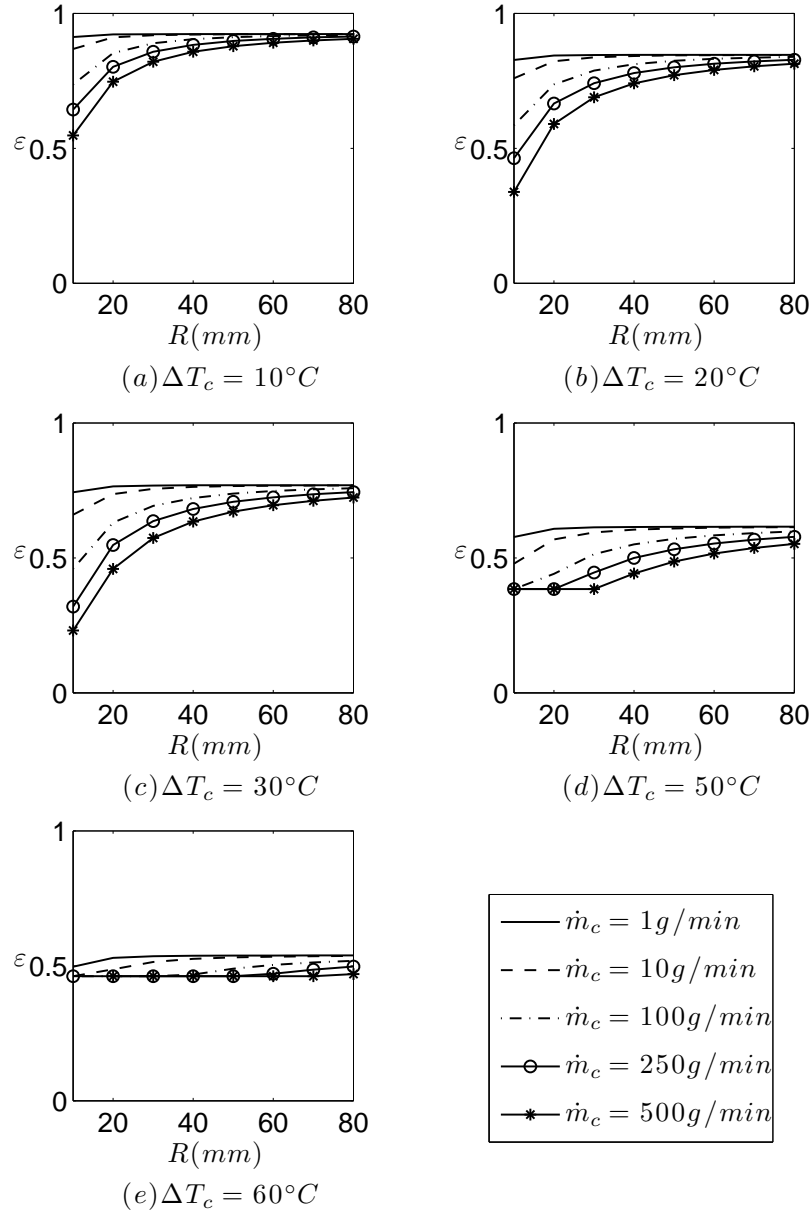


Figure B.17: Heat exchanger effectiveness of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

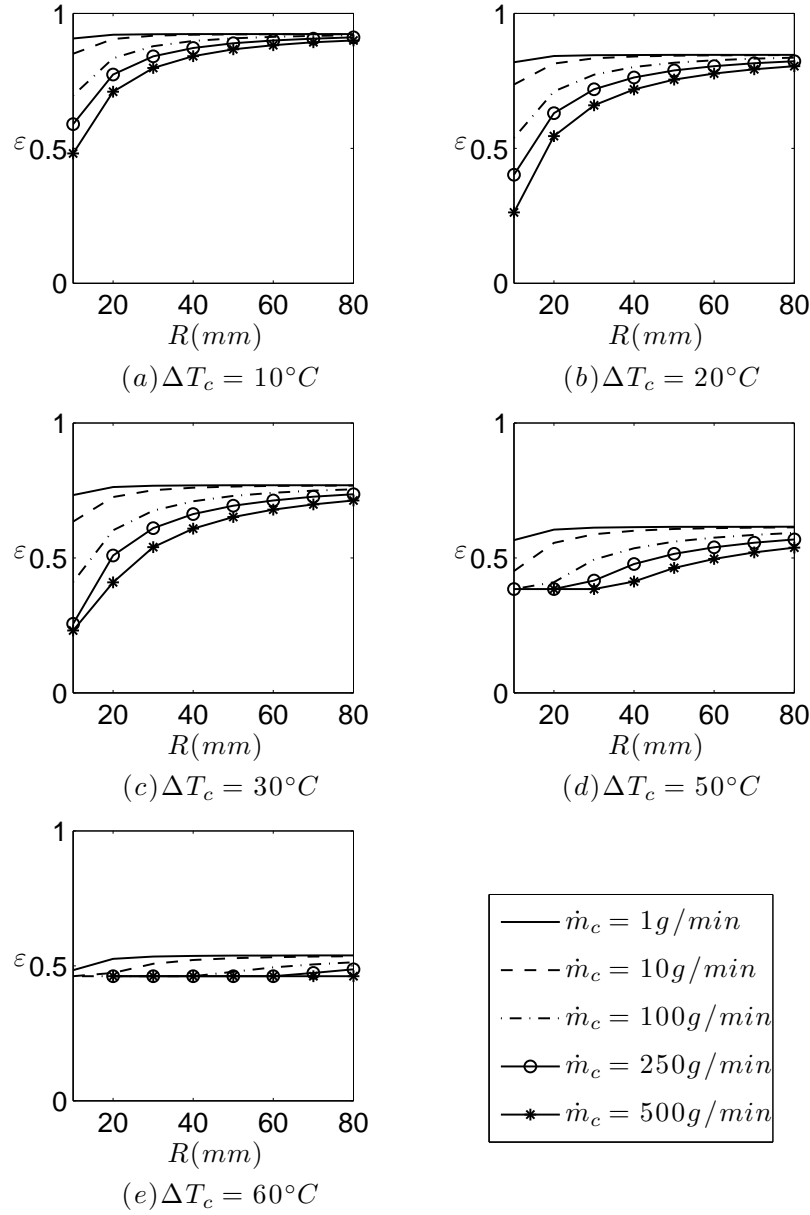


Figure B.18: Heat exchanger effectiveness of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

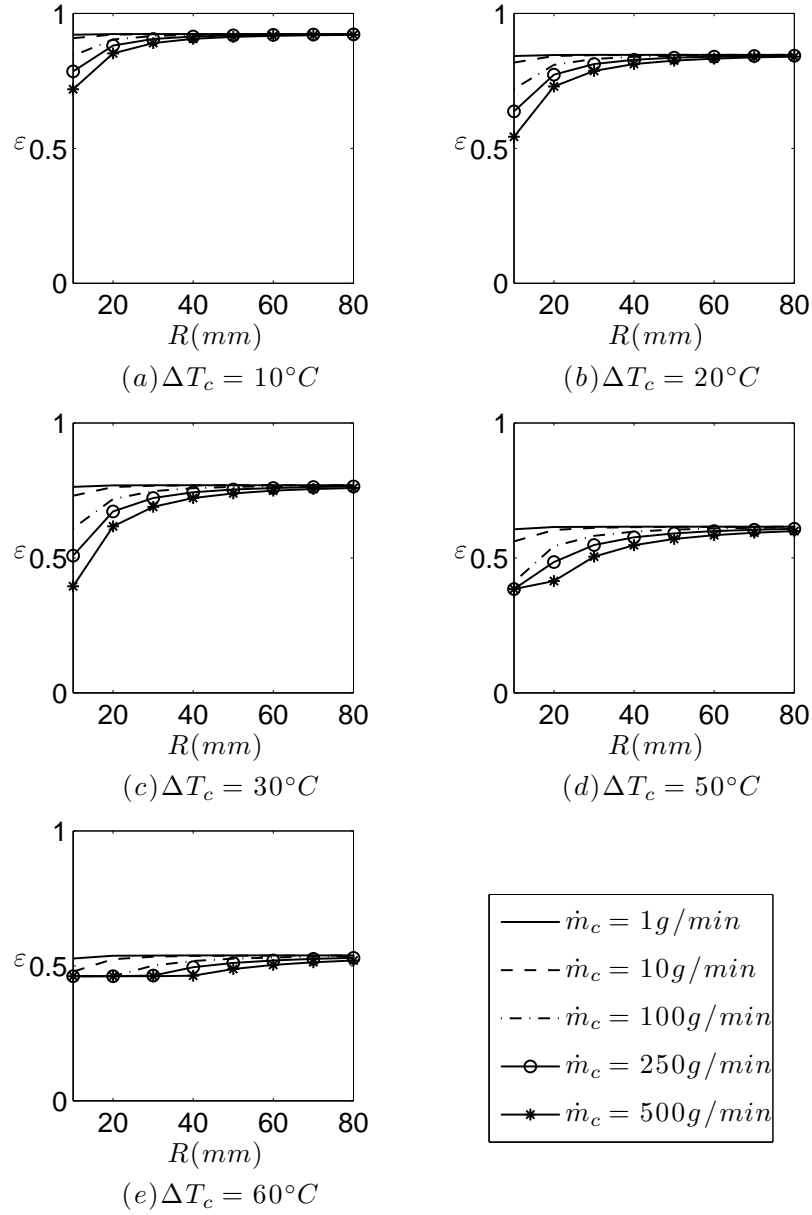


Figure B.19: Heat exchanger effectiveness of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

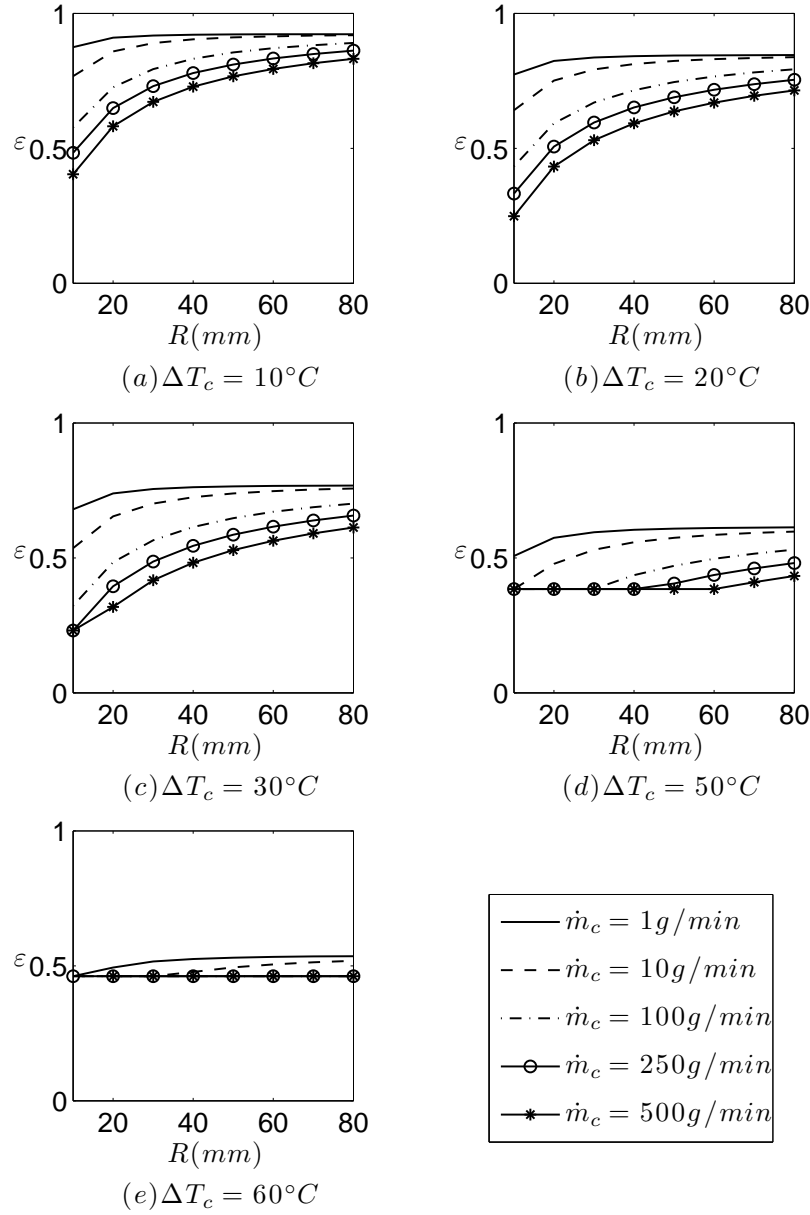


Figure B.20: Heat exchanger effectiveness of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the co-flow configuration

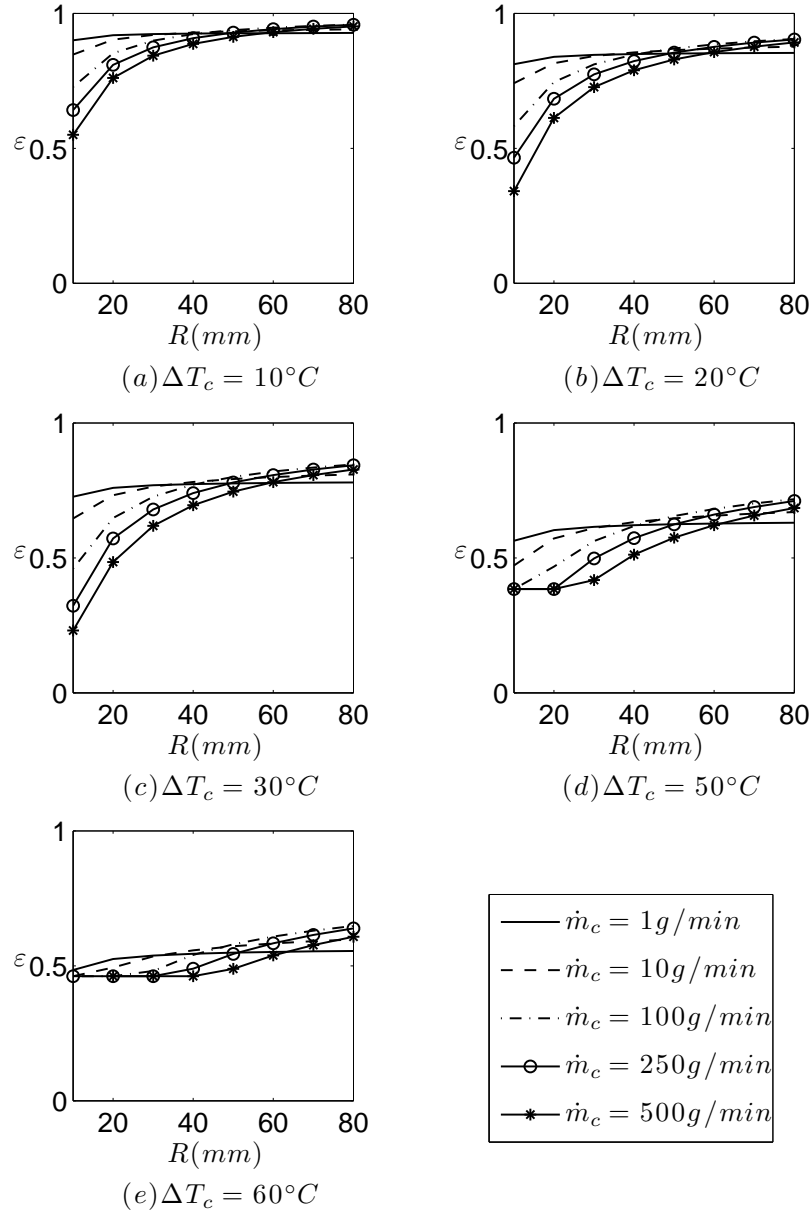


Figure B.21: Heat exchanger effectiveness of geometry G1 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

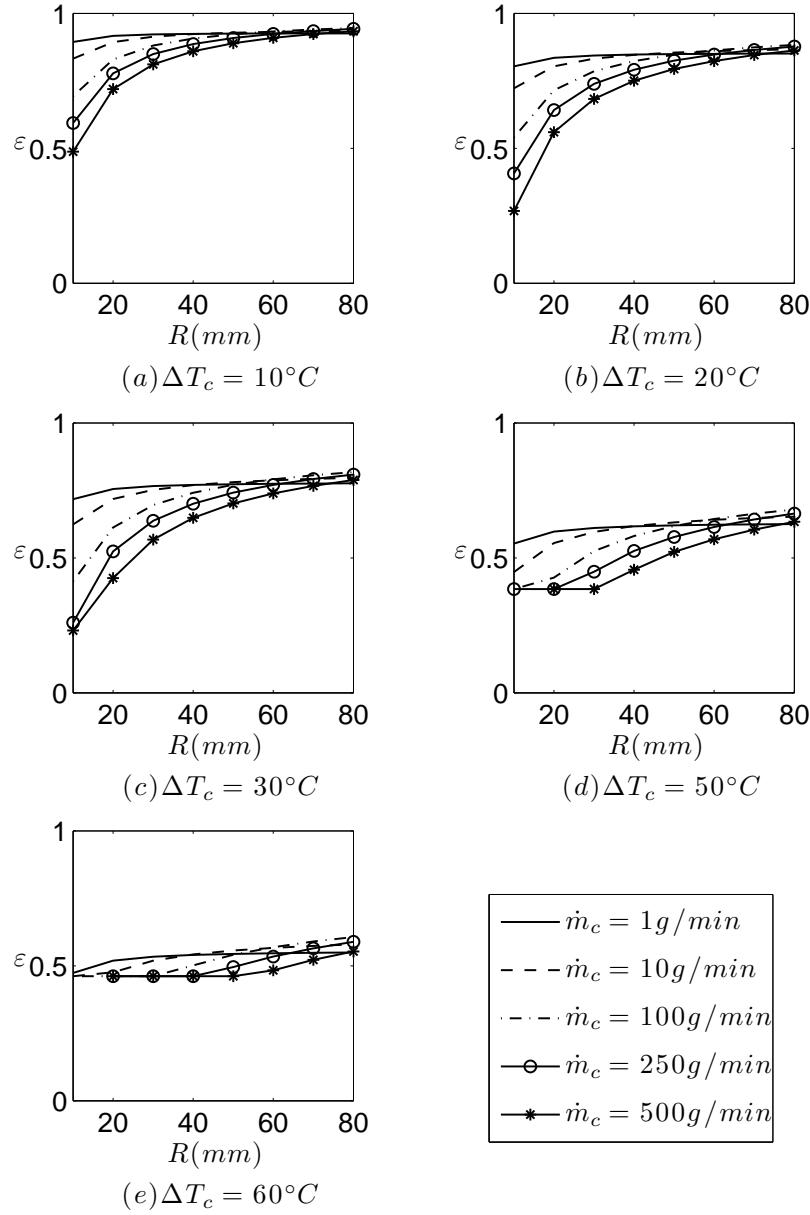


Figure B.22: Heat exchanger effectiveness of geometry G2 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

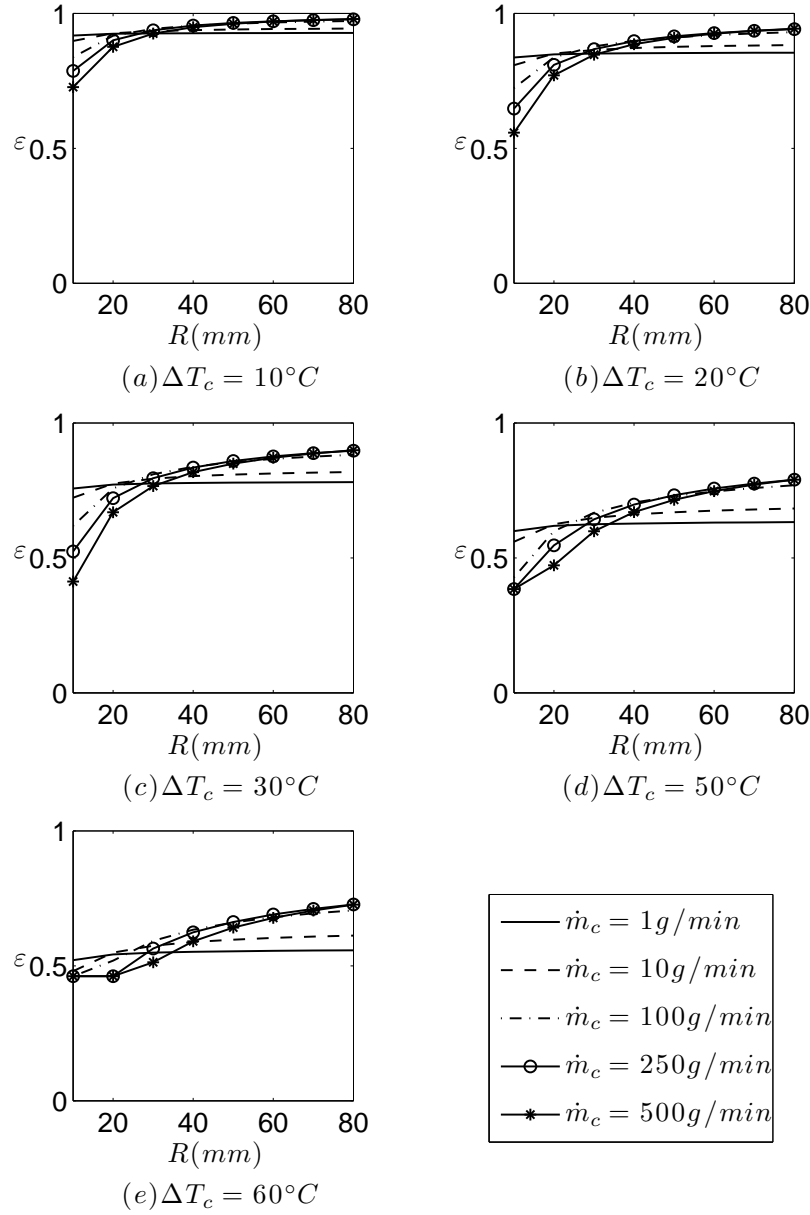


Figure B.23: Heat exchanger effectiveness of geometry G3 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

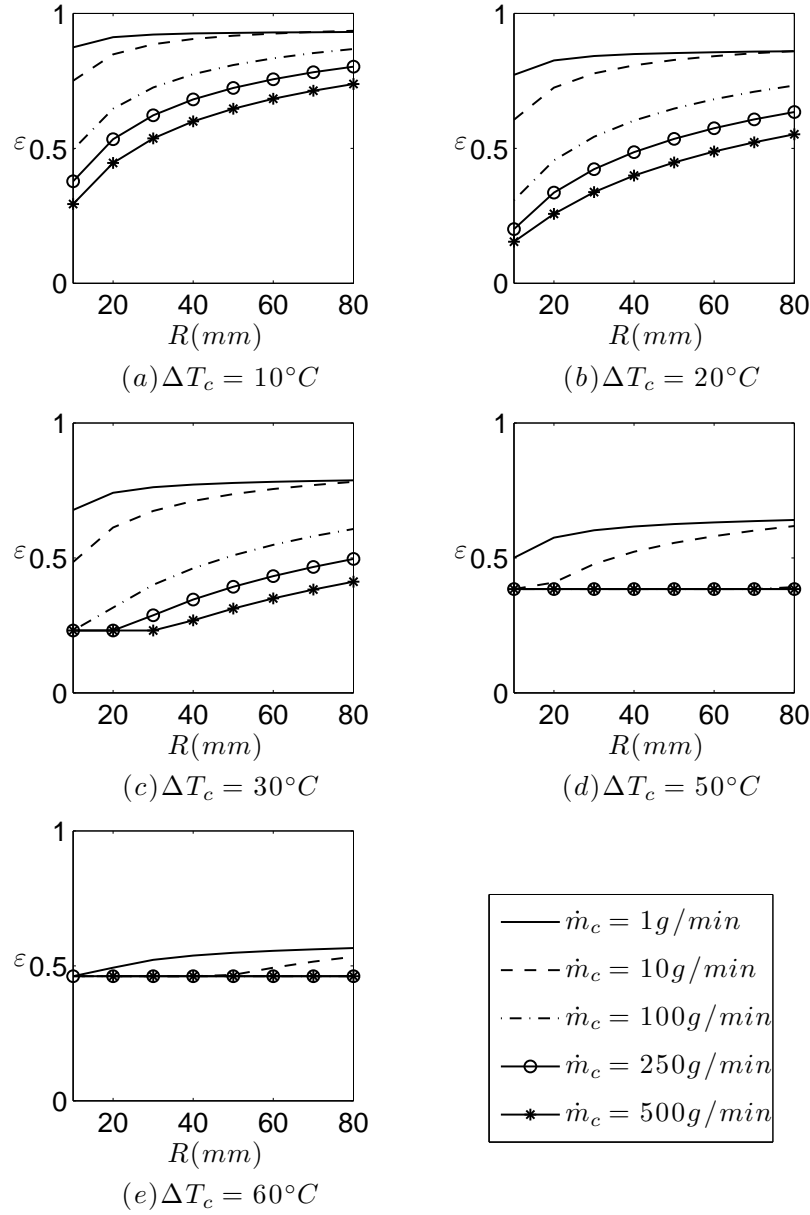


Figure B.24: Heat exchanger effectiveness of geometry G4 for varying ΔT_c , R , and \dot{m}_c in the counter-flow configuration

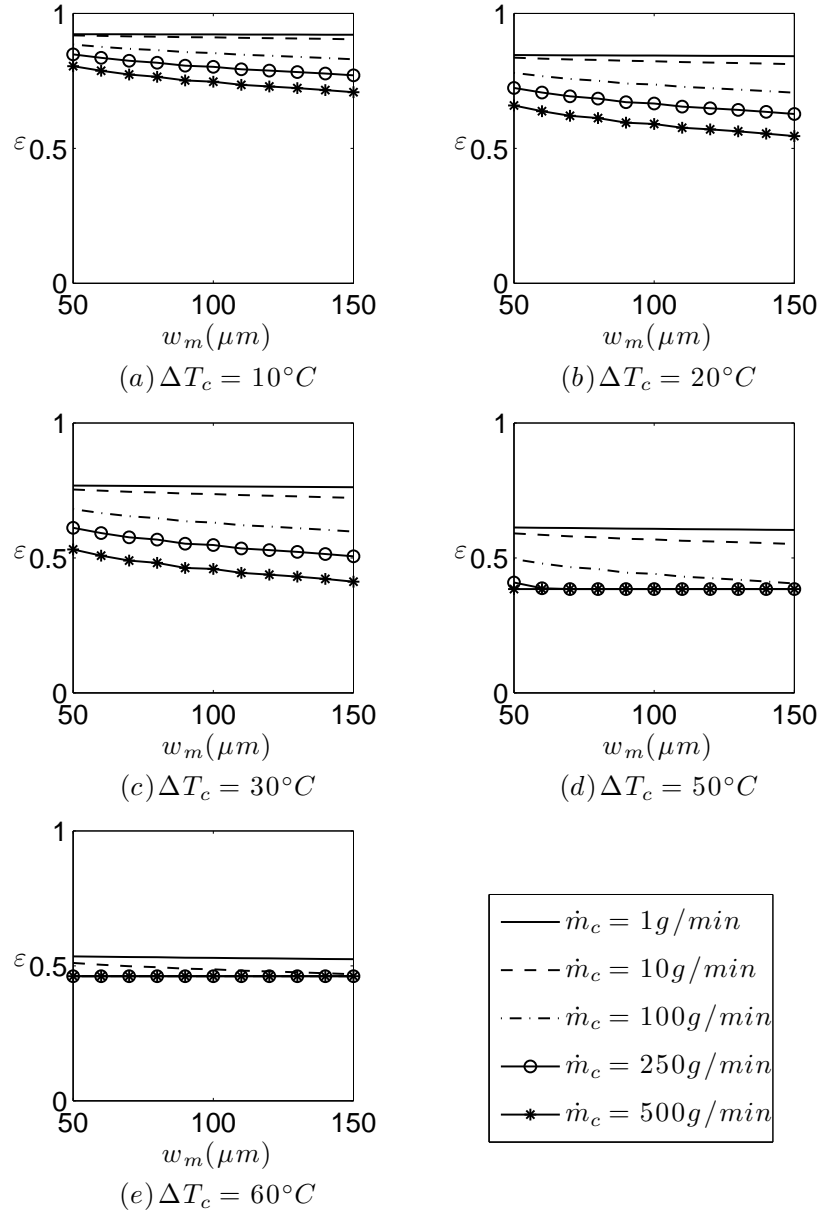


Figure B.25: Heat exchanger effectiveness of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

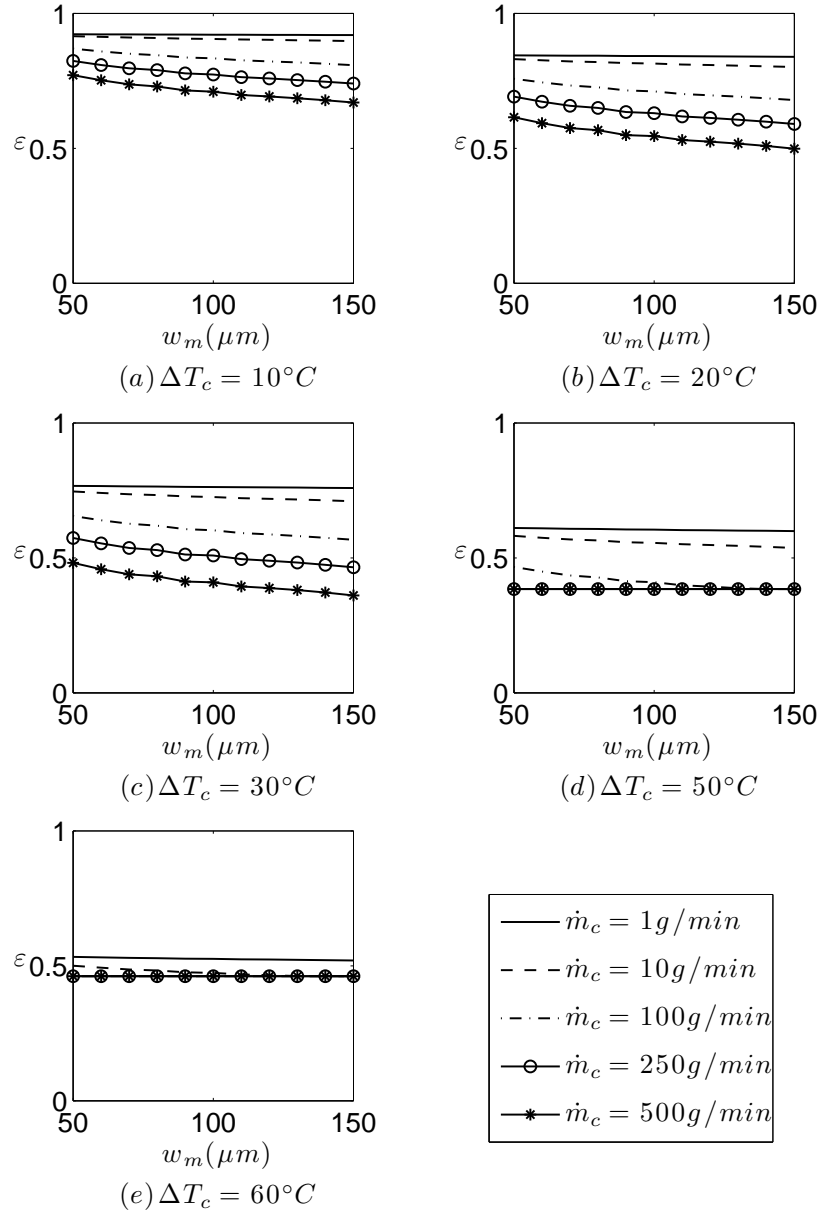


Figure B.26: Heat exchanger effectiveness of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

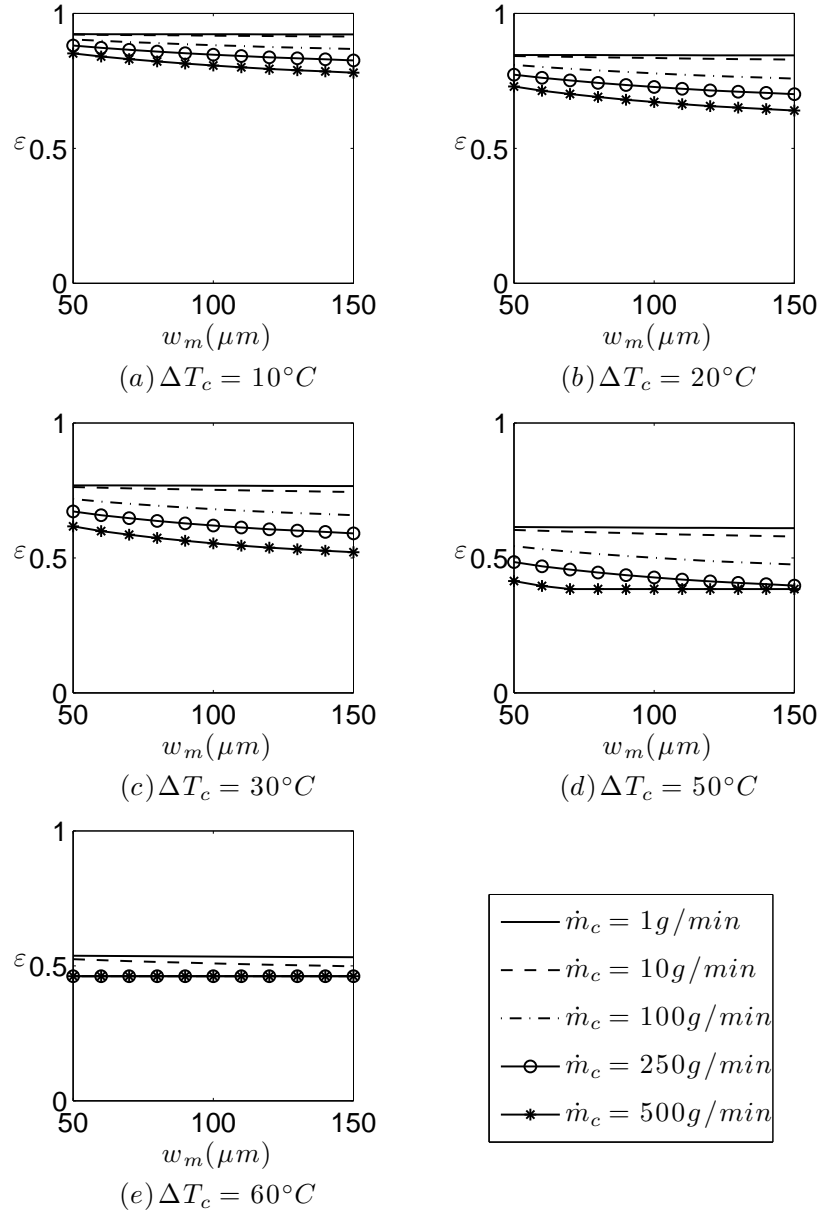


Figure B.27: Heat exchanger effectiveness of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

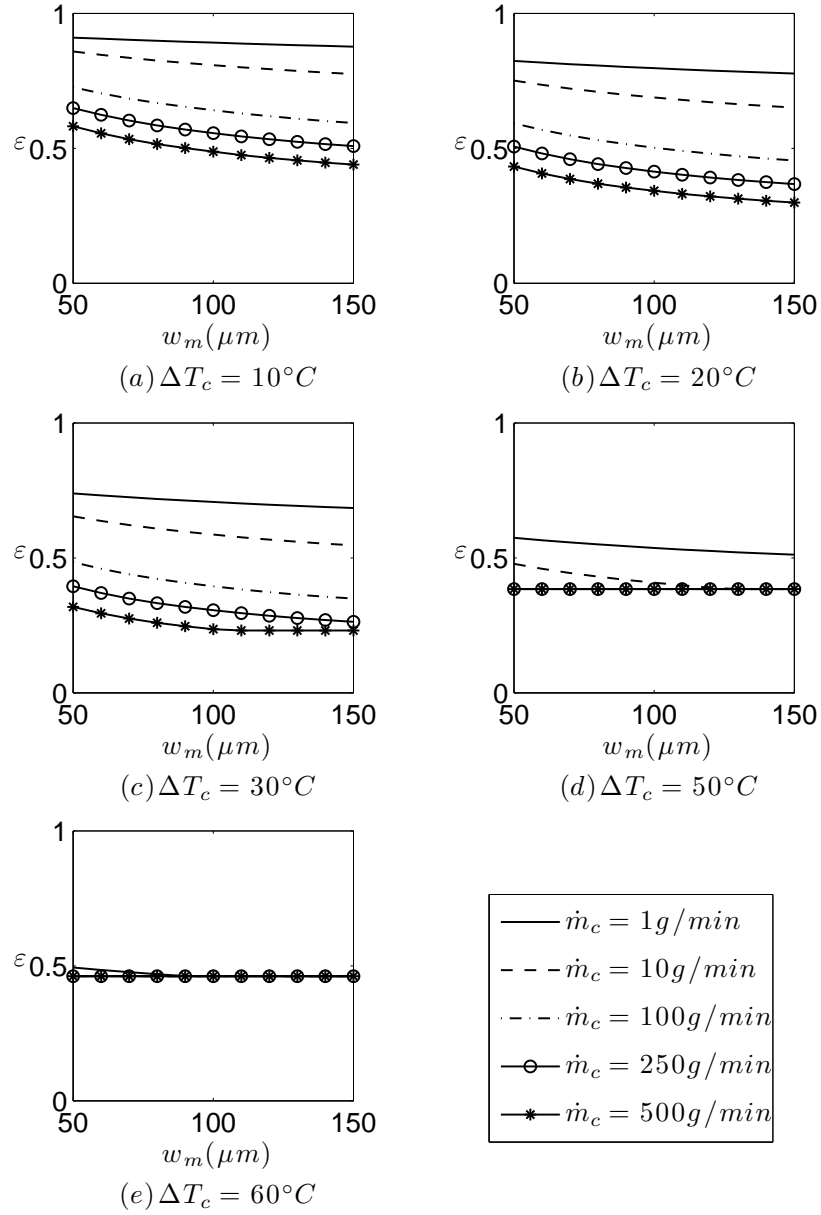


Figure B.28: Heat exchanger effectiveness of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the co-flow configuration

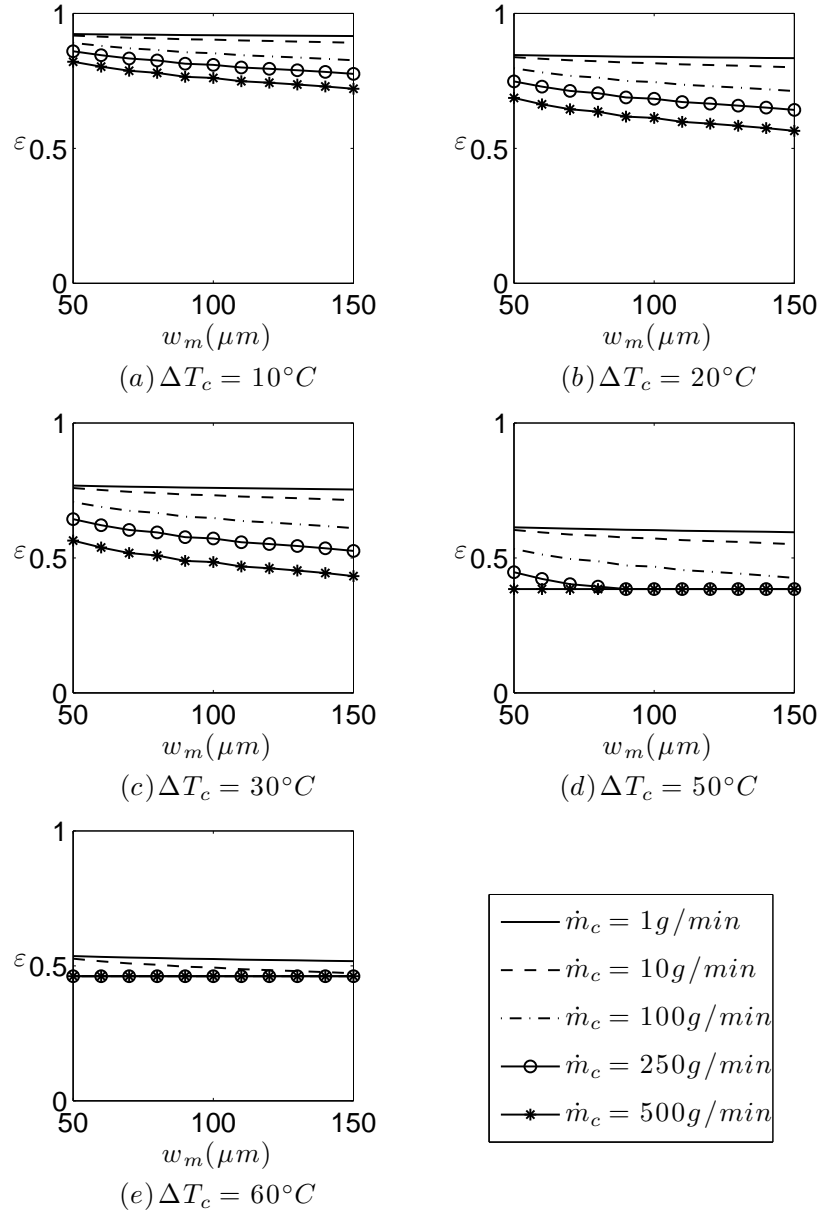


Figure B.29: Heat exchanger effectiveness of geometry G1 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

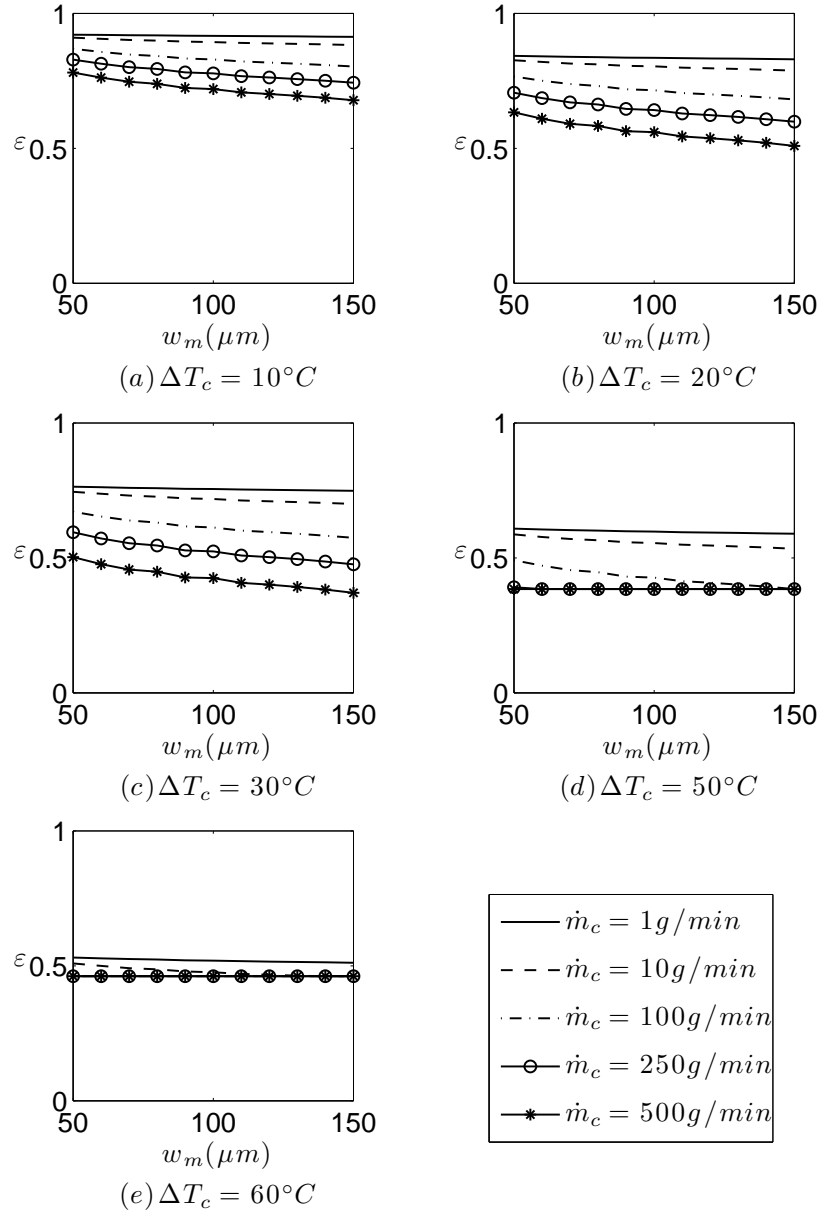


Figure B.30: Heat exchanger effectiveness of geometry G2 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

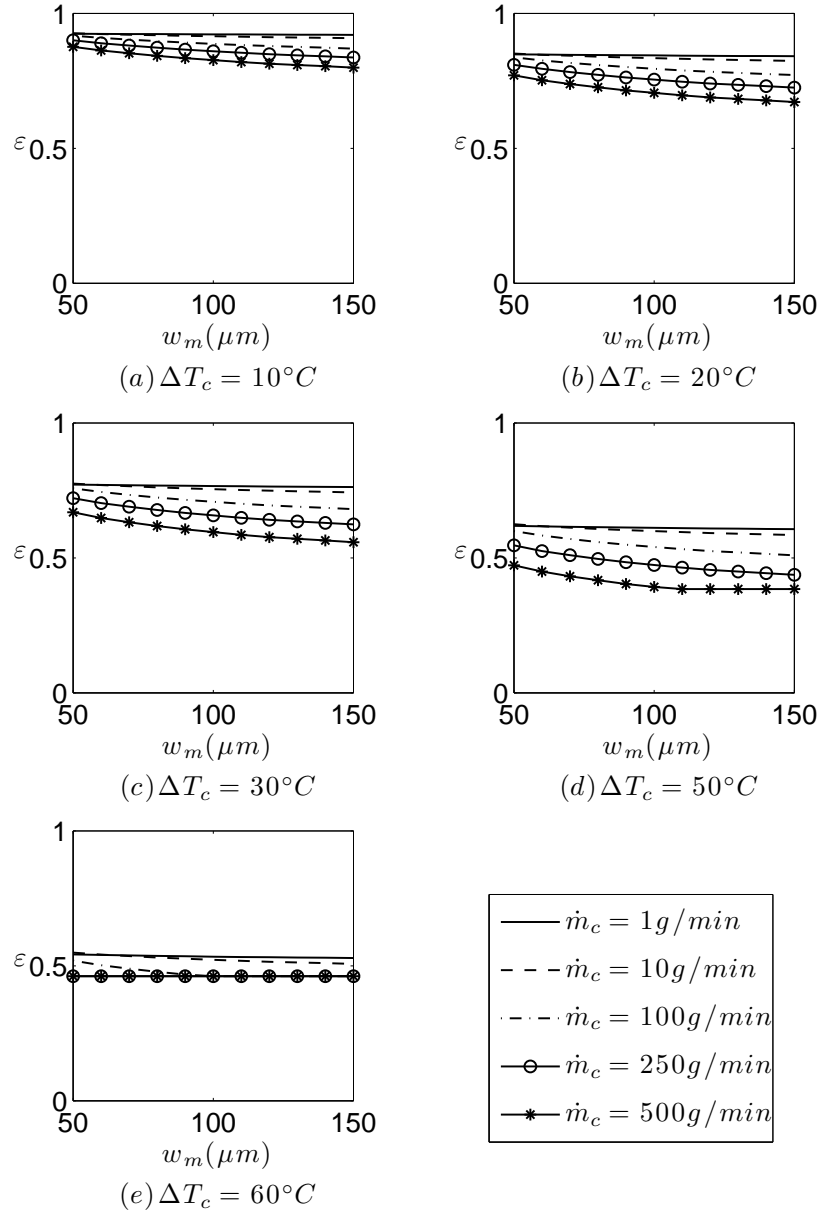


Figure B.31: Heat exchanger effectiveness of geometry G3 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

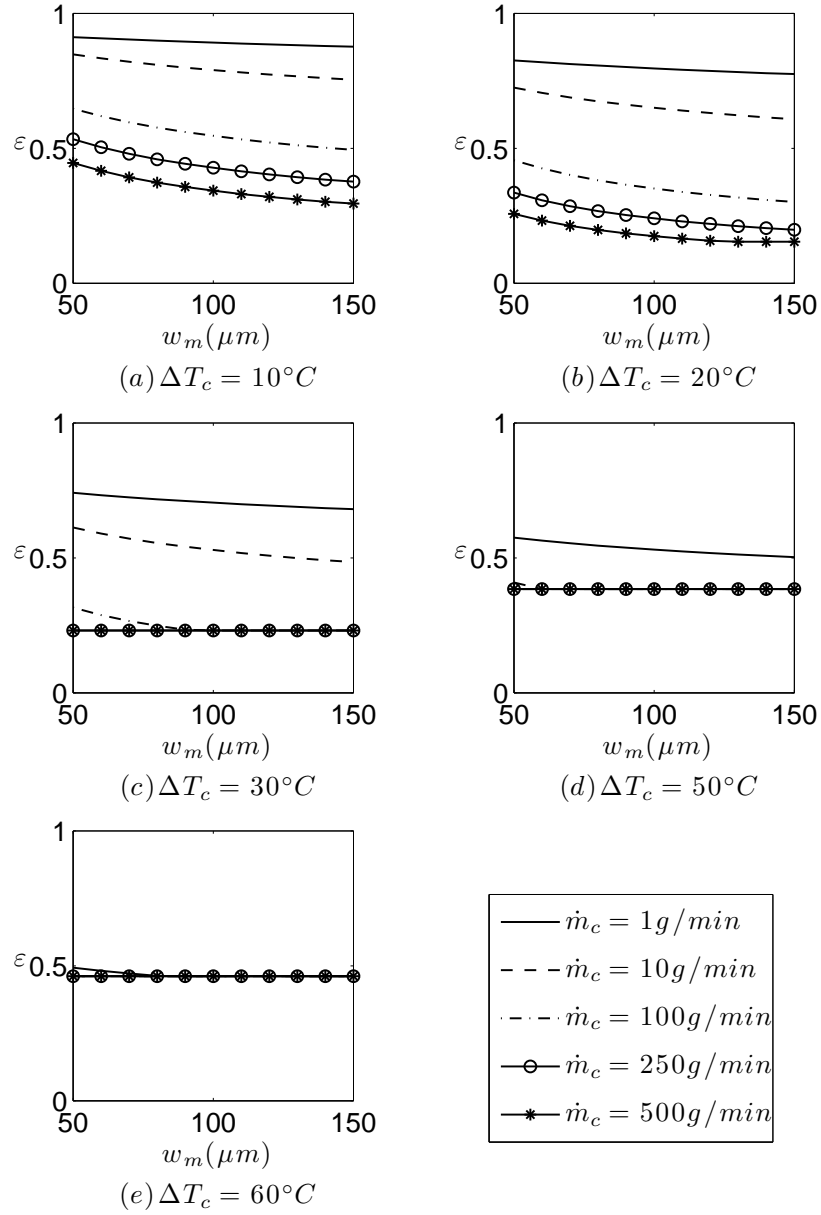


Figure B.32: Heat exchanger effectiveness of geometry G4 for varying ΔT_c , w_m , and \dot{m}_c in the counter-flow configuration

