

# Three-dimensional conductive heat-spreading layouts obtained using topology optimisation for passive internal electronic cooling

---

Francois Hector Burger

2014-02-14

# Three-dimensional conductive heat-spreading layouts obtained using topology optimisation for passive internal electronic cooling

---

*by*

**Francois Hector Burger**

27002234

*Submitted in partial fulfilment of the requirements for the degree*

MASTER OF ENGINEERING

*in the*

Department of Mechanical and Aeronautical Engineering

Faculty of Engineering, Built Environment and Information Technology

University of Pretoria

2014-02-14



## Abstract

**Title:** Three-dimensional conductive heat-spreading layouts obtained using topology optimisation for passive internal electronic cooling

**Author:** Francois Hector Burger (27002234)

**Supervisors:** Dr J. Dirker and Prof J.P. Meyer

**Department:** Mechanical and Aeronautical Engineering

**University:** University of Pretoria

**Degree:** Master of Engineering (Mechanical Engineering)

*In this study, topology optimisation for heat-conducting paths in a three-dimensional domain was investigated. The governing equations for the temperature distribution were solved using the finite volume method, the sensitivities of the objective function (average temperature) were solved using the adjoint method, and finally, the optimal architecture was found with the method of moving asymptotes (MMA) using a self-programmed code. A two-dimensional domain was evaluated first as a validation for the code and to compare with other papers before considering a three-dimensional cubic domain.*

*For a partial Dirichlet boundary, it was found that the converged architecture in three dimensions closely resembled the converged architectures from two dimensions, with the main branches extending to the outer corners of the domain. However, the partial Dirichlet boundary condition was not realistic, and to represent a more realistic case, a full Dirichlet boundary was also considered.*

*In order to investigate a full Dirichlet boundary condition, the domain had to be supplied with an initial base for the architecture to allow variation in the sensitivities. It was found that the width and height of this base had a significant effect on the maximum temperature. A height of 0.04 with a base width of 0.24 proved to be the most effective, since this small base gave the MMA enough freedom to generate a tree structure. It was first assumed that this base should be in the centre of the bottom boundary and this was later proved. The results showed again that the maximum temperature decreased with an increase in the conductivity ratio or volume constraint. The architectures were similar to the partial Dirichlet boundary, again with the main branches extending to the outer corners of the domain. The main branches were thinner compared with the partial Dirichlet boundary and fewer secondary branches were observed. It was concluded that a full Dirichlet boundary could be solved using topology optimisation, if the boundary was supplied with an initial base.*

*With the successful implementation of the full Dirichlet boundary with one initial base, multiple bases were investigated. First, two bases were used and it was assumed that the optimal placement for these*

*bases was in the centre of each respective half of the bottom boundary, which was later confirmed. The optimal width and height of 0.24 and 0.04 respectively were again optimal for each specific base. The same procedure was followed for four bases and it was assumed that the optimal placement was in the centre of each respective quadrant of the bottom boundary, which was also later confirmed. The optimal width and height of 0.12 and 0.04 respectively were found for this case. With this established, optimisation runs for different conductivity ratios and volume constraints were completed for two and four bases. It was found that two bases offered increased performance in terms of the maximum temperature compared with one base. An increase in performance was also observed when using four bases compared with two bases. A maximum of 20.4% decrease in the maximum temperature was observed when comparing four bases with one.*

**Keywords:** *topology, optimisation, conduction, three-dimensional*

## Publications

### Article in referred journal

- [1] F.H. Burger, J. Dirker and J.P. Meyer, “Three-dimensional conductive heat transfer topology optimisation in a cubic domain for the volume-to-surface problem”, *International Journal of Heat and Mass Transfer*, vol. 67, pp. 214-224, 2013.

## Acknowledgements

I want to thank:

the NRF, TESP, University of Stellenbosch/University of Pretoria, SANERI/SANEDI, CSIR, EEDSM Hub and NAC for the funding provided that made this study possible;

study supervisor, Dr J. Dirker, for his guidance, patience, friendship and always having an open door throughout this study. It has been a pleasure completing this study under him;

co-supervisor, Prof J.P. Meyer, for his critical eye and financial support without which this study would not have been possible;

my wife, Suselna, for always listening to me explaining everything and for her support;

my parents, Chris and Luzette, for their continuous support throughout the years;

Logan Page, for teaching and converting me to Python;

Gerald Nel, for all his help with the complicated coding and everyday problems;

Christo Langenhoven, for his help in creating the iso-surface drawings and the design and building of the Architecture Visualiser program;

Marc Stocks, for his assistance with the virtual private network allowing me to work from home;

# Contents

Abstract.....	i
Publications.....	iii
Acknowledgements.....	iv
List of Figures.....	ix
List of Tables.....	xiii
Nomenclature.....	xv
Chapter 1 Introduction.....	1
1.1 Background.....	1
1.2 Problem Statement and Purpose of Study.....	2
1.3 Dissertation Methodology.....	3
Chapter 2 Literature.....	4
2.1 Introduction.....	4
2.2 Geometric Optimisation.....	4
2.3 Methods of Topology Optimisation.....	5
2.3.1 Continuous Design Variable Methods.....	6
2.3.2 Discrete Design Variable Methods.....	9
2.3.3 Other Methods.....	11
2.4 Numerical Instabilities.....	14
2.4.1 Checkerboards.....	14
2.4.2 Filtering Techniques.....	15
2.5 Summary.....	15
Chapter 3 Two-Dimensional Numerical Model.....	17
3.1 Introduction.....	17
3.2 Domain Discretisation and Thermal Model.....	17
3.2.1 Methods.....	17
3.2.2 Governing Equations of the Finite Volume Method.....	18
3.2.3 Domain for a Two-Dimensional Thermal Model.....	19
3.2.4 Finite Volume Method for a Two-Dimensional Thermal Model.....	21
3.3 Adjoint Method.....	23

3.3.1	Governing Equations.....	23
3.3.2	Adjoint Method for Two Dimensions.....	24
3.4	Method of Moving Asymptotes.....	27
3.4.1	Governing Equations.....	27
3.4.2	Method of Moving Asymptotes for Two Dimensions.....	28
3.5	Program Methodology.....	30
3.6	Summary.....	32
Chapter 4	Two-Dimensional Validation and Results.....	33
4.1	Introduction.....	33
4.2	Validation.....	33
4.2.1	Temperature Distribution.....	33
4.2.2	Adjoint Method.....	34
4.2.3	Method of Moving Asymptotes.....	34
4.3	Methodology and Results.....	35
4.3.1	Influence of Input Parameters.....	35
4.3.2	The Influence of Asymptote Parameters, $s$ and $s_0$ .....	38
4.3.3	Mesh-Dependence Study.....	41
4.3.4	Comparing Results With Other Papers.....	42
4.4	Summary.....	43
Chapter 5	Three-Dimensional Methodology and Results for a Partial Dirichlet Boundary.....	44
5.1	Introduction.....	44
5.2	Domain.....	44
5.3	Implementing the Third Dimension.....	45
5.3.1	Temperature Distribution.....	45
5.3.2	Adjoint Method.....	46
5.3.3	Method of Moving Asymptotes.....	47
5.4	Validation.....	47
5.4.1	Temperature Distribution.....	47
5.4.2	Adjoint Method.....	47
5.5	Methodology.....	47
5.5.1	Dimensionless Maximum Temperature.....	47
5.5.2	Effect of the Asymptote Parameters, $s$ and $s_0$ .....	48
5.5.3	Mesh-Dependence Study.....	51
5.5.4	Iteration-Dependence Study.....	54
5.5.5	Constant Penalisation vs. Incremental Increasing Penalisation.....	58
5.5.6	Initial Density Distribution.....	62



5.5.7	Effect of Cold Spot Size.....	65
5.6	Results for Different Conductivity Ratios and Volume Constraints.....	66
5.7	Summary .....	70
Chapter 6	Three-Dimensional Methodology and Results for the Full Dirichlet Boundary.....	71
6.1	Introduction.....	71
6.2	Domain.....	71
6.3	Methodology .....	73
6.3.1	Dimensionless Maximum Temperature .....	73
6.3.2	The Effect of the Fixed Subdomain Height .....	73
6.3.3	Effect of the Asymptote Parameters, $s$ and $s_0$ .....	79
6.3.4	Mesh-Dependence Study .....	79
6.3.5	Iteration-Dependence Study.....	82
6.3.6	Placement of Base Structure .....	83
6.4	Results for Different Conductivity Ratios and Volume Constraints.....	88
6.5	Summary .....	91
Chapter 7	Three-Dimensional Methodology and Results for Multiple Bases.....	92
7.1	Introduction.....	92
7.2	Two Base Structures .....	92
7.2.1	Domain.....	92
7.2.2	Size of the Base Structures.....	94
7.2.3	Placement of the Base Structures.....	94
7.2.4	Results for Different Conductivity Ratios and Volume Constraints.....	97
7.3	Four Base Structures .....	98
7.3.1	Domain.....	98
7.3.2	Size of the Base Structures.....	99
7.3.3	Placement of the Base Structures.....	100
7.3.4	Results for Different Conductivity Ratios and Volume Constraints.....	100
7.4	Comparison of Thermal Results for Different Number of Initial Bases .....	102
7.5	Summary .....	104
Chapter 8	Conclusions and Recommendations .....	105
References.....		108
A.	Two-Dimensional Results.....	I
A.1	Effect of $s$ and $s_0$ for Two Dimensions .....	I
B.	Three-Dimensional Results.....	III
B.1	Partial Dirichlet Boundary Results.....	III
B.1.1	Results for Different Conductivity Ratios and Volume Constraints .....	III

B.2 Full Dirichlet Boundary Results.....	VI
B.2.1 Effect of the Asymptote Parameters, $s$ and $s_0$ .....	VI
B.2.2 Mesh-Dependence Study.....	VIII
B.2.3 Iteration-Dependence Study.....	VIII
B.2.4 Results for Different Conductivity Ratios and Volume Constraints.....	IX
B.3 Full Dirichlet Boundary Results for Two Bases.....	XII
B.3.1 Size of the Base Structure.....	XII
B.4 Full Dirichlet Boundary Results for Four Bases.....	XIV
B.4.1 Size of the Base Structures.....	XIV

## List of Figures

Figure 1.1: Optimised architectures for heat transfer using a) SIMP [8] b) bionic optimisation [11] c) constructal theory [12].	2
Figure 2.1: Material distribution obtained for a load-bearing structure using topology optimisation [19].	4
Figure 2.2: Typical topological domain showing the conductivity of elements, element boundaries and the isothermal boundary.	6
Figure 2.3: Material distribution at different thermal conductivity ratios [9].	8
Figure 2.4: The results for a 128x128 grid with a) FEM with a filter b) FVM with a filter c) FVM with no filter [8].	8
Figure 2.5: Optimal configuration obtained using an isothermal boundary in the corner of the domain, using homogenisation [22].	9
Figure 2.6: Optimal geometry obtained with all outer boundaries at a constant temperature, with four internal heat sources, using ESO [25].	10
Figure 2.7: Optimal material distribution obtained with all boundaries at a constant temperature and uniform internal heat generation, using BESO [26].	11
Figure 2.8: The cellular automaton procedure [27].	11
Figure 2.9: Optimal geometry obtained for multiple load cases, using the level set method [18].	12
Figure 2.10: a) optimal solution obtained by bionic optimisation b) regularised solution for use in engineering application [11].	13
Figure 2.11: Completed structures obtained by constructal theory [12].	13
Figure 2.12: Fractal theory illustrated in both shapes, where the final fractal is just an enlarged version of the smallest fractal [38].	14
Figure 2.13: Checkerboard patterns [19].	14
Figure 3.1: Control volume for a general case for a current and neighbour element.	19
Figure 3.2: Domain for partial Dirichlet boundary located on the bottom edge, for the two- dimensional case.	20
Figure 3.3: Program methodology.	31
Figure 4.1: Domain for one-dimensional validation.	33
Figure 4.2: A scale of the density.	35
Figure 4.3: The influence of $s_0$ for the two-dimensional partial Dirichlet boundary.	39
Figure 4.4: The influence of $s_0$ and $\theta_0$ for $s = 0.7$ for the two-dimensional case.	40
Figure 4.5: The effect of $s$ on the maximum temperature for different values of $s_0$ with $\theta_0 = 0.1$ .	40
Figure 4.6: Mesh-dependence for the two-dimensional case.	41
Figure 5.1: Domain for a partial Dirichlet boundary condition located on the bottom boundary, for a three-dimensional case.	45
Figure 5.2: The effect of $s_0$ and $s$ on $\tau$ for a three-dimensional domain partial Dirichlet boundary.	48
Figure 5.3: The effect of $s_0$ and $s$ on the converged volume ratio for a three-dimensional domain with a partial Dirichlet boundary.	49

Figure 5.4: Mesh-dependence study for a three-dimensional domain using a partial Dirichlet boundary. .... 51

Figure 5.5: The converged volume ratio for the mesh-dependence study for a three-dimensional domain with a partial Dirichlet boundary. .... 52

Figure 5.6: Shown in a) a side view of the architecture b) a top view of the architecture ( $M = 50$ ,  $s_0 = 0.15$ ). .... 53

Figure 5.7: An isometric view showing an isosurface of the converged architecture for the  $M = 50$  case from the mesh-dependence study for a) full domain b) half of the domain  $y$ . .... 54

Figure 5.8: The effect of the number of iterations on  $\tau$  for a three-dimensional domain with a partial Dirichlet boundary. .... 55

Figure 5.9: The effect on the volume ratio due to the number of iterations for a three-dimensional domain with a partial Dirichlet boundary. .... 56

Figure 5.10: The effect of the  $s_0$  and  $I$  on  $\tau$  for a three-dimensional domain with a partial Dirichlet boundary. .... 57

Figure 5.11: The effect of the asymptotes and  $I$  on the volume ratio for a three-dimensional domain with a partial Dirichlet boundary. .... 58

Figure 5.12: The effect of constant and increasing penalisation on  $\tau$  for a three-dimensional domain with a partial Dirichlet boundary. .... 59

Figure 5.13: The effect of constant and increasing penalisation on the volume ratio for a three-dimensional domain with a partial Dirichlet boundary. .... 60

Figure 5.14: The effect on  $\tau$  when using a random initial density distribution (2 – 8) compared with an even initial density distribution (1). .... 63

Figure 5.15: Comparison of all volume constraints to all conductivity ratios with  $s_0 = 0.15$  for a three-dimensional domain with a partial Dirichlet boundary. .... 68

Figure 5.16: The architecture for  $k^* = 2000$  and  $V^* = 0.05$  a) side view b) top view. .... 68

Figure 5.17: Isometric view showing an iso-surface of the architecture obtained for  $k^* = 2000$  and  $V^* = 0.05$ . .... 69

Figure 5.18: Temperature distribution for  $k^* = 500$  for a three-dimensional domain with a partial Dirichlet boundary. .... 70

Figure 6.1: A converged architecture for a full Dirichlet boundary. .... 71

Figure 6.2: Domain for the full Dirichlet boundary for the three-dimensional case. .... 72

Figure 6.3: The effect of the base width and height on the maximum temperature for  $k^* = 500$  for a three-dimensional domain with a full Dirichlet boundary. .... 74

Figure 6.4: Influence of  $H_b/L_D$  for  $c/L_D = 0.24$  and  $0.32$ . .... 74

Figure 6.5: The effect on  $\tau$  for different  $k^*$  and  $V^*$  ratios for  $H_b/L_D = 0.04$  and different values of the seed width,  $c/L_D$ . .... 77

Figure 6.6: Comparison of converged architectures with  $H_b/L_D = 0.04$  and solid blocks with maximum heights. .... 77

Figure 6.7: Temperature distribution for a) the solid block ( $c/L_D = 0.08$ ,  $H_b/L_D = 1.0$ ) and b) the MMA converged architecture ( $c/L_D = 0.08$ ,  $H_b/L_D = 0.04$ ). .... 79

Figure 6.8: A top view of a converged architecture for a) a partial Dirichlet boundary and b) a full Dirichlet boundary ( $k^* = 500$ ,  $V^* = 0.05$ ). .... 81

Figure 6.9: A side view of a converged architecture for a) a partial Dirichlet boundary and b) a full Dirichlet boundary ( $k^* = 500$ ,  $V^* = 0.05$ ). .... 81

Figure 6.10: An isometric view showing the converged architecture for a full Dirichlet boundary a) a full domain and b) half the domain ( $k^* = 500$ ,  $V^* = 0.05$ ). .... 82

Figure 6.11: The parameters used for moving the base of the tree. .... 84

Figure 6.12: The effect on  $\tau$  due to the position of the initial base tree for a three-dimensional domain with a full Dirichlet boundary. .... 85

Figure 6.13: A converged architecture with the initial block on the edge of the bottom boundary ( $x_i/L_D = 0.38, y_i/L_D = 0.0$ ). .... 86

Figure 6.14: Isometric view for the architecture obtained with the initial block just off-centre in one of the quarters of the bottom boundary ( $x_i/L_D = 0.26, y_i/L_D = 0.26$ ). .... 87

Figure 6.15: Isometric view for the architecture obtained with the initial block in the corner of the bottom boundary ( $x_i/L_D = 0.38, y_i/L_D = 0.38$ ). .... 87

Figure 6.16: The graph showing a comparison of all the volume constraints for a  $s_0 = 0.15$  for a three-dimensional domain with a full Dirichlet boundary. .... 89

Figure 6.17: Temperature distribution for  $k^* = 500$  for a three-dimensional domain with a full Dirichlet boundary. .... 90

Figure 7.1: Domain for two base structures for a three-dimensional domain with a full Dirichlet boundary. .... 93

Figure 7.2: The placement of the base structures. .... 93

Figure 7.3: The effect on  $\tau$  when moving the two base structures in the same half of the domain. .... 95

Figure 7.4: Domain for moving two base structures in different halves of the domain. .... 96

Figure 7.5: The effect on  $\tau$  when moving the two base structures in the different halves of the bottom domain. .... 96

Figure 7.6: Comparison of all the volume constraints for a three-dimensional domain with a full Dirichlet boundary and two initial bases. .... 97

Figure 7.7: A converged architecture with two initial bases,  $k^* = 500$  and  $V^* = 0.05$ . .... 98

Figure 7.8: The placement of the four base structures. .... 99

Figure 7.9: The effect on  $\tau$  when moving the bases symmetrically in the domain. .... 100

Figure 7.10: Comparison of all the volume constraints for a three-dimensional domain with a full Dirichlet boundary and four initial bases. .... 101

Figure 7.11: A converged architecture using four initial bases, with  $k^* = 500$  and  $V^* = 0.05$ . .... 102

Figure 7.12: Comparison of the size of the base structure for different values of  $c/L_D$  and  $H/L_D = 0.04$ , for one, two and four bases. .... 103

Figure 7.13: A thermal performance comparison of one, two and four bases with the effect on  $\tau$  for  $V^* = 0.05$  and varying values of  $k^*$  for a three-dimensional domain with full Dirichlet boundary. The graph also shows the % difference between one and two bases and one and four bases. .... 104

Figure A.1: The influence of  $s_0$  and  $\theta_0$  for  $s = 0.3$ . .... I

Figure A.2: The influence of  $s_0$  and  $\theta_0$  for  $s = 0.5$ . .... I

Figure A.3: The influence of  $s_0$  and  $\theta_0$  for  $s = 0.9$ . .... II

Figure A.4: The influence of  $s_0$  and  $\theta_0$  for  $s = 0.999$ . .... II

Figure B.1: Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.1$  for different values of  $k^*$  ..... III

Figure B.2: Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.15$  for different values of  $k^*$  ..... IV

Figure B.3: Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.2$  for different values of  $k^*$  ..... IV

Figure B.4: Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.25$  for different values of  $k^*$  ..... V

Figure B.5: Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.3$  for different values of  $k^*$  ..... V

Figure B.6: The effect of the asymptote parameters on  $\tau$  for a three-dimensional domain with a full Dirichlet boundary. .... VI

Figure B.7: The mesh dependence for a three-dimensional domain using a full Dirichlet boundary with respect to  $\tau$ . .... VIII

Figure B.8: The iteration study for a three-dimensional full Dirichlet boundary with respect to  $\tau$ . .. VIII

Figure B.9: The effect of  $k^*$  on the maximum temperature for a  $V^* = 0.05$  for a three-dimensional domain with a full Dirichlet boundary. .... IX

Figure B.10: Results for  $\tau$  for a full Dirichlet boundary with  $V^* = 0.1$  for different values of  $k^*$ . ..... X

Figure B.11: Results for  $\tau$  for a full Dirichlet boundary with  $V^* = 0.15$  for different values of  $k^*$ . ..... X

Figure B.12: Results for  $\tau$  for a full Dirichlet boundary with  $V^* = 0.2$  for different values of  $k^*$ . ..... XI

Figure B.13: Results for  $\tau$  showing the effect of the size of the bases for a three-dimensional domain with a full Dirichlet boundary. .... XII

Figure B.14: The effect of  $H_b/L_D$  on  $\tau$  for  $c/L_D = 0.16$ . .... XII

Figure B.15: Results for different conductivity ratios and volume constraints for different values of  $c/L_D$  and  $H_b/L_D = 0.04$  for a three-dimensional domain with a full Dirichlet boundary and two initial bases. .... XIII

Figure B.16: The effect of the base size on  $\tau$  for a three-dimensional domain with a full Dirichlet boundary with four initial bases. .... XIV

Figure B.17: Results for different conductivity ratios and volume constraints for different values of  $c/L_D$  and  $H_b/L_D = 0.04$  for a three-dimensional domain with a full Dirichlet boundary and four initial bases. .... XV

## List of Tables

Table 3.1: Coefficients for the general two-dimensional formula for the internal nodes. ....	21
Table 4.1: Nodal temperatures for the example in Versteeg and the finite volume code. ....	34
Table 4.2: Nodal temperature comparison between FVM and OpenFoam.....	34
Table 4.3: Progression of the MMA. ....	35
Table 4.4: Input parameters for determining the dimensionless temperature. ....	35
Table 4.5: Architectures for different values of $T_\infty$ . ....	36
Table 4.6: Architectures for different conductivity ratios with $k_L$ kept constant. ....	36
Table 4.7: Architectures for the same conductivity ratio with different values of $k_H$ and $k_L$ . ....	37
Table 4.8: Architectures for different values of $q_H$ . ....	37
Table 4.9: Architectures for different sizes of the domain.....	38
Table 4.10: Input parameters for the influence of $s$ and $s_0$ .....	39
Table 4.11: Input parameters used for the mesh dependence of the two-dimensional case.....	41
Table 4.12: Architectures for the two-dimensional mesh dependence. ....	42
Table 4.13: Input parameters for validation of code with Gersborg-Hansen <i>et al.</i> .....	42
Table 4.14: Comparison of architectures from this study and the paper by Gersborg-Hansen <i>et al.</i> ...	43
Table 5.1: Coefficients for the general three-dimensional formula for the internal nodes. ....	46
Table 5.2: Input parameter for the effect of the asymptotes for a three-dimensional domain with a partial Dirichlet boundary. ....	48
Table 5.3: Architectures as viewed on the diagonal section plane $D$ for different values of $s$ and $s_0$ with $k^* = 500$ and $V = 0.1$ .....	50
Table 5.4: Input parameters for the mesh dependence for a three-dimensional partial Dirichlet boundary. ....	51
Table 5.5: Architectures for the mesh dependence for $s_0 = 0.15$ . ....	53
Table 5.6: Input parameters for the number of iterations for a three-dimensional partial Dirichlet boundary. ....	54
Table 5.7: Architectures for the iteration-dependence study for $s_0 = 0.15$ .....	56
Table 5.8: Input parameters for effect of asymptotes for a three-dimensional domain with a partial Dirichlet boundary. ....	57
Table 5.9: Input parameters for constant and incremental increasing penalisation for a three-dimensional domain with a partial Dirichlet boundary. ....	59
Table 5.10: Architectures as seen on the diagonal section $D$ for constant penalisation with $s_0 = 0.15$ . 61	
Table 5.11: Architectures as seen on the diagonal section $D$ for increasing penalisation with $s_0 = 0.15$ for a three-dimensional domain with a partial Dirichlet boundary. ....	62
Table 5.12: Input parameters for the random initial density distribution using a three-dimensional domain with a partial Dirichlet boundary. ....	63
Table 5.13: Architectures for an even distribution (1) and random distribution (2, 7-8) as seen on the diagonal slice $D$ for a three-dimensional domain with a partial Dirichlet boundary .....	64
Table 5.14: Input parameters for $p = 1$ initial distribution.....	64

Table 5.15: Architectures for different initial density distributions for a three-dimensional domain with a partial Dirichlet boundary. ....	65
Table 5.16: Converged architectures showing the effect of the base width $c/L_D$ for a three-dimensional domain with a partial Dirichlet boundary. ....	66
Table 5.17: Input parameters for conductivity and volume simulations for a three-dimensional partial Dirichlet boundary. ....	67
Table 5.18: Comparison of the conductivity ratios for $V^* = 0.1$ and $s_0 = 0.15$ for a three-dimensional domain with a partial Dirichlet boundary. ....	67
Table 5.19: Architectures for the different volume constraints at $k^* = 500$ and $s_0 = 0.15$ for a three-dimensional domain with a partial Dirichlet boundary. ....	69
Table 6.1: Input parameters for the effect of the base size for a three-dimensional full Dirichlet boundary. ....	73
Table 6.2: Comparison of architectures for different heights using $k^* = 500$ , $c/L_D = 0.24$ and $V^* = 0.05$ for a three-dimensional full Dirichlet boundary. ....	75
Table 6.3: Comparison of architectures for different values of $c/L_D$ using $k^* = 500$ , $H_b/L_D = 0.04$ and $V^* = 0.05$ for a three-dimensional full Dirichlet boundary. ....	76
Table 6.4: Architectures for MMA optimised architectures and solid blocks ....	78
Table 6.5: The input parameters for the effect of the asymptotes. ....	79
Table 6.6: Input parameters for the mesh dependence of a three-dimensional domain with a full Dirichlet boundary. ....	80
Table 6.7: Converged architectures as seen on the diagonal section plane $D$ for the mesh dependence with $s_0 = 0.15$ . ....	80
Table 6.8: Input parameters for the iteration dependence of a three-dimensional full Dirichlet boundary. ....	82
Table 6.9: Converged architectures as seen on the diagonal section plane $D$ for the iteration-dependence. ....	83
Table 6.10: Input parameters for the placement of the base structure for a three-dimensional domain with a full Dirichlet boundary. ....	84
Table 6.11: Input parameters for practical boundary condition for a three-dimensional domain with a full Dirichlet boundary. ....	88
Table 6.12: Comparison of the conductivity ratio for $V^* = 0.1$ and $s_0 = 0.15$ for a three-dimensional full Dirichlet boundary ( $c/L_D = 0.24$ , $H_b/L_D = 0.04$ , $x_l/L_D = 0.0$ and $y_l/L_D = 0.0$ ). ....	89
Table 6.13: Architectures for the different volume constraints at $k^* = 500$ and $s_0 = 0.15$ for a three-dimensional domain with a full Dirichlet boundary and a partial Dirichlet boundary. ....	90
Table 7.1: Input parameters for the size of the two base structures for a three-dimensional domain with a full Dirichlet boundary. ....	94
Table 7.2: Input parameters for the placement of the two base structures. ....	95
Table 7.3: Input parameters for the simulations of varying $k^*$ and $V^*$ . ....	97
Table 7.4: Input parameters for the size of the base structure, using four bases. ....	99
Table 7.5: Input parameters for the placement of the base structures, using four initial bases. ....	100
Table 7.6: Input parameters for different conductivity ratios and volume constraints for a three-dimensional domain with a full Dirichlet boundary, using four initial bases. ....	101
Table B.1: Architectures for different values of $s$ and $s_0$ . ....	VII



# Nomenclature

## Latin Symbols

Symbol	Description	Unit
$a$	FVM coefficient	W/K
$A$	Area	$m^2$
$\mathbf{b}$	Source term vector	W
$c$	Length of cold boundary or fixed base width	m
$CV$	Control volume	$m^3$
$D$	Diagonal section	-
$f$	Function	-
$g_0$	Objective function	K
$g_1$	Constraint function	-
$\bar{g}$	Adjoint equation	-
$\hat{g}$	Constraint limit	-
$H_D$	Height of domain	m
$H_b$	Height of fixed sub-domain	m
$i$	Element counter or number of element faces	-
$I$	Number of MMA iterations	-
$j$	Number of constraints in MMA	-
$k$	Thermal conductivity	W/(mK)
$k^*$	Conductivity ratio	-
$K$	Conductivity matrix	W/K
$L$	Lower MMA asymptote	-
$L_D$	Length of domain	m
$m$	Number of constraint functions	-
$M$	Number of nodes for $x$ -, $y$ - or $z$ -direction	-
$M_\Omega$	Total number of nodes in the domain	-
$n$	Normal direction or vector	-
$N$	Neighbour element	-
$P$	Current element	-
$p$	Penalisation factor	-
$p_f$	Final penalisation factor	-
$p_{inc}$	Penalisation factor increment	-
$O$	Origin	-
$q''$	Heat flux	W/ $m^2$
$q'''$	Heat generation density	W/ $m^3$
$Q$	MMA sub-problem	-
$s$	MMA moving asymptote parameter	-
$s_0$	MMA fixed asymptote parameter	-
$S$	Source term	W
$S_u$	Load vector source term	W
$S_p$	Current element source term	W

$\bar{S}$	Source term	W
$t$	Time	s
$T$	Temperature	K
$\mathbf{T}$	Temperature vector	K
$u$	Velocity	m/s
$U$	Upper MMA asymptote	-
$V$	Volume	m <sup>3</sup>
$V^*$	Volume constraint	-
$V_f$	Volume ratio obtained from MMA	-
$W_D$	Width of domain	m
$x$	$x$ -direction	-
$y$	$y$ -direction	-
$z$	$z$ -direction	-

### Greek Symbols

Symbol	Description	Unit
$\alpha_l$	Harmonic mean distance coefficient	-
$\Delta$	Delta / Change in	-
$\Gamma$	Diffusion coefficient	W/(mK)
$\delta$	Distance	m
$\partial$	Partial derivative	-
$\lambda$	Adjoint method vector	-
$\nabla$	Gradient operator	-
$\iota$	Interface between two elements	-
$\phi$	Place holder	-
$\rho$	Density	kg/m <sup>3</sup>
$\theta$	Element density vector	-
$\theta$	Element density / matrix	-
$\theta_0$	Initial density distribution	-
$\bar{\theta}$	Maximum density	-
$\underline{\theta}$	Minimum density	-
$\tau$	Dimensionless temperature measure	-
$\gamma$	Dual objective function solution for MMA	-

### Subscripts

Symbol	Description	Unit
1	Distance from origin to the 1 <sup>st</sup> base	m
2	Distance from origin to the 2 <sup>nd</sup> base	m
3	Distance from origin to the 3 <sup>rd</sup> base	m
4	Distance from origin to the 4 <sup>th</sup> base	m
$b$	Bottom element	-
$e$	East element	-
$f$	Face of the element	-
$H$	Highest value	-
$i$	Element counter	-
$\iota$	Interface of element and neighbour	-
$\iota P$	From $\iota$ to $P$	-
$j$	Number of constraints in MMA	-

$L$	Lowest value	-
$max$	Maximum	-
$n$	North element	-
$N$	Neighbour element	-
$Ni$	From $N$ to $i$	-
$\phi$	Place holder	-
$P$	Current element	-
$PN$	From $P$ to $N$	-
$s$	South element	-
$t$	Top element	-
$w$	West element	-
$x$	$x$ -direction	-
$y$	$y$ -direction	-
$z$	$z$ -direction	-
$\infty$	Boundary temperature	-

### Superscripts

Symbol	Description	Unit
$I$	MMA iteration	-
$T$	Transpose	-
$-1$	Inverse of matrix	-
$'$	Derivative	-

# Chapter 1      **Introduction**

## **1.1 Background**

Heat transfer as a method of energy transfer is present in almost all aspects of life. According to the first law of thermodynamics, the change in the internal energy of a system is equal to the difference between the heat added to a system and the work done by that system [1]. Heat generation is thus a direct effect of work being done.

The effects of heat generation can be seen in both natural and man-made objects. Humans for example, as warm-blooded mammals, start to sweat and our blood comes to the surface of our skin to dispel heat when it is hot. When it is cold, our blood moves to the core of the body to limit the amount of energy we use. This process of keeping the body at the right temperature is known as thermoregulation [2]. Dogs use panting to dispel heat since they can only perspire through their toes.

The rate at which heat is extracted from a given system is one of the limiting factors in the design of man-made systems. In automobiles, the heat generated by the engine is cooled by a liquid, usually water, which is circulated through the engine. The water is then cooled using a radiator and recirculated. Electronic systems also generate heat, which is often extracted by a combination of fans and heat sinks.

The heat generated in a system can be extracted by various methods, given below [3]:

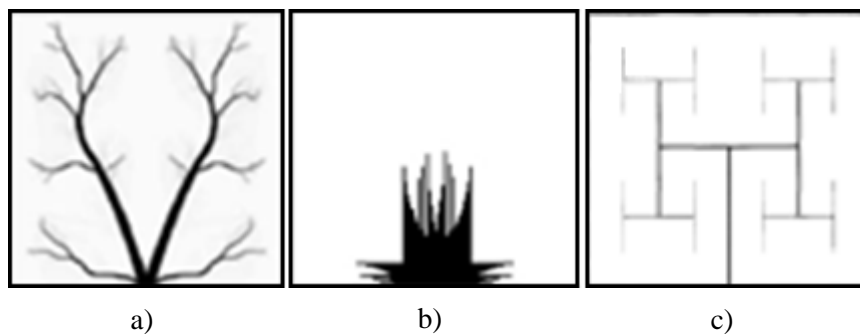
- *Conduction:* Conduction is the transfer of energy between adjacent particles as a result of interactions between the particles. In gases, conduction is caused by the collisions and diffusion of molecules during random motion. In solids, conduction is caused by the vibrations of the molecules and the energy transport of free electrons.
- *Convection:* Convection is a mode of energy transfer between a solid and an adjacent liquid or gas. Convection can either be forced, where the fluid or gas is forced to flow over the solid surface through the use of a pump or fan; or natural, where the fluid motion is caused by buoyancy forces.
- *Radiation:* Radiation (specifically thermal radiation) is the energy emitted by matter in the form of electromagnetic waves as a result of the changes in electronic configurations of the atoms or molecules.

Conduction and convection are the main methods used to transfer energy in heat exchangers. Forced convection occurs inside the pipes where the fluid flows, conduction occurs in the solid material of the pipe and natural convection occurs between the pipe and the surroundings. Conduction and convection can also be used to cool electronics.

One of the major limiting factors in the power density (the amount of power per unit volume) of today's electronics is the heat generated by the device [4]. In personal and server computers, the central processing unit (CPU) is cooled by a heat sink (usually made from copper and aluminium),

which is, in turn, cooled by a fan, thus utilising both conduction and convection. This is, however, not possible in smaller-scaled electronics, where the space is limited and conduction (passive cooling) is the only form of heat transfer available. Previous work done at the University of Pretoria considered heat extraction from low thermal conductive power electronic modules in order to increase the effective power density [5 - 7].

The limited space problem has sparked numerous papers in recent years with the goal of minimising the thermal resistance in a small volume [8 - 10]. The material layout in a volume can be optimised with the goal of maximising heat removal, through changing the thermal conductivity of the material in order to create a path/s for the heat to flow along more efficiently. This optimised material layout is referred to as the architecture. Optimised architectures can be achieved by numerous methods including but not limited to solid isotropic material with penalisation (SIMP), constructal theory, bionic optimisation and evolutionary structural optimisation (ESO). Some examples of converged two-dimensional architectures are shown in Figure 1.1, where different optimisation approaches were followed to obtain suitable distributions of high-conductive thermal material embedded in a heat-producing domain.



**Figure 1.1:** Optimised architectures for heat transfer using a) SIMP [8] b) bionic optimisation [11] c) constructal theory [12].

Darker regions represent high-conductive thermal paths that improve heat transfer from the internal white regions (low-conductive material) to a predefined uniform temperature section on the domain edge. It should be noted that different optimisation approaches may result in significantly dissimilar internal architectures.

## 1.2 Problem Statement and Purpose of Study

The area-to-point or volume-to-point and volume-to-surface thermal optimisation problem is considered in this study. Topology optimisation, specifically the method of moving asymptotes, is used to determine and study optimised internal architectures in a cubic three-dimensional domain. Up to now mostly two-dimensional cases have been considered and little is known about optimised three-dimensional internal architectures. Bejan and co-workers [13 - 14] and Feng *et al.* [15] did consider three dimensional cases, but mostly cylindrical and conical type domain were considered using constructal theory. For the rectangular and cubic domains, few results are available and is difficult to generalise, thus further enforcing the need for this study.

### 1.3 Dissertation Methodology

This study has the following structure:

- In Chapter 2 a literature study will be conducted to give the reader an overview of the different methods of topology optimisation and some of the numerical instabilities that are usually encountered.
- In Chapter 3 the adopted numerical model for this study will be discussed in full for all stages of the optimisation process.
- In Chapter 4 a two-dimensional investigation will be done for a partial Dirichlet boundary to serve as a validation of the numerical method used in this study via comparison with other studies already conducted. The two-dimensional case will form the foundation for the three-dimensional case, which is the main outcome of this study.
- Chapter 5 will discuss the implementation of the three-dimensional domain and a three-dimensional investigation will be done for a partial Dirichlet boundary in preparation for the full Dirichlet boundary case.
- In Chapter 6 a full Dirichlet boundary will be considered in three dimensions with a single optimised cooling structure.
- In Chapter 7 multiple cooling structure optimisations will be considered and compared with a single cooling structure.
- Chapter 8 will discuss the conclusions that may be drawn and the recommendations that may be made for this study.

## Chapter 2 Literature

### 2.1 Introduction

The purpose of the literature study is to give the reader an overview of topology optimisation and the different topological optimisation methods that can be used, in preparation for the rest of the study. This chapter will discuss the basic process of topology optimisation as well the different methods, some noted papers in the field, as well as the difficulties that can arise when using topology optimisation.

### 2.2 Geometric Optimisation

Optimisation in simple terms is the process of finding the best result for certain prescribed conditions, subject to given constraints. Since the optimisation problem can either be to minimise the effort or maximise the benefit, optimisation can best be described as the process of finding the value which gives the minimum or the maximum of the objective function  $g_0(\theta)$ , where  $\theta$  is a design variable (degree of freedom), for certain prescribed conditions.

Geometrical optimisation can be divided into three categories, namely size, shape and topology optimisation. Size optimisation is the process of finding the optimal designs, for example, a load-bearing truss, by changing the size of the cross-section (variables). Shape optimisation is mainly performed on continuum structures by modifying the predetermined boundaries to achieve the optimal designs. Topology optimisation is the search for the optimal order and placement of a limited amount of material to achieve predetermined goals. In continuum structures, topology optimisation is the method used to find the optimal design by determining the best locations and geometries of cavities in the design domain.

Until recently, topology optimisation was mostly used to successfully obtain improved load-bearing structures [16 - 17]. An example is shown in Figure 2.1, where the material distribution was optimised in order to minimise deflection. The methods employed in topology optimisation are not limited to structures though, and can also be applied to other disciplines like heat transfer [18] and fluid flow. For example, by using topology optimisation, an optimised path for a fluid particle/stream in a microchannel or the optimum distribution of heat-conducting material can be obtained.



**Figure 2.1:** Material distribution obtained for a load-bearing structure using topology optimisation [19].

The gains in thermal optimisation are endless and are used in this study to obtain improved three-dimensional conductive structures in a solid-state cube. With regard to the topic of this study, topology optimisation can be used to decrease the internal maximum temperature of electrical components via conductive cooling using high-conductive pathways. Reducing the maximum temperature will allow engineers to increase the power density of, for example, processors in computers.

## 2.3 Methods of Topology Optimisation

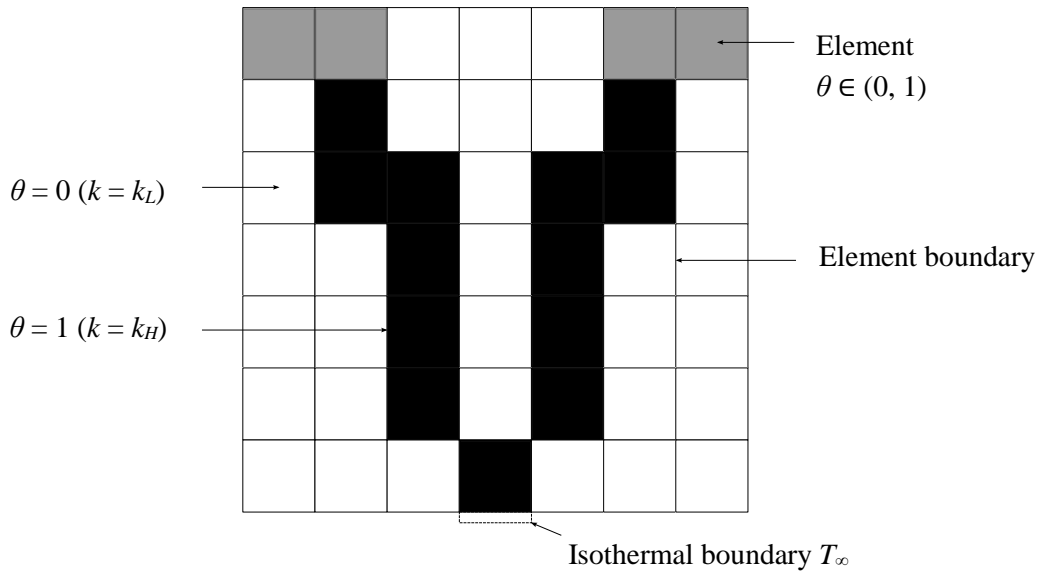
The basic steps in any topology optimisation process are as follows:

- *Numerical Modelling*: This involves creating a mesh, discretising the governing equations, and in this case, solving the temperature distribution.
- *Sensitivity analysis*: This involves determining the derivative of the temperature distribution according to the design variables. The sensitivity analysis is a crucial step in most optimisation routines.
- *Optimisation algorithm*: Using the sensitivity analysis, the optimal direction for the conductive paths can be placed using an optimisation algorithm, for example, MMA or sequential linear programming (SLP). Once the optimal direction has been found, a topological method is used to determine what should happen with the material's conductivity distribution. There are numerous optimisation methods available, each with its own advantages and disadvantages.
- *Filtering*: Some methods of topology optimisation produce a checkerboard pattern, which is discussed in Section 2.4.1. Filtering techniques are applied to prevent the appearance of checkerboard patterns.
- *Post-processing*: The optimised architectures obtained using topology optimisation must be post-processed in order to display the three-dimensional solutions better on paper. An example of this is showing the structure through a diagonal slice, or adding shadows to the three-dimensional architectures to make it easier to visualise.

Topology optimisation methods can be divided into continuous and discrete design variable methods. Continuous methods penalise the design variables to either a 0 (white substrate) or 1 (black high-conductive material) solution, see Figure 1.1. It does not always guarantee that there will be no intermediate densities (grey areas). These penalisation methods include SIMP and homogenisation.

In the discrete methods, the method automatically assumes 0-1 values. This means that the method forces the material density to be either 0 or 1. These methods include evolutionary structural optimisation, bi-directional evolutionary structural optimisation (BESO) and cellular automaton. There are other methods which are difficult to categorise, namely constructal theory, fractal theory, level set method and bionic optimisation. These methods and some noted papers will be discussed in Sections 2.3.1 - 2.3.3.





**Figure 2.2:** Typical topological domain showing the conductivity of elements, element boundaries and the isothermal boundary.

Figure 2.2 shows a typical topological domain. The domain is divided into elements, each of which is a design variable. Each element has a density,  $\theta$ , which describes the conductivity and the heat generation rate of that element. This density can range between 0 and 1. If the density is 0, the conductivity,  $k$ , is equal to  $k_L$ , the lowest conductivity value in the domain and also known as the substrate, shown as the white element in the figure. If the density is equal to 1, the conductivity is equal to  $k_H$ , the highest conductivity in the domain, shown as the black element in the figure. The ratio between these values is known as the conductivity ratio,  $k^* = k_H/k_L$ . In the case of continuous design variable methods, the density can also be any value between 0 and 1, shown as the grey element in the figure, which means the conductivity of that element is between  $k_L$  and  $k_H$ . Elements are connected at their boundaries, as shown in the figure. The conductivity value at this boundary must be calculated using an average, which will be discussed in Section 3.2.2.

In most topology optimisation cases where conduction is used, a small isothermal boundary is used. The heat generated in the domain is extracted through this isothermal boundary and guided there by the material distribution of high-conductive elements. The heat generation rate of each element is also dependent on the density of the element. If  $\theta = 0$ , the heat generation rate is equal to  $q_H'''$ , the selected rate of the problem. If  $\theta = 1$ , the heat generation is equal to 0.

## 2.3.1 Continuous Design Variable Methods

### 2.3.1.1 Solid Isotropic Material with Penalisation Method

In this method, the topology optimisation problem is formulated as a discrete valued design problem, also known as a 0-1 problem. The SIMP method is known as a continuous design variable method, where the relative densities (0-1) of elements are treated as the design variables. It is well known that this kind of problem suffers from a lack of effective solutions [20]. This can be due to a lack of convergence and is also highly dependent on the number of design variables.

Some optimisation algorithms produce grey areas, which do not conform to the 0-1 design. To compensate for this, material models are used to force the topology design towards the limiting values

of 0 (substrate material) or 1 (high-conductive material). The SIMP method was successfully implemented by Zhang and Liu in their paper [9] using Eq (2.1):

$$k(\theta) = k_L + \theta^p(k_H - k_L) \quad (2.1)$$

where  $p$  is the penalisation factor and  $\theta$  is the element density, which has an effect on the penalising of intermediate densities. Zhang and Liu used  $p = 2$  in their study. To obtain solutions of only black and white elements ( $\theta = 0$  or  $\theta = 1$ ), the power  $p$  should have a fairly large value of 3 or above [19]. Black and white solutions are preferred because a grey solution,  $\theta \in (0; 1)$ , would mean a mixture of high-conductive material and low-conductive material, essentially a composite, which is difficult and expensive to manufacture. The problem with a high value of  $p$  is that it will in some cases, depending on initial design, lead to designs where  $\theta$  has converged to a local minima and thus only a suboptimal distribution is obtained. To counteract this, it is advised to use an increasing penalisation scheme, which means that the power  $p$  is raised from 1 (no penalisation) to the final value in equal increments during the optimisation procedure until the final design is found. This scheme does not guarantee a 0-1 design, but it works in most cases if combined with the filtering of sensitivities [19].

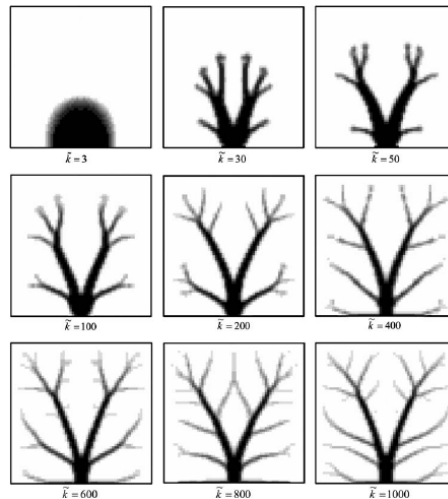
An unfortunate disadvantage of these penalisation techniques is the onset of checkerboards, better described as rapid oscillations in the element density of the material in the domain, which is possible since topological methods are iterative and the density of each element can change after each iteration. Checkerboards are discussed in detail in Section 2.4.1 and the methods used to counteract/account for these checkerboards are discussed in Section 2.4.2.

Zhang and Liu [9] investigated the two-dimensional design of conducting paths to solve the volume-to-point (VP) heat conduction problem using topology optimisation and the SIMP method. The VP problem had an adiabatic boundary except for a small isothermal cold spot. The authors calculated the best distribution of high-conductive material in a domain such that the highest (hotspot) temperature is minimised while also adhering to a specific volume constraint. The volume constraint limits the number of solid elements in the final solution. However, the position of the hot spot changes as the material distribution changes, thus the highest temperature, as a function of material distribution, is non-continuous in some cases. It must be noted that as long as the temperature distribution is a continuous function of the density distribution, the maximum temperature will be continuous as well, regardless of where the hot spot occurs. This non-continuity can make the optimisation process difficult. An alternative is to use the dissipation of heat transport potential capacity as the objective function.

The authors used the finite element method (FEM) and quadrilateral elements to discretise the domain. To solve the optimisation problem, a sequential quadratic programming method (which is a gradient-based method) was used and to force the optimisation to a discrete 0-1 problem, the SIMP method was used. The authors used the filtering technique by Bendsøe and Sigmund to prevent the onset of checkerboards [20].

The authors concluded that the optimal path obtained by the methods was similar to that of a natural tree. It was found that the routine used had performed much better compared with bionic optimisation and the tree-like network constructal method. In the case of uniform heat sources, heat transfer performance was increased by up to 30%. A performance increase of 38% was obtained using non-uniform heat sources. It was also noted that the configuration of the optimal paths with different conductivity ratios corresponded to the configuration of a tree in different stages of growing. With the increase of the ratio of high-conductive material to low-conductive material, the number of branches of the tree increased. This is illustrated in Figure 2.3.

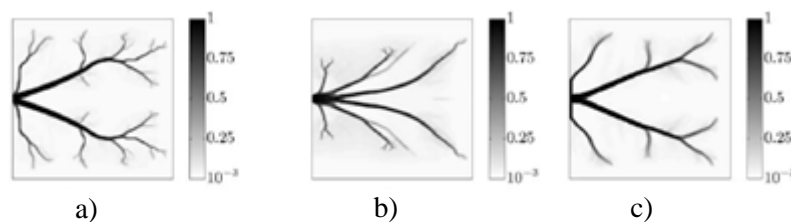
Another study concerning two-dimensional conduction in heat transfer was conducted by Gersborg-Hansen *et al.* [8]. The purpose was to optimally place a limited amount of highly conductive material in a design space that will optimise their objective function and boundary conditions. They considered more than one objective function type and with different boundary cases. It was noted that when using harmonic averages to calculate the conductivity at the element boundaries, checkerboards did not form during the topology optimisation process.



**Figure 2.3:** Material distribution at different thermal conductivity ratios [9].

The authors used the finite volume method (FVM) to discretise the energy equation, but used both FVM and FEM for the calculation of the cost function. To calculate the sensitivities of the objective function, the authors used the adjoint method, which is discussed in Section 3.3. The authors used the SIMP method to force the solution towards a 0-1 solution. To solve the optimisation problem, the authors used the method of moving asymptotes, by Svanberg as discussed in Section 3.4.

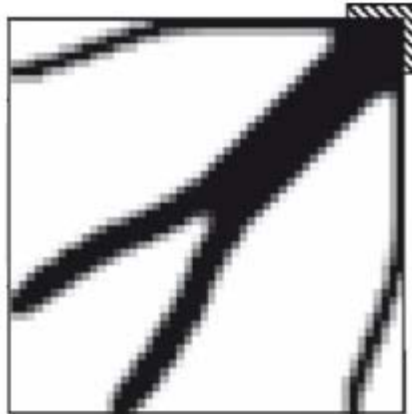
To account for the checkerboards, the authors used the filter method by Sigmund and Bendsøe to control the checkerboards [20]. Some example solutions are shown in Figure 2.4. In this problem, the walls of the domain are adiabatic except for the cold spot at the base of the tree structure. The authors concluded that thermal topology optimisation is possible using the finite volume method for heat conduction and that it only requires minor deviations in the sensitivity analysis when compared with the finite element method.



**Figure 2.4:** The results for a 128x128 grid with a) FEM with a filter b) FVM with a filter c) FVM with no filter [8].

### 2.3.1.2 Homogenisation

The homogenisation method introduces micro-perforated composites as admissible designs to relax the originally ill-posed 0-1 problem [21]. Iga *et al.* [22] investigated the influence of a design-dependent effect upon heat convection, conduction and internal heat generation for the optimal designs developed using the homogenisation scheme in two dimensions. In their study, the authors developed a structural optimisation method for the design of thermal conductors that aim to maximise the temperature diffusivity. An example of a converged architecture is showed in Figure 2.5. Haslinger *et al.* [23] investigated the application of the homogenisation scheme to heat conductivity problems using two isotropic materials.



**Figure 2.5:** Optimal configuration obtained using an isothermal boundary in the corner of the domain, using homogenisation [22].

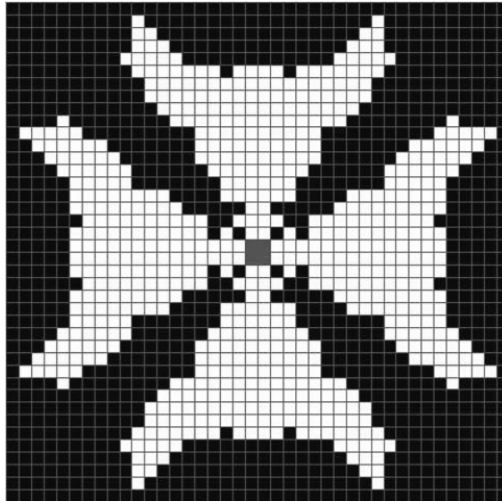
## 2.3.2 Discrete Design Variable Methods

### 2.3.2.1 Evolutionary Structural Optimisation

ESO is the simple concept of gradually removing inefficient material from a structure. This process will iteratively evolve the structure to a more optimal shape and topology. Although, theoretically, this process cannot always guarantee the optimal solution, it still provides a useful way to explore new shapes in the conceptual phase of design. ESO is one of the most popular techniques used for topology optimisation. The method was first proposed by Xie and Steven in the early 1990s [21]. The method was first used for structural optimisation, but has been further developed to other applications such as heat transfer.

The removal of elements is done by the concept of rejection criterion (e.g. temperature or heat flux), where elements are removed if the integral level of the element is less than the rejection ratio (ratio that defines if an element is removed) multiplied by the integral level of the highest element. This process is repeated until an ESO steady state is reached, which means that there are no more elements that can be removed at the current rejection ratio. When ESO steady state is reached, an evolution rate is introduced and added to the rejection ratio. The process is repeated until a new ESO steady state is reached. The evolution rate is set quite small, between 0.1% and 1% so that the elements removed can be controlled within a small value. This procedure can, for instance, be used to find as close to an even distribution of temperature or heat flux as possible in the domain.

Li *et al.* [24] investigated the extension of ESO to steady-state two-dimensional heat conduction. The authors used fixed temperatures on all four boundaries of a square domain, with certain areas in the domain that are fixed in terms of the density. The authors concluded that the ESO method can easily be implemented in heat conduction problems. Li *et al.* [25] also investigated evolutionary topology optimisation with the purpose of reducing the temperature in heat-conducting fields. The authors investigated heat sources in a square domain using ESO. The authors concluded that the application of ESO in heat conduction is easy and does not increase the computational complexity. An example of a converged architecture is shown in Figure 2.6.

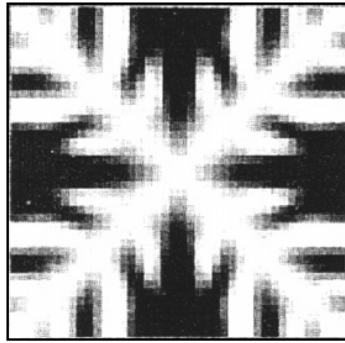


**Figure 2.6:** Optimal geometry obtained with all outer boundaries at a constant temperature, with four internal heat sources, using ESO [25].

### 2.3.2.2 Bi-directional Evolutionary Structural Optimisation

The BESO method is an extension of the ESO method. The main difference between ESO and BESO is that with BESO, material can be added and removed simultaneously. The first research on BESO was done by Yang *et al.* [21] for stiffness optimisation. In the study, the sensitivity of the void elements is determined using linear extrapolation of the displacement field. After this step, the solid elements with the lowest sensitivity are removed and the void elements with the highest sensitivity are changed into solid elements. This adding and removal of elements are determined by the rejection ratio and the inclusion ratio (ratio which defines if an element is added), which are unrelated.

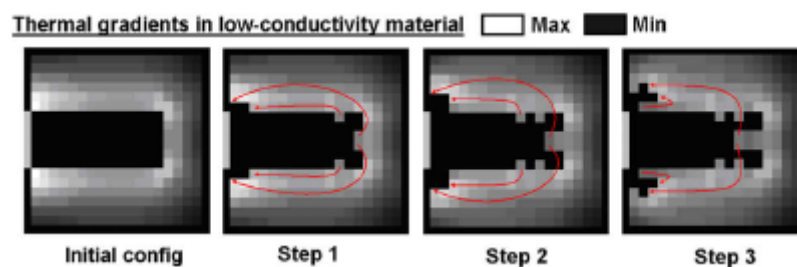
Gao *et al.* [26] investigated two-dimensional topology optimisation of heat conduction problems under design-dependent and design-independent loads using a modified bi-directional evolutionary structural optimisation scheme. The authors tested uniformly heated square domains as well as point loads. The authors concluded that their modified BESO scheme had potential in the layout of design electronic components and it was effective in the dissipation of heat generated using conducting paths. An example of a converged architecture is shown in Figure 2.7.



**Figure 2.7:** Optimal material distribution obtained with all boundaries at a constant temperature and uniform internal heat generation, using BESO [26].

### 2.3.2.3 Cellular Automaton

Cellular automaton starts with an initial shape of a certain quantity of, for instance, high-conductive material. By shaping the initial shape with the use of thermal gradients, the shape evolves step by step until it converges. The number of high conductive cells is kept constant during the optimisation process, by redistributing the high-conductive material to different locations. Thus, no new cells are generated and the volume ratio of low-conductive material to high-conductive material is kept constant. This is illustrated in Figure 2.8, where thermal gradients in the low-conductive material are used to determine the best redistribution strategy. Thus, high-conductive elements are moved from areas with low thermal gradients to areas with high thermal gradients.



**Figure 2.8:** The cellular automaton procedure [27].

Cellular automaton was implemented for two-dimensional conduction by Boichot *et al.* [27]. The authors investigated cellular automaton by effectively cooling a heat-generating surface by arranging the configuration of high-conductive material links which discharged the generated heat to a heat sink. For an initial shape, they used conducting paths optimised by constructal theory. The authors noted that the final shape is a multi-scale tree-like network. They also concluded that the approach may not always decrease the maximum temperature, but for all cases tested, the results generated by cellular automaton are at least as effective as the analytical constructal theory.

## 2.3.3 Other Methods

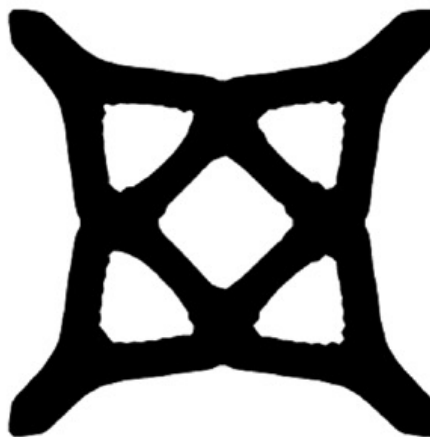
### 2.3.3.1 Level Set Method

The level set method is an implicit representational approach of deformed surfaces and it is an efficient numerical technique to capture the propagating interfaces in topology optimisation. The basic idea of the level set method is representing the boundary by a level set model and evolving the

boundary using an appropriate velocity field. The velocity field is constructed using the sensitivity analysis. As the geometric changes are implemented, the level set function also changes.

Zhuang *et al.* [18] investigated two-dimensional heat conduction using a level set method for multiple load cases. The authors used the level set method to implicitly represent the boundary of the conductive material. By using numerical examples, they concluded that the approach is effective in the topology optimisation of a heat conduction problem.

Ha *et al.* [28] also implemented the level set method to implicitly represent the thermal boundaries in a heat conduction problem. The authors used the Hamilton-Jacobi-type equation to set up the level set method. The authors concluded that the method had no numerical instabilities, delivered similar optimal shapes compared with the density approach and was easy to implement. An example of a converged architecture using the level set method is shown in Figure 2.9.

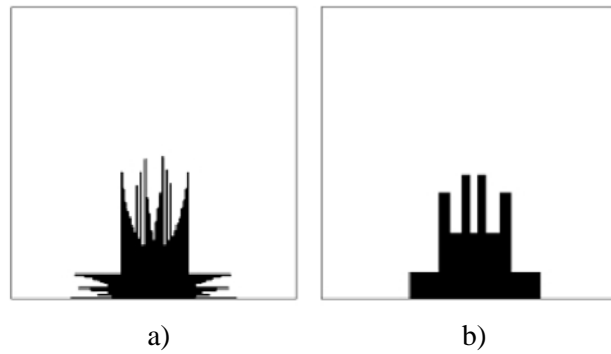


**Figure 2.9:** Optimal geometry obtained for multiple load cases, using the level set method [18].

### 2.3.3.2 Bionic Optimisation

In bionic optimisation, the optimal constructs of the high-conductive material are obtained by numerically simulating the evolution and degeneration process according to the uniformity principle of the temperature gradient [11]. This is similar to the approach used in BESO. In this method, the high-conductive material is regarded as being alive. The process is divided into two components, generation and degeneration. In the generation process, high-conductive material is evolved at the point in the domain with the largest temperature gradient. It follows then that the degeneration process will remove high-conductive material from the points in the domain with the lowest temperature gradient.

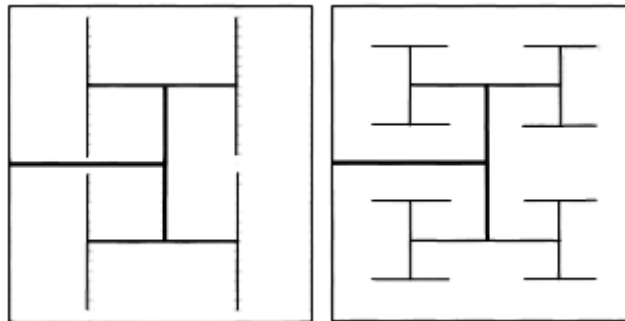
Cheng [11] implemented bionic optimisation in the construction of highly effective heat conduction paths. The author used a square two-dimensional domain with adiabatic walls, a cold spot on the bottom boundary and a uniform heat source throughout the domain. If the thermal conductivity ratio is high, the construct obtained is similar to a tree-like network obtained by constructal theory. The author regularised the solution to make it easier to use in an engineering application as shown in Figure 2.10 and noted that the regularised solution had similar performance to the non-regularised solution. The author concluded that bionic optimisation was able to find the optimal solution for a heat conduction problem.



**Figure 2.10:** a) optimal solution obtained by bionic optimisation b) regularised solution for use in engineering application [11].

### 2.3.3.3 Constructal Theory

The constructal law states that, “For a finite-size (flow) system to persist in time (to live), its configuration must evolve such that it provides easier and easier access to its currents” [29]. Constructal theory has been implemented differently by different researchers at different times. Below is a description of how constructal theory has been implemented for a conduction problem. Constructal theory starts up with an optimal element, and then assembles the element to a larger one, which is called the first assembly. The second assembly is assembled by the optimal first assembly as a basic element. This process is repeated until the solution is optimal. An example of a solution obtained with constructal theory is shown in Figure 2.11. The solution on the left shows the complete structure of the fourth assembly, while the figure on the right shows the optimised network by higher-order assemblies.



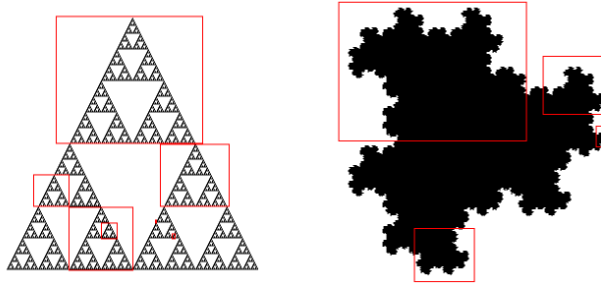
**Figure 2.11:** Completed structures obtained by constructal theory [12].

Some of the first works in the heat conduction volume-to-point problem were done by Bejan and co-workers [30–35] for rectangular and other geometrically shaped bodies using the constructal theory approach. Bejan [12] investigated conducting paths for the cooling of a heat-generating volume while keeping the volume of the high-conductive material constant, using constructal theory. The total heat generation in the domain is also fixed. The author concluded that although the paper was purely theoretical, it deserves attention and could help with the cooling of structures using conducting paths. Bejan *et al.* [36] also published a paper on constructal theory of heat trees at micro- and nanoscales. The authors concluded that from the elemental level to the second-construct level, the high-conductivity regions form tree-like structures.



### 2.3.3.4 Fractal Theory

A fractal is defined as an object that is invariant at any scale of magnification or reduction [37]. The basic idea of fractal theory is the process of using a simple shape or process to solve a complex problem by iterating the simple process. This process is repeated until the limit of the fractal is reached. Almost all fractals are self-similar, which means that the final iteration of the fractal is identical to the smaller parts of the fractal. This is better explained in Figure 2.12.



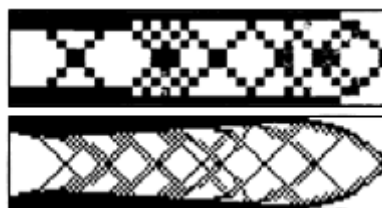
**Figure 2.12:** Fractal theory illustrated in both shapes, where the final fractal is just an enlarged version of the smallest fractal [38].

Wang *et al.* [39] investigated the characteristic size of fractal porous media using heat conduction. Redner investigated fractal and multifractal scaling of electrical conduction in random resistor networks [37]. Wang *et al.* [40] analysed the thermal conductivity of fractal porous media using the finite volume method.

## 2.4 Numerical Instabilities

### 2.4.1 Checkerboards

The checkerboard problem is illustrated in Figure 2.13 and is best explained by the alternation of solid and void elements, sometimes resembling a checkerboard. It was believed that checkerboards represent some optimal microstructure, but it was later shown that checkerboarding is due to unsuitable numerical modelling [20]. Sigmund and Peterson also investigated mesh dependencies and local minima in their paper [20], which is not discussed here. A typical example of non-convergence is the formation of checkerboards. The SIMP and homogenisation methods are prone to the appearance of checkerboards. Several methods have been produced to help with the appearance of checkerboards, namely smoothing, higher-order elements, patches and filters.



**Figure 2.13:** Checkerboard patterns [19].

If a solution is found with checkerboards, smoothing can be implemented. Smoothing is the process of removing checkerboards with image processing. The problem is that although it provides a solution

with no checkerboards, the underlying problem is ignored. Many commercial packages use smoothing of the output images, and care should be taken to make sure how the program works.

Papers suggest that using higher-order elements, can avoid the onset of checkerboards. Using eight or nine noded elements in the homogenisation method removes most of the checkerboard. However, in the SIMP method, checkerboards are only prevented when using eight or nine noded elements and keeping the penalisation power lower than 2.29 [20]. The major problem with using higher-order elements is the increase in computational time.

To remove checkerboards but still save computational time, the patches technique was proposed. The technique effectively introduces a kind of superelement to the finite element formulation and has, in a practical test, been shown to dampen the appearance of checkerboards [20]. The technique unfortunately does not remove checkerboards entirely using topology optimisation. More filtering techniques are discussed in Section 2.4.2.

In this study, no checkerboards were experienced. A possible reason for this is the implementation of the harmonic mean to calculate the value of the conductivity at the interface of two neighbouring elements. Gersborg-Hansen *et al.* also noted that they did not encounter checkerboards when using a harmonic average to calculate the value of the conductivity at the interface of elements, but did, however, encounter checkerboards when using an arithmetic average [8].

## 2.4.2 Filtering Techniques

### 2.4.2.1 Local Constraint on Gradient of Material Density

As the name implies, this method introduces a constraint on the local density variation [17]. This is written as a point-wise constraint on the derivatives of the function  $\theta$  (element density). It was shown that the method eliminates checkerboards and other numerical anomalies, or rather decreases the occurrence to such an extent that it can be ignored. The main disadvantage of the technique is that it adds  $2M_\Omega$  (where  $M_\Omega$  is the total number of nodes) constraints, which slows down the optimisation procedure considerably. This method has been applied to three-dimensional structural topology problems with success [20].

### 2.4.2.2 Filtering Technique

The filtering technique is based on techniques from image processing, and was first proposed by Bendsoe and Sigmund [20]. This method was suggested to prevent the onset of checkerboards in numerical solutions. This is done by modifying the sensitivities used in each iteration of the optimisation algorithm. The filter makes the design sensitivity of a given element dependent on a weighted average over the element itself and its neighbours. It was shown that the filtering technique gives similar results to the local constraint technique. The technique also does not require significantly more computations, does not add any constraints and is relatively easy to implement. The method can even, in some cases, stabilise convergence, but the main drawback is that it is based on heuristics [20]. This method has been applied to three-dimensional structural topology problems with success [20].

## 2.5 Summary

In conclusion, there are numerous topological methods that can be used to solve the heat conduction problem. In this study, the SIMP method will be implemented together with the adjoint method (to solve the sensitivities of the objective function) and the method of moving asymptotes (to find the

optimal material layout), which will be discussed in the next chapter. The next chapter will discuss the governing equation for the finite volume method, the adjoint method and the method of moving asymptotes, as well as the implementation of these methods for a two-dimensional domain.

## Chapter 3 Two-Dimensional Numerical Model

### 3.1 Introduction

This chapter will discuss the various parts in the two-dimensional thermal numerical model, namely the thermal model (finite volume method), the calculation of the gradients of the objective function (adjoint method) and the optimal layout material densities (method of moving asymptotes). The governing equations of each subprocess of the optimisation routine will be discussed as well as the implementation of these governing equations for a two-dimensional thermal model using a partial Dirichlet boundary.

### 3.2 Domain Discretisation and Thermal Model

#### 3.2.1 Methods

The first step in the optimisation process is to model and discretise the domain and in the case of heat conduction, find the temperature distribution field. There are three basic methods that are available to solve the temperature distribution of the domain numerically namely the finite element method, finite volume method and the finite difference method (FDM).

With the FEM approach, a domain is divided into multiple smaller domains [41]. Each domain is connected to the neighbouring domains using nodes. Each node is specified using an algebraic equation, together forming a matrix of equations. There are different cells that can be used to discretise the domain, such as quadrilateral or triangular cells. In two-dimensional topology optimisation, quadrilateral (four node) cells are mostly used. For three dimensions, triangular or square cells can be used. The main advantage of FEM is that it can easily handle complex geometries, loading conditions and constraints.

With the FVM approach, the domain is divided into a finite number of volumes, with volume-centred nodes. Neighbouring volumes in the domain are connected at their boundaries. The first step in FVM is to integrate the case-specific governing equation over a control volume [42]. This is the key step that distinguishes FVM from FEM and FDM. The advantages of FVM are that it can also easily accommodate any geometry and it is easy to extend the method to three dimensions.

With the FDM approach, the direct definition of a derivative is used [43]. Partial derivatives are replaced by approximations, resulting in one equation per grid node. The domain is also divided into nodes as in FEM. FDM approximates the partial derivatives using central, forward and backward differencing. The advantage of FDM is that it gives excellent accuracy, provided that a sufficient number of nodes is used. The disadvantage of FDM is that it can only handle simple geometries.

For this study, FVM is used because of personal experience, the ease of implementation as well as the ease of extending it to three dimensions.

### 3.2.2 Governing Equations of the Finite Volume Method

To effectively use the optimisation process, the governing equations for heat conduction must be known and fully understood. The derivation of the finite volume method is followed using the textbook by Versteeg and Malalasekera [42]. Consider an arbitrarily volume, starting with the general transport theorem:

$$\frac{\partial(\rho\phi)}{\partial t} + \text{div}(\rho\phi u) = \text{div}(\Gamma_{\phi}\nabla\phi) + S_{\phi} \quad (3.1)$$

$\Gamma_{\phi}$  is the diffusion coefficient in this general formula, which will later be replaced with  $k$ , the thermal conductivity of a material. Assuming steady-state pure diffusion, Eq. (3.1) reduces to:

$$\text{div}(\Gamma_{\phi}\nabla\phi) + S_{\phi} = 0 \quad (3.2)$$

To use the control volume method, Eq. (3.2) must be integrated over the control volume (CV).

$$\int_{CV} \text{div}(\Gamma_{\phi}\text{grad}\phi)dV + \int_{CV} S_{\phi} dV = 0 \quad (3.3)$$

Now using the Gauss divergence theorem, this states that:

$$\int_{CV} \text{div}(\phi)dV = \int_A n \cdot \phi dA \quad (3.4)$$

Where  $\phi$  can be any variable and  $n$  is the unit normal to the surface element area  $A$ . Thus:

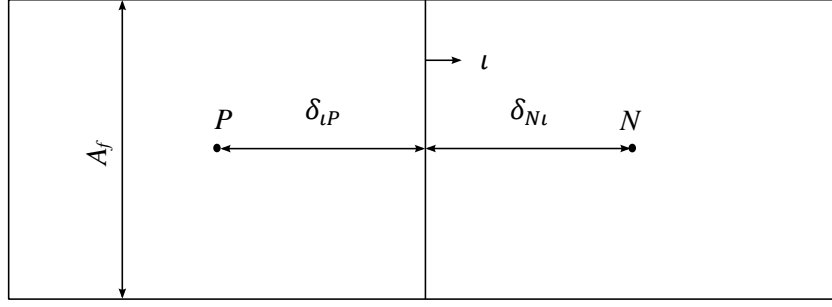
$$\int_A n \cdot (\Gamma_{\phi}\nabla\phi)dA + \int_{CV} S_{\phi} dV = 0 \quad (3.5)$$

Consider a multidimensional case, where the control volume for an element and its neighbour is defined as in Figure 3.1. Now, since this is a specific case, pure diffusion,  $\phi$  is replaced with  $T$  (temperature) and  $\Gamma$  is replaced with  $k$  (conductivity).

Thus from Eq. (3.5), the following holds:

$$\int_{\Delta V} \frac{d}{dn} \left( kA \frac{dT}{dn} \right) dV + \int_{\Delta V} S dV = \sum_1^i \left( kA \frac{dT}{dn} \right)_i + \bar{S}\Delta V = 0 \quad (3.6)$$

Where  $i$  is the number of faces of an element. Eq. (3.6) is easy to interpret in a physical sense since it states that the summation of the flux of all the neighbours entering the volume minus the flux leaving the volume must equal the heat generation, thus a heat balance in the system. In Eq. (3.6), to approximate the value for  $k$  at the interface of the boundary,  $i$ , a harmonic mean [44] is used, where  $N$  denotes the neighbour and  $P$  the current element as shown in Figure 3.1.



**Figure 3.1:** Control volume for a general case for a current and neighbour element.

$$k_l = \frac{k_N k_P}{\alpha_l k_P + (1 - \alpha_l) k_N} \quad (3.7)$$

where

$$\alpha_l = \frac{\delta_{Nl}}{\delta_{Nl} + \delta_{lP}} \quad (3.8)$$

For a structured mesh where all the nodes have the same length,  $\alpha_l$  will always have a value of 0.5. The flux terms can also be approximated using central differencing:

$$\left( kA \frac{dT}{dn} \right)_i \approx k_l A_f \left( \frac{T_N - T_P}{\delta_{PN}} \right) \quad (3.9)$$

Here  $A_f$  is equal to the length of an element,  $\delta$ , times the unit depth  $\delta_z$ , for a square uniformly divided domain. In practical situations, the source term can be a function of the dependent variable, in which case, the source term  $S$  is approximated in a linear form:

$$\bar{S} \Delta V = S_u + S_p T_p \quad (3.10)$$

Substituting Eq. (3.9) - (3.10) into Eq. (3.6) and rearranging, the following is found:

$$T_p \left( \sum_1^i (a_i) - S_p \right) = \sum_1^i (a_i T_i) + S_u \quad (3.11)$$

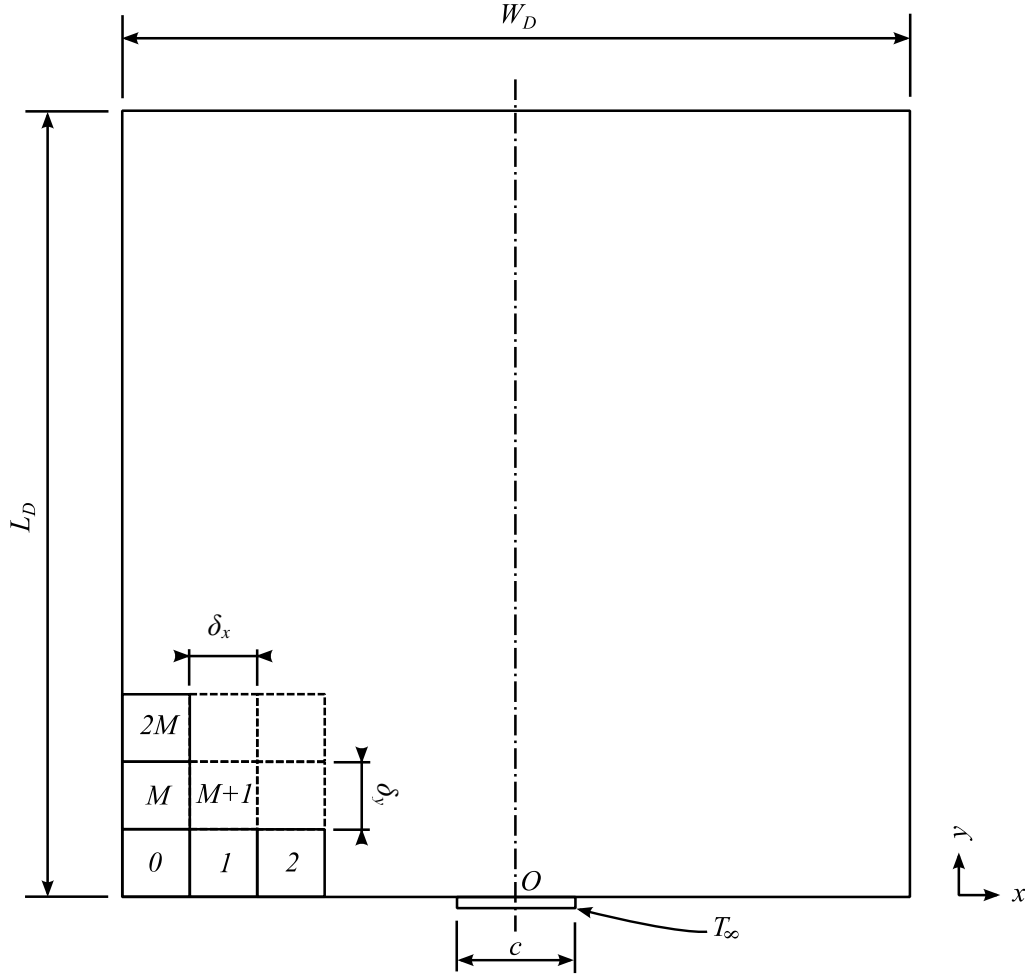
Where  $a_i$  is equal to:

$$a_i = \frac{k_l A_f}{\delta} \quad (3.12)$$

This coefficient  $a_i$  will be different for each face of the element and will depend on the direction of the face. Eq. (3.11) is a general solution for the multidimensional case using FVM for pure diffusion for the internal nodes of a control volume. The attractive feature of the FVM method is the ease of modifying the general equation for two-dimensional and three-dimensional cases.

### 3.2.3 Domain for a Two-Dimensional Thermal Model

In preparing to produce three-dimensional topologies, a two-dimensional code was first considered. Two-dimensional topology optimisation of heat conduction has been done by numerous authors in the scientific community. With this, the self-developed code, the optimisation code and model could be verified. The two-dimensional code was used as the building block of the three-dimensional code, since the third dimension does not add any new equations to the algorithm, only the third dimension.



**Figure 3.2:** Domain for partial Dirichlet boundary located on the bottom edge, for the two-dimensional case.

Figure 3.2 shows the computational domain for the two-dimensional case with the origin located in the centre of the bottom edge. Only a square domain was considered here, requiring that  $L_D = W_D$ . The domain was divided uniformly, such that the number of nodes in the  $x$ -direction,  $M$ , and the number of nodes in the  $y$ -direction,  $M$ , were equal. This also meant that the length and width of the element are equal, thus  $\delta_x = \delta_y = \delta$ . This also means that the face area of an element,  $A_f$  is equal to the length of the element  $\delta$  times unit depth  $\delta_z$ . The elements are numbered from  $i = 0$  at the bottom left corner in horizontal direction to the right edge  $i = M - 1$ , then continues on the left again for the next row of elements. This numbering system is important for the conductivity matrix, discussed in the next section.

An integral part of the topology optimisation is that each element has a density  $\theta$ , which can range between 0 and 1. This density has an effect on the conductivity and internal heat generation of each element and is used to display the final architecture. The conductivity and internal heat generation are defined as follows:

$$k(\theta_i) = k_L + \theta_i(k_H - k_L) \quad (3.13)$$

$$q'''(\theta_i) = q'''_H(1 - \theta_i) \quad (3.14)$$

Eq. (3.13) indicates that the conductivity of the element will be  $k_H$  W/(mK) when the density is 1 and  $k_L$  W/(mK) when the density is 0. The heat generation will be 0 W/m<sup>3</sup> when the density is 1 and  $q_H'''$  W/m<sup>3</sup> when the density is 0, as given in Eq. (3.14). This means that the heat is only generated in the substrate material.

For this problem, all boundaries were adiabatic, except for an isothermal boundary at a temperature  $T_\infty$  having a width of  $c$  located in the centre of the bottom edge. This is known as a partial Dirichlet boundary.

### 3.2.4 Finite Volume Method for a Two-Dimensional Thermal Model

The governing equation for two dimensions for internal elements is easily found from Eq. (3.11) and given in Eq. (3.15).

$$T_p a_p = T_w a_w + T_e a_e + T_n a_n + T_s a_s + S_u \quad (3.15)$$

The equation above is applied to each internal element in the domain. The coefficients for internal elements are given in

Table 3.1.

**Table 3.1:** Coefficients for the general two-dimensional formula for the internal nodes.

$a_w$	$a_e$	$a_n$	$a_s$	$a_p$
$\frac{k_l A_f}{\delta_x}$	$\frac{k_l A_f}{\delta_x}$	$\frac{k_l A_f}{\delta_y}$	$\frac{k_l A_f}{\delta_y}$	$a_w + a_e + a_n + a_s - S_p$

For internal elements,  $S_p = 0$ . For boundary elements, Eq. (3.15) will change according to the specific boundary. The terms that can be influenced are the source terms  $S_p$  and  $S_u$  and in some cases, the coefficient of a boundary element.

*Internal heat generation:* Internal heat generation applies to the whole volume, internal nodes and boundary nodes. Thus all nodes will have the following term:

$$S_u = q'''V \quad (3.16)$$

For the two-dimensional domain, the volume is equal to:

$$V = \delta_x \delta_y \delta_z \quad (3.17)$$

Where  $\delta_z$  is unit depth in this case, which has no influence on the results and will not be discussed further. There are a few different conditions that can occur on the boundary, namely fixed temperature, fixed heat flux and adiabatic. These conditions are discussed below.

*Fixed temperature:* Depending on where the boundary is located, the corresponding flux,  $a_i$ , is set to zero and the value of  $S_p$  and  $S_u$  changes, as shown below:

$$S_p = \frac{2k_p A_f}{\delta} \quad (3.18)$$

$$S_u = \frac{2k_p A_f T_\infty}{\delta} + q'''V \quad (3.19)$$



The reason for the 2 in the numerator in Eq. (3.18) and Eq. (3.19) is that the boundary node is only  $\delta/2$  from the edge of the domain. Important to note here is that the conductivity used is the conductivity of the element on the boundary and not the harmonic mean.

*Heat flux:* For a heat flux, the influence from the boundary direction is zero. Thus  $a_i$  for that specific boundary is set to zero. The only part that changes is  $S_u$  as shown below:

$$S_p = 0 \quad (3.20)$$

$$S_u = q''A_f + q'''V \quad (3.21)$$

*Adiabatic:* The adiabatic boundary is just a special case of the fixed flux boundary, where the value of the heat flux  $q''$  is zero. Thus, for adiabatic cases, the corresponding coefficient  $a_i$  is set to zero just like with a fixed flux, but the following also holds:

$$S_p = 0 \quad (3.22)$$

$$S_u = q'''V \quad (3.23)$$

Using the formulas outlined above, Eq. (3.15) is adjusted for each element, depending on where the element lies (internal or boundary). The coefficients must then be rewritten in terms of the current element  $P$ , for each equation of each element. For example, consider an arbitrary internal element  $i$ :

$$T_i a_i - T_{i-1} a_{i-1} - T_{i+1} a_{i+1} - T_{i+M} a_{i+M} - T_{i-M} a_{i-M} = S_u \quad (3.24)$$

The coefficients  $a_p$ ,  $a_e$ ,  $a_w$ ,  $a_n$  and  $a_s$  have now been rewritten in terms of the numbering scheme  $i$  and all unknowns are moved to the left of the equation. This process is repeated for all elements. All the coefficients and source terms can be calculated, leaving only the temperature of each element as an unknown. Using Eq. (3.24) for each element, a matrix of linear equations can be set up as follows:

$$K\mathbf{T} = \mathbf{b} \quad (3.25)$$

Where  $K [M_\Omega \times M_\Omega]$  is a matrix of the coefficients, also known as the conductivity matrix,  $\mathbf{T} [M_\Omega \times 1]$  is a vector of the unknown temperatures and  $\mathbf{b} [M_\Omega \times 1]$  is the load vector containing the source terms. Thus, Eq. (3.24) for each element is placed in its own row, with the coefficients placed in the appropriate columns.

The matrix  $K$  will be symmetrical and linear, but also very sparse, because for two dimensions, each row can have a maximum of 5 inscriptions for a two-dimensional domain. For this reason, a compressed sparse row (CSR) matrix is used to store the  $K$  matrix. The temperature distribution is then solved with Python<sup>1</sup> using a sparse solver from the SciPy library. For the two-dimensional case, the solver *spsolve* was used. This is a direct sparse solver that solves a sparse matrix  $K\mathbf{T} = \mathbf{b}$ .

---

<sup>1</sup> Python is a powerful programming language used in a wide variety of application domains and runs on Windows, Mac and Linux.

### 3.3 Adjoint Method

#### 3.3.1 Governing Equations

As discussed in Section 2.3, after the temperature distribution is calculated with FVM, the sensitivity analysis must be done. The sensitivity analysis is the process of finding the derivatives of the objective function, in this case, average temperature, with respect to the design variables. These sensitivities are needed in Step I of the MMA algorithm when the gradients of  $g_0$  and  $g_j$  are determined, as discussed in Section 3.4.1. Since the sensitivity analysis depends on the differentiation of a function, there are a number of ways to accomplish this. The first method that comes to mind is analytical differentiation. This elementary method is easy to implement if the function is trivial. With the increase in variables, partial differentiation is required, which complicates the matter. Another method is numerical differentiation, which is done by calculating the function value at two input values and dividing this by the difference between the input values. This is also a useful method to differentiate functions which are too difficult for analytical differentiation. The drawback is that it requires the function value at two points, which greatly increases the computational cost. Another method that can be used is the adjoint method, which will be further discussed in this section.

The adjoint method provides an efficient method of calculating  $dg_0/d\theta$  (where  $g_0$  is the objective function and  $\theta$  is the design variable matrix), which is comparable to the cost of calculating  $\mathbf{T}$  (temperature distribution) once [45]. Consider the matrix of equations needed to calculate the temperature distribution,  $K\mathbf{T} = \mathbf{b}$ . If we want to compute  $dg_0/d\theta$ , this could be done directly using Eq. (3.26).

$$\frac{dg_0}{d\theta} = \frac{\partial g_0}{\partial \theta} + \frac{\partial g_0}{\partial \mathbf{T}} \frac{\partial \mathbf{T}}{\partial \theta} \quad (3.26)$$

Since the objective function  $g_0$  is known, the calculations of  $\partial g_0/\partial \theta$  and  $\partial g_0/\partial \mathbf{T}$  are relatively easy. Calculating  $\partial \mathbf{T}/\partial \theta$  is relatively difficult. This can be done directly from  $K\mathbf{T} = \mathbf{b}$ , by differentiating it by the design parameter  $\theta_i$  as shown in Eq. (3.27):

$$\frac{\partial \mathbf{T}}{\partial \theta_i} = K^{-1} \left( \frac{\partial \mathbf{b}}{\partial \theta_i} - \frac{\partial K}{\partial \theta_i} \mathbf{T} \right) \quad (3.27)$$

This would mean that we would have to solve the Eq. (3.27)  $M_\Omega$  times (number of design variables), once for every component of  $\theta$ . This illustrates the scale of computations that would be required if the mesh size increases. The basic idea of the adjoint method is to note that the function  $f(\mathbf{T}, \theta) = K\mathbf{T} - \mathbf{b}$  is zero. This means that we can replace the objective function  $g_0$  with Eq. (3.28). In all the subsequent equations, the superscript of  $T$  denotes transpose, not temperature.

$$\bar{g} = g_0 - \boldsymbol{\lambda}^T f \quad (3.28)$$

Since  $f$  is zero, we can define  $\boldsymbol{\lambda}$  as any arbitrary vector. The vector will be defined in such a way that the complex derivative  $\partial \mathbf{T}/\partial \theta$  in Eq. (3.26) is removed. Thus:

$$\left. \frac{dg_0}{d\theta} \right|_{f=0} = \left. \frac{d\bar{g}}{d\theta} \right|_{f=0} = \frac{\partial g_0}{\partial \theta} - \boldsymbol{\lambda}^T \frac{\partial f}{\partial \theta} + \left( \frac{\partial g_0}{\partial \mathbf{T}} - \boldsymbol{\lambda}^T \frac{\partial f}{\partial \mathbf{T}} \right) \frac{\partial \mathbf{T}}{\partial \theta} \quad (3.29)$$

From Eq. (3.29), it can be seen that  $\partial \mathbf{T}/\partial \theta$  disappears if the term  $(\partial g_0/\partial \mathbf{T} - \boldsymbol{\lambda}^T \partial f/\partial \mathbf{T})$  is zero or if  $(\partial g_0/\partial \mathbf{T})^T = (\partial f/\partial \mathbf{T})^T \boldsymbol{\lambda}$ . This fact is used to reduce the computational cost by selecting a convenient value of  $\boldsymbol{\lambda}$ . For this case,  $f = K\mathbf{T} - \mathbf{b}$ , the partial derivative  $\partial f/\partial \mathbf{T}$  is equal to  $K$ . From this, the adjoint equation is defined as [46]:

$$K^T \boldsymbol{\lambda} = \left( \frac{\partial g_0}{\partial \mathbf{T}} \right)^T \quad (3.30)$$

Eq. (3.30) is relatively easy to solve since the matrix  $K$  is known and the partial derivative  $\partial g_0 / \partial \mathbf{T}$  is also easy to compute since the objective function is known. Now Eq. (3.29) reduces to:

$$\left. \frac{dg_0}{d\theta_i} \right|_{f=0} = \frac{\partial g_0}{\partial \theta_i} - \boldsymbol{\lambda}^T \left( \frac{\partial K}{\partial \theta_i} \mathbf{T} - \frac{\partial \mathbf{b}}{\partial \theta_i} \right) \quad (3.31)$$

For this specific temperature distribution, the matrix  $K$  and the partial derivative  $\partial g_0 / \partial \mathbf{T}$  are constant, thus,  $\boldsymbol{\lambda}$  only needs to be computed once per temperature iteration. After some manipulation, it comes down to solving a single  $M_\Omega \times M_\Omega$  matrix of equations and then solving Eq. (3.31). Eq. (3.31) must be evaluated  $M_\Omega$  times, thus giving the sensitivities of all the elements in the domain. The adjoint method was successfully implemented in topology optimisation in a number of papers [8], [47] and [10], and was also implemented in this study.

### 3.3.2 Adjoint Method for Two Dimensions

This section will describe how the approach discussed in Section 3.3.1 was implemented with the two-dimensional model. To encourage the solution to consist mainly of 0-1 densities, the intermediate densities were penalised with SIMP. The SIMP approach, even though it does not supply discrete material distribution, allows for additional flexibility in the material density and has been accepted widely among both engineers and researchers interested in topology optimisation in a variety of fields [48] and for this reason was adopted in this study. The penalisation was done by calculating  $\partial g_0 / \partial \theta$  in Step I of the MMA algorithm with penalised densities. Step I of the MMA requires the  $\partial g_0 / \partial \theta$  to find the optimal densities, which will be discussed in Section 3.4.1. The  $K$  matrix and the temperature distribution used in the adjoint method were calculated with the following formulas for conductivity and internal heat generation:

$$k(\theta_i) = k_L + \theta_i^p (k_H - k_L) \quad (3.32)$$

$$q'''(\theta_i) = q_H''' (1 - \theta_i^p) \quad (3.33)$$

For a constant penalisation scheme,  $p$  is kept constant throughout the whole optimisation process. When an increasing penalisation scheme is used, the power  $p$  is raised from 1 in the first iteration to the final chosen value,  $p_f$ , in the final iteration of the optimisation process, in equal increments thus:

$$p_{inc} = \frac{p_f - 1}{I - 1} \quad (3.34)$$

Where  $I$  is the number of iterations in the MMA. Thus, the value of  $p$  for each iteration is:

$$p = \sum_1^I (p_{inc}(I - 1)) + 1 \quad (3.35)$$

The implementation of the adjoint method for a linear system of equations is quick and efficient. The first step is to solve Eq. (3.30) and compute the adjoint vector  $\boldsymbol{\lambda}$ . For this study, the objective function was the average temperature, as defined in Eq. (3.36).

$$g_0(\mathbf{T}) = \frac{1}{M_\Omega} \sum_1^{M_\Omega} T_i \quad (3.36)$$

Since the partial derivative of  $g_0$  in Eq. (3.30) is derived to  $T_i$ , the local element temperature, each derivative will equal to  $1/M_\Omega$ . Thus for an average temperature objective function,  $(\partial g_0/\partial \mathbf{T})^T$  is equal to a  $[M_\Omega \times 1]$  vector with each element containing the value of  $1/M_\Omega$ .  $K^T$  is the transposed matrix of the conductivity matrix. For the FVM, the conductivity matrix is symmetrical along the diagonal, thus  $K$  will not change when transposed. Now Eq. (3.30), repeated below in Eq. (3.37), is easy to solve since it is a linear set of equations.

$$K^T \boldsymbol{\lambda} = \left( \frac{\partial g_0}{\partial \mathbf{T}} \right)^T \quad (3.37)$$

The rest of the derivatives in Eq. (3.31), repeated below in (3.38), will now be discussed individually.

$$\left. \frac{dg_0}{d\theta_i} \right|_{f=0} = \frac{\partial g_0}{\partial \theta_i} - \boldsymbol{\lambda}^T \left( \frac{\partial K}{\partial \theta_i} \mathbf{T} - \frac{\partial \mathbf{b}}{\partial \theta_i} \right) \quad (3.38)$$

The first term in Eq. (3.38) is zero since the average temperature objective function is not dependent on  $\theta$ . This will not be the same for other objective functions. The second term  $\boldsymbol{\lambda}^T$  has already been calculated as shown above and it only needs to be calculated once per iteration. The most complicated and time-consuming part of the adjoint equation is the term  $\partial K/\partial \theta_i$ . To find this term requires that the whole conductivity matrix must be derived to  $\theta_i$ , once for each design variable. Although this initially seems to be a major challenge, with some manipulation, this term is calculated quite easily as discussed later. The only variable type in the conductivity matrix that contains the design variable  $\theta$ , is  $k$ , the thermal conductivity, as shown in Eq. (3.39).

*Non-boundary elements:*

$$k(\theta_i) = k_L + \theta_i^p (k_H - k_L) \quad (3.39)$$

Fortunately, for the two-dimensional FVM domain, a maximum of 13 entries in the  $K$  matrix (of size  $M_\Omega \times M_\Omega$ ) is dependent on each design variable  $\theta_i$  for non-boundary elements. This is due to the harmonic average used, which is dependent on the conductivity of the current nodal position and the neighbouring nodal positions in four directions.

Thus, all other entries in the matrix are not dependent on the design variable  $\theta_i$ , which means when the matrix is differentiated to  $\theta_i$ , all those entries will result in zero derivatives. This is where the real gain in the adjoint method becomes clear, since it eliminates a lot of computations. This will be exactly the same for all the internal nodes. For boundary nodes, in the case where coefficients are set to zero, that derivative will also be zero. The derivative of the coefficients in the matrix is done as in Eq. (3.40).

$$\frac{da_i}{d\theta_i} = \frac{A_f}{\delta_{PN}} \frac{dk_l}{d\theta_i} \quad (3.40)$$

Only  $k_l$  is a function of  $\theta_i$  as described by the harmonic mean and the SIMP method, thus for a structured mesh where all the nodes have the same length:

$$k_l = \frac{2k_P k_N}{k_P + k_N} \quad (3.41)$$

Using the quotient rule for the derivation where  $'$  indicates a derivative to  $\theta_i$  ( $d/d\theta_i$ ):

$$\frac{dk_l}{d\theta_i} = \frac{2[(k_P k_N)'(k_P + k_N) - (k_P + k_N)'(k_P k_N)]}{(k_P + k_N)^2} \quad (3.42)$$

After some simplification, since  $k_N$  is not dependent on  $\theta_i$ :

$$\frac{dk_L}{d\theta_i} = \frac{2[(k'_P k_N)(k_P + k_N) - (k'_P)(k_P k_N)]}{(k_P + k_N)^2} \quad (3.43)$$

All the variables in this equation are known, except  $k'_P$ . This is, however, easy to compute from the SIMP equation:

$$k_P = k_L + \theta_i^p (k_H - k_L) \quad (3.44)$$

$$\frac{dk_P}{d\theta_i} = p\theta_i^{p-1} (k_H - k_L) \quad (3.45)$$

If the element is on a boundary, the term  $a_i$  in the conductivity matrix will contain a source term,  $S_P$ . If this source term is dependent on the design variable  $\theta_i$ , it must also be derived and added.

#### *Boundary elements*

*Fixed temperature:* For a fixed temperature, the  $S_P$  term is dependent on the design variable, thus:

$$S_P = \frac{2k_P A_f}{\delta_{PN}} \quad (3.46)$$

$$\frac{dS_P}{d\theta_i} = \frac{2[p\theta_i^{p-1} (k_H - k_L)] A_f}{\delta_{PN}} \quad (3.47)$$

*Fixed heat flux:* For a fixed heat flux boundary,  $S_P$  is zero, thus  $S_P$  is in no way dependent on the design variable  $\theta_i$  and the derivative is zero.

*Adiabatic boundary:* For an adiabatic boundary,  $S_P$  is zero and thus the derivative is also zero.

*Internal heat generation:* For internal heat generation,  $S_P$  is also zero, thus the derivative is zero.

Since all the non-zero elements are known, the multiplication with the  $\mathbf{T}$  vector can also be simplified by just multiplying the elements that are non-zero. Only 13 entries are non-zero, thus for a 100 x 100 mesh, there will be 99.99 x 10<sup>6</sup> zero entries. Multiplying the whole matrix by the complete  $\mathbf{T}$  vector would waste computational time. Since the exact locations of all the non-zero entries are known, these only need to be multiplied with the corresponding entries in the  $\mathbf{T}$  vector.

The next step is to calculate the source term vector derivative  $d\mathbf{b}/d\theta_i$ . This is easy since there will only be one non-zero entry, corresponding to the index of the current element.

#### *Non-boundary elements*

Non-boundary elements only contribute to the  $\mathbf{b}$  vector if internal heat generation is present. Internal heat generation only has an effect on  $S_u$  in the following way:

$$S_u = q''' V \quad (3.48)$$

Where the internal heat generation is defined as follows:

$$q''' = q'''_H (1 - \theta_i^p) \quad (3.49)$$

Thus:

$$\frac{dS_u}{d\theta_i} = q_H'''(-p\theta_i^{p-1}) \quad (3.50)$$

*Boundary elements*

Fortunately, boundary elements are not difficult to derive either for the  $db/d\theta_i$  term. The following source terms and their derivatives are applicable:

*Fixed temperature:* For a fixed temperature, the following applies:

$$S_u = \frac{2k_p A_f T_\infty}{\delta_{PN}} \quad (3.51)$$

$$\frac{\partial S_u}{\partial \theta_i} = \frac{2[p\theta_i^{p-1}(k_H - k_L)]A_f T_\infty}{\delta_{PN}} \quad (3.52)$$

*Fixed heat flux:* For a fixed heat flux boundary,  $S_u$  is equal to the flux on the boundary (which is a fixed value) multiplied by the area of that face. Thus,  $S_u$  is in no way dependent on the design variable  $\theta_i$  and the derivative is zero.

*Adiabatic boundary:* For an adiabatic boundary,  $S_u$  is zero and thus the derivative is also zero.

Using the above equations, Eq. (3.38) is easily solved to give the sensitivities of an element according to the objective function. The process outlined in this subsection is repeated for all the elements in the domain to give the complete sensitivity matrix.

## 3.4 Method of Moving Asymptotes

### 3.4.1 Governing Equations

The method of moving asymptotes, introduced by Krister Svanberg [49], is a method of solving a non-linear optimisation problem by introducing a strictly convex subproblem. The generation of the subproblem is controlled by the “moving asymptotes”. It is based on a first-order Taylor series of the objective and constraint functions. The method, by moving the so-called asymptotes, stabilises and speeds up the convergence rate of the general optimisation problem. The MMA has been shown to be well suited for structural and multi-disciplinary optimisation applications, especially where reciprocal or reciprocal-like approximations are used [50]. The algorithm has attracted significant interest from the topology optimisation community [50 - 51] and is for this reason adopted for this investigation. This method was successfully implemented in topology optimisation routines [8], [47].

The general description is shown in Eq. (3.53) - (3.55).

The objective is to minimise a function,  $g_0$ :

$$g_0(\theta) \quad \theta \in R^n \quad (3.53)$$

Subject to a set of one or more constraints:

$$g_j(\theta) \leq \hat{g}_j \quad \text{for } j = 1, \dots, m \quad (3.54)$$

Such that the design variable for each volume in the FVM scheme is restricted according to:

$$\underline{\theta} \leq \theta_i \leq \bar{\theta} \quad \text{for } i = 1, \dots, M_\Omega \quad (3.55)$$

The following approach is used to generate and solve a sequence of explicit subproblems. Although it is quite general, it explains the process:

- Step 0: Choose a starting point  $\theta^{(0)}$  and let the iteration index  $I = 0$
- Step I: Given iteration point  $\theta^{(I)}$ , calculate  $g_j(\theta^{(I)})$  and the gradients  $\nabla g_j(\theta^{(I)})$  for  $j = 0, 1, \dots, m$
- Step II: Generate the subproblem  $Q^{(I)}$  by replacing in  $Q$ , the (usually implicit) functions  $g_j$  and approximating explicit functions  $g_j^{(I)}$ , based on the calculations from *Step I*
- Step III: Solve  $Q^{(I)}$  and let the optimal solution for this subproblem be the next iteration point  $\theta^{(I+1)}$ . Let  $I = I + 1$  and go to *Step I* until some convergence or termination criteria are reached.

Some of the advantages of MMA are:

- It can handle non-linear problems;
- It is not sensitive to translation or scaling of variables.

Svanberg also introduced a dual method for solving the subproblems. He concluded that the method is very flexible and in the cases that he tested, converged faster than sequential linear programming (SLP), which in some cases did not even converge.

### 3.4.2 Method of Moving Asymptotes for Two Dimensions

The final step in the topology optimisation process is to use the temperature distribution and sensitivities obtained from FVM and the adjoint method to find the optimal distribution of the densities. The MMA will not be explained in detail here, since it is quite tedious and complex. As explained in Section 3.4.1, the MMA replaces the usually implicit objective and constraint functions with explicit functions  $g_0^{(I)}$  and  $g_j^{(I)}$ . There are several objective functions that can be used for heat conduction including average temperature, maximum temperature and dissipation of heat transport potential capacity. For the two-dimensional case, the average temperature was used for the objective function as mentioned earlier. Dirker and Meyer showed in their paper that average temperature is a suitable objective function [10]. Although the maximum temperature is important in the maximum power density of electronic components, it was proven by Dirker and Meyer that for the conditions used in this study, maximum temperature is not a viable objective function [10].

$$g_0(\mathbf{T}) = \frac{1}{M_\Omega} \sum_1^{M_\Omega} T_i \quad (3.56)$$

For heat conduction, the only necessary constraint is a volume constraint, even though the MMA can handle multiple constraints. The effective volume proportion occupied by the high-conductive solid can be calculated by the sum of the densities of all the elements divided by the total number of elements:

$$g_1(\boldsymbol{\theta}) = V_f = \frac{1}{M_\Omega} \sum_1^{M_\Omega} \theta_i \leq V^* \quad (3.57)$$

Where  $V_f$  is defined as the volume ratio obtained after the MMA has solved and  $V^*$  is the maximum allowed volume constraint. The first iteration requires an initial guess for the density matrix,  $\theta_0$ . A good start for the algorithm is to set the density of each element equal to the volume constraint  $V^*$ , thus the initial guess will satisfy the volume constraint. The above two functions are used to generate the explicit functions needed to generate the subproblem of the MMA. The part that makes the MMA special is the use of moving asymptotes. These asymptotes are calculated depending on the signs of the previous two iteration values of the density matrix. These equations contain the values  $s$  and  $s_0$ , which control how much the solution relaxes. For the first two iterations, the asymptotes are calculated as follows, where  $\bar{\theta}$  and  $\underline{\theta}$  are the maximum and minimum allowed density respectively. For this study,  $\bar{\theta} = 1$  and  $\underline{\theta} = 0$ .

$$L_i^{(I)} = \theta_i^{(I)} - s_0(\bar{\theta} - \underline{\theta}) \quad (3.58)$$

$$U_i^{(I)} = \theta_i^{(I)} + s_0(\bar{\theta} - \underline{\theta}) \quad (3.59)$$

Where the value of  $s_0$  can be any real number. For  $I > 2$ , the asymptotes are calculated using the previous iteration asymptotes and the densities as follows:

- a) If the signs of  $\theta_i^{(I)} - \theta_i^{(I-1)}$  and  $\theta_i^{(I-1)} - \theta_i^{(I-2)}$  are opposite, it indicates an oscillation in the variable  $\theta_i^{(I)}$ , thus:

$$L_i^{(I)} = \theta_i^{(I)} - s(\theta_i^{(I-1)} - L_i^{(I-1)}) \quad (3.60)$$

$$U_i^{(I)} = \theta_i^{(I)} - s(U_i^{(I-1)} - \theta_i^{(I-1)}) \quad (3.61)$$

- b) If the signs of  $\theta_i^{(I)} - \theta_i^{(I-1)}$  and  $\theta_i^{(I-1)} - \theta_i^{(I-2)}$  are equal, it indicates that the asymptotes are slowing down the variable  $\theta_i^{(I)}$ , thus:

$$L_i^{(I)} = \theta_i^{(I)} - \frac{(\theta_j^{(I-1)} - L_j^{(I-1)})}{s} \quad (3.62)$$

$$U_i^{(I)} = \theta_i^{(I)} - \frac{(U_j^{(I-1)} - \theta_j^{(I-1)})}{s} \quad (3.63)$$

Where the value of  $s$  can be any number less than unity but bigger than zero. It was found that the first two asymptotes and the value of  $s$  and  $s_0$  are critical to the final solution.

Once the explicit functions are generated, they are used to generate the subproblem. The subproblem is solved by introducing a dual method, by rewriting the subproblem as a dual objective function (for more information, refer to the paper), which is a concave function and can easily be solved with any minimisation optimisation algorithm like the spherical quadratic steepest descent method (SQSD), which was used for the two-dimensional domain. Once the solution to the subproblem is found, it is used to calculate the optimal density distribution for that iteration.

At the end of each MMA iteration, the value of  $p$  is updated according to the scheme used (constant or increasing). The whole process is then started again, using the optimised density distribution, the new temperature distribution is calculated as well as the updated sensitivities. The MMA then calculates the optimised density distribution and this process is repeated for a fixed number of iterations. This process is explained in detail in the next section.

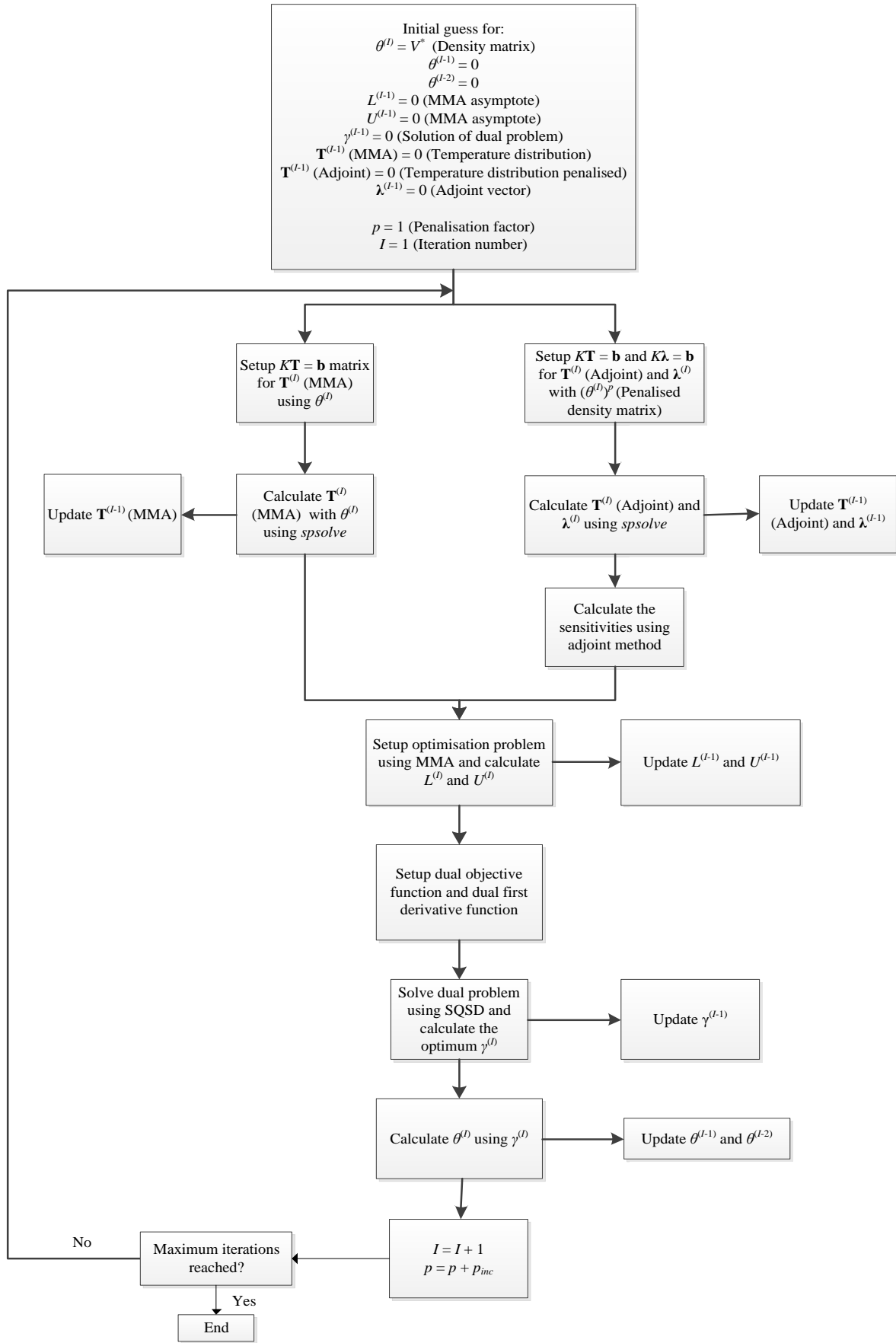


### 3.5 Program Methodology

The basic process of the topology optimisation for any dimension is as follows:

1. Provide an initial guess for the density field
2. Calculate the temperature distribution
3. Calculate the sensitivities of the density distribution according to the objective and constraint function
4. Use the MMA together with the sensitivities and temperature distribution to find the new density distribution
5. Go to Step 1.

This process is explained in more detail in the flow chart in Figure 3.3. For three dimensions, this process is followed exactly, only the third dimension is added in the temperature distribution and sensitivities.



**Figure 3.3:** Program methodology.

### 3.6 Summary

This chapter covered the governing equations and implementation of FVM, the adjoint method and the MMA for a two-dimensional thermal domain. These three methods together are used to find the optimal material distribution in a domain, for a specific set of boundary conditions. The next chapter will cover the validation of all the methods as well as some optimisation runs for a two-dimensional partial Dirichlet boundary.

## Chapter 4 Two-Dimensional Validation and Results

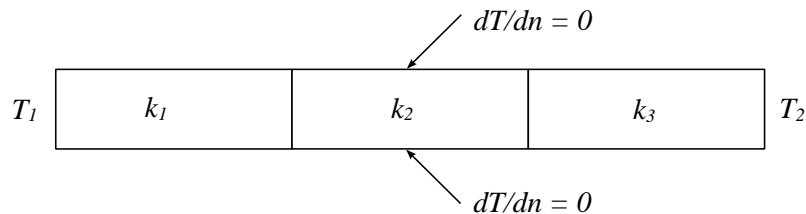
### 4.1 Introduction

The purpose of this chapter is to validate and test the two-dimensional code for a partial Dirichlet boundary. The temperature distribution and adjoint method were validated using theory and the MMA was validated using other papers. The effect of the asymptote parameters,  $s$  and  $s_0$ , was tested extensively and a mesh-dependence study was conducted.

### 4.2 Validation

#### 4.2.1 Temperature Distribution

Since the thermal FVM model formed the basis of the optimisation study, it was vital to check its accuracy. To test this, both one-dimensional and two-dimensional test cases were evaluated. For the one-dimensional validation, the two-dimensional problem was converted to a one-dimensional problem, by setting boundaries across from each other to a fixed temperature and adiabatic respectively. The nodal temperature can be easily calculated theoretically. This was done for both  $x$ - and  $y$ -directions with and without heat generation. The domain is shown in Figure 4.1. All results correlated with the theory.



**Figure 4.1:** Domain for one-dimensional validation.

Boundary behaviour was validated against a two-dimensional case found in Versteeg [42] for fixed temperature, heat flux and adiabatic boundary types. It was found that the results correlated well with the example as shown in Table 4.1. All the results correlated well with the example except one node. This was suspected to be a typing mistake. This validation was also done with the use of the OpenFOAM numerical code. Comparison of the results is given in Table 4.2.

**Table 4.1:** Nodal temperatures for the example in Versteeg and the finite volume code.

FVM	1	2	3	Versteeg	1	2	3
1	146.322 K	129.696 K	123.982 K	1	146.3 K	129.7 K	124.0 K
2	205.592 K	178.178 K	166.23 K	2	205.6 K	178.1 K	166.2 K
3	242.275 K	211.195 K	196.53 K	3	242.2 K	211.1 K	196.5 K
4	260.037 K	227.799 K	212.164 K	4	260.0 K	222.7 K	212.1 K

**Table 4.2:** Nodal temperature comparison between FVM and OpenFoam.

FVM	1	2	3	OpenFoam	1	2	3
1	146.322 K	129.696 K	123.982 K	1	146.3 K	129.7 K	124.0 K
2	205.592 K	178.178 K	166.23 K	2	205.6 K	178.1 K	166.2 K
3	242.275 K	211.195 K	196.53 K	3	242.2 K	211.1 K	196.5 K
4	260.037 K	227.799 K	212.164 K	4	260.0 K	222.7 K	212.1 K

The largest error is  $4.033 \times 10^{-4}$ , which was deemed acceptable. The partial isothermal boundary on the lower edge of the domain was tested with a number of symmetry tests. It was also tested using energy balance by comparing the heat flow at this boundary with the steady-state heat generation in the domain as described in Eq. (4.1):

$$\frac{kA(T - T_{\infty})}{\frac{1}{2}\delta_x} = q'''V \quad (4.1)$$

The energy balance was satisfied within a 0.01% margin of error. It was concluded that the two-dimensional FVM thermal model was working correctly.

#### 4.2.2 Adjoint Method

The accuracy of the sensitivities obtained with the adjoint method was checked by comparing with those obtained via the direct finite difference method:

$$\frac{\partial g_0}{\partial \theta_i} = \frac{g_0((\theta_i + \Delta\theta)^p) - g_0(\theta_i^p)}{\Delta\theta} \quad (4.2)$$

Where  $\Delta\theta$  was taken as 0.000001. This was calculated for every elemental density to form the matrix of sensitivities. The adjoint method results correlated perfectly with the finite difference results, when  $p_f = 1$  (i.e. no penalisation). For  $p_f \neq 1$  (with penalisation), the average difference was less than  $1 \times 10^{-6}$ .

#### 4.2.3 Method of Moving Asymptotes

Since the MMA algorithm is quite long and complex, it is difficult to validate the self-developed code with theory. The code was, however, validated by comparing its results with the results from other researchers. This is discussed in Section 4.3.4.

**Table 4.3:** Progression of the MMA.

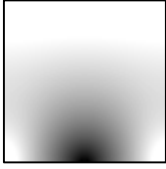

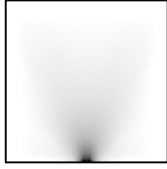
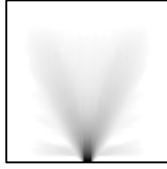
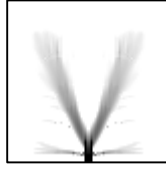
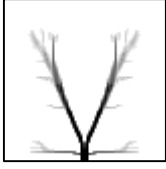
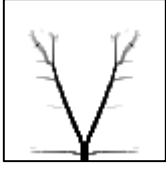
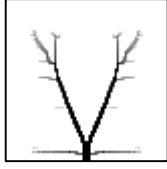
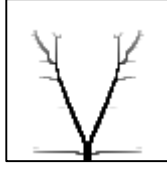
$I$ [-]	1	5	10	15	20
Density distribution					
$I$ [-]	25	30	35	40	
Density distribution					

Table 4.3 shows a set of densities from an arbitrary optimisation run. The figure shows how the MMA steadily evolves the tree structure from the first iteration, placing most of the material close to the cold spot. After  $I = 25$ , the tree structure is established, with only changes in the secondary branches after that point.

### 4.3 Methodology and Results

#### 4.3.1 Influence of Input Parameters

In this section, the effects of  $k_L$ ,  $k_H$ ,  $q_H$ ,  $T_\infty$  and the dimensions of the domain ( $L_D$ ,  $W_D$ ) are considered. Values for these parameters were chosen to cover approximate realistic application restrictions. More will be said later on regarding the choice of  $c$ , the width of the isothermal cold boundary. A course mesh is used here to facilitate a fast investigation. The mesh density will be increased later on. The input parameters are shown in Table 4.4.

**Table 4.4:** Input parameters for determining the dimensionless temperature.

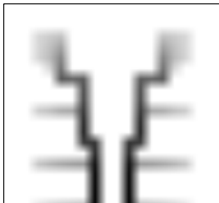
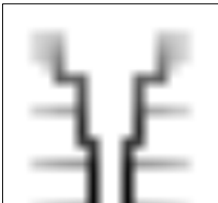
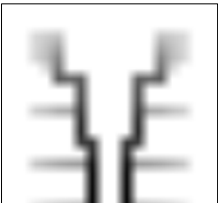
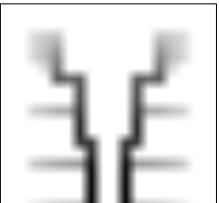
Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	500.0 W/(mK)	$I$	40 iterations
$q_H$	10.0 W/m <sup>3</sup>	$c/L_D$	0.2
$M \times M$	20x20	$T_\infty$	0 K

Figure 4.2 shows the scale used to represent the density of each element. This scale will be used throughout the study.


**Figure 4.2:** A scale of the density.

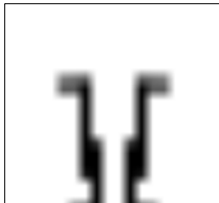
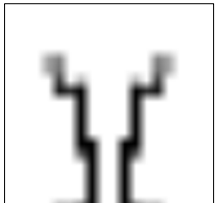
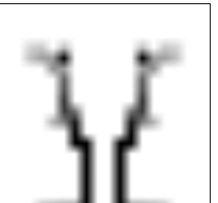
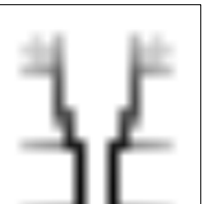
The first practical parameter to test is the temperature of the cold spot on the bottom of the boundary. Table 4.5 shows architectures for different values of  $T_\infty$ . The results show that an increase in  $T_\infty$  results in an increase in the maximum temperature, as expected. The architecture does not change for different values of  $T_\infty$  though. The maximum temperature can be normalised by subtracting the boundary temperature from the maximum temperature as shown in the table.

**Table 4.5:** Architectures for different values of  $T_\infty$ .

$T_\infty$ [K]	0.0	10.0	50.0	100.0
$T_{max}$ [K]	$1.496 \times 10^{-1}$	$1.015 \times 10^1$	$5.015 \times 10^1$	$1.001 \times 10^2$
$T_{max} - T_\infty$ [K]	$1.496 \times 10^{-1}$	$1.496 \times 10^{-1}$	$1.496 \times 10^{-1}$	$1.496 \times 10^{-1}$
Density distribution				





Secondly, the conductivity of the substrate material,  $k_L$ , is kept constant while varying  $k_H$ . This will result in different conductivity ratios, defined as  $k^* = k_H/k_L$ . As shown in Table 4.6, as  $k_H$  is increased, the conductivity ratio increases, which results in a lower maximum temperature  $T_{max}$ , as expected. The architectures also change for each conductivity ratio with the branches decreasing in size with an increase in  $k^*$ . This shows that even if  $k_L$  is kept constant, when the conductivity ratio changes, there will be a change in the final solution.

**Table 4.6:** Architectures for different conductivity ratios with  $k_L$  kept constant.

$k_L$ [W/(mK)]	2	2	2	2
$k_H$ [W/(mK)]	100	200	400	800
$k^*$ [-]	50	100	200	400
$T_{max} - T_\infty$ [K]	$7.788 \times 10^{-1}$	$4.718 \times 10^{-1}$	$2.750 \times 10^{-1}$	$1.562 \times 10^{-1}$
Density distribution				

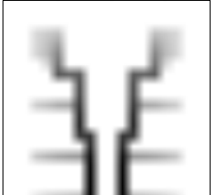
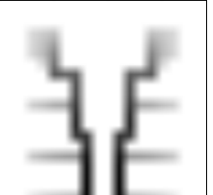
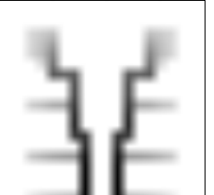
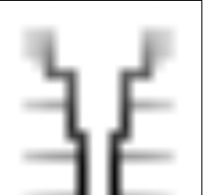
To test whether the conductivity ratio influences the final architecture, a particular constant  $k^*$  value is investigated but at different  $k_L$  and  $k_H$  values. As seen in Table 4.7, different values for  $k_L$  and  $k_H$  which give the same value for  $k^*$ , result in the same architecture. The maximum temperature is higher when the values of  $k_L$  and  $k_H$  are lower. However, the maximum temperature can be normalised when multiplying it by  $k_L$ , resulting in the same value for all the ratios, as shown in the table.

**Table 4.7:** Architectures for the same conductivity ratio with different values of  $k_H$  and  $k_L$ .

$k_L$ [W/(mK)]	5	10	50	100
$k_H$ [W/(mK)]	500	1000	5000	10000
$k^*$ [-]	100	100	100	100
$T_{max} - T_{\infty}$ [K]	$1.887 \times 10^{-1}$	$9.437 \times 10^{-2}$	$1.887 \times 10^{-2}$	$9.437 \times 10^{-3}$
$(T_{max} - T_{\infty})k_L$ [W/m]	$9.437 \times 10^{-1}$	$9.437 \times 10^{-1}$	$9.437 \times 10^{-1}$	$9.437 \times 10^{-1}$
Density distribution				

Another input parameter that has an influence of the MMA converged maximum temperature is the internal heat generation density  $q_H'''$ . This influence is demonstrated in Table 4.8 for a conductivity ratio of 500. As can be seen, when the internal heat generation rate is increased, the maximum temperature increases but the architectures remain the same. When the maximum temperature is divided by the internal heat generation, a normalised temperature difference is obtained.

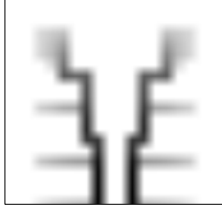
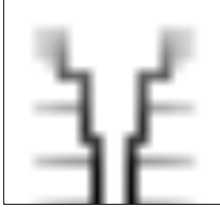
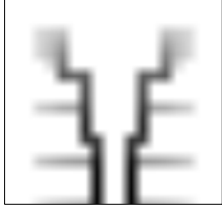
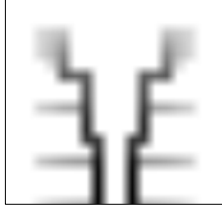
**Table 4.8:** Architectures for different values of  $q_H$ .

$q_H'''$ [W/m <sup>3</sup> ]	0.01	0.1	1.0	10.0
$T_{max} - T_{\infty}$ [K]	$1.496 \times 10^{-4}$	$1.496 \times 10^{-3}$	$1.496 \times 10^{-2}$	$1.496 \times 10^{-1}$
$(T_{max} - T_{\infty})/q_H'''$ [Km <sup>3</sup> /W]	$1.496 \times 10^{-2}$	$1.496 \times 10^{-2}$	$1.496 \times 10^{-2}$	$1.496 \times 10^{-2}$
Density distribution				

The same can now be done for the domain size. As seen in Table 4.9, the size of the domain does not have an influence on the final architecture. For these results, the relative size of the cold spot was kept the same at  $c/L_D = 0.2$ . The size of the domain does, however, affect the maximum temperature. A smaller domain yields a lower maximum temperature, however, the maximum temperature can be normalised if divided by the characteristic length of the domain, squared.



**Table 4.9:** Architectures for different sizes of the domain.

$L_D = W_D$ [m]	0.01	0.1	1.0	10.0
$T_{max} - T_{\infty}$ [K]	$1.496 \times 10^{-5}$	$1.496 \times 10^{-3}$	$1.496 \times 10^{-1}$	$1.496 \times 10^1$
$(T_{max} - T_{\infty})/L_D^2$ [K/m <sup>2</sup> ]	$1.496 \times 10^{-1}$	$1.496 \times 10^{-1}$	$1.496 \times 10^{-1}$	$1.496 \times 10^{-1}$
Density distribution				

From the above, a single dimensionless temperature measure can be defined:

$$\tau = \frac{(T_{max} - T_{\infty})k_L}{q_H''L_D^2} \quad (4.3)$$

Other investigations that also used such a measure include [13], [52]. The other input parameters such as volume constraint and size of isothermal boundary were also considered but not reported here. It might be noted that the objective function in this investigation is the average domain temperature, while the temperature measure,  $\tau$ , uses the peak temperature in the domain. These are not strictly speaking equivalent, however, it has been found that their relative behaviour is very similar for the condition covered here.

In order to conform to previously published results [10], the user of  $\tau$  is adopted in this investigation. Also, it would be exceedingly computationally expensive to redefine the objective function to be equal to the peak domain temperature, since the adjoint method is not suitable convenient for this, and also it has been shown that the use of the peak domain temperature as the objective function resulted in poorer performing material distributions [10].

### 4.3.2 The Influence of Asymptote Parameters, $s$ and $s_0$

An important part of the MMA algorithm is the moving asymptotes, which make the algorithm unique. These moving asymptotes depend on the signs of the previous two densities and the current density. Thus, for the first two iterations, the asymptotes are fixed as follows:

$$L_i^{(l)} = \theta_i^{(l)} - (\bar{\theta} - \theta) \quad (4.4)$$

$$U_i^{(l)} = \theta_i^{(l)} + (\bar{\theta} - \theta) \quad (4.5)$$

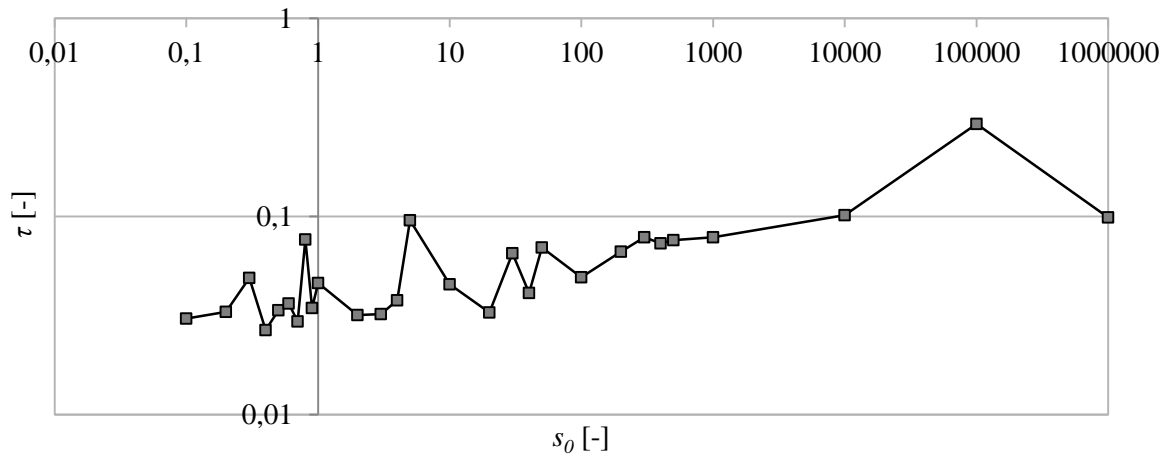
It was found that the first two iterations' asymptotes have a significant impact on the final result of the optimisation. If Eq. (4.4) and Eq. (4.5) are used for the first two iterations, the density distribution is primarily located near the cold spot at the bottom and stays there until the final iteration and does not spread widely in the domain. Svanberg suggests another formula for the fixed asymptotes, which uses the parameter,  $s_0$ , and was discussed in Section 3.4.2.

Optimisation runs were conducted to test the influence of  $s_0$  while using the default value for  $s$ . It was found that with, for instance,  $s_0 = 0.1$ , a greatly improved final temperature distribution could be achieved. The influence of  $s_0$  is given in Figure 4.3. The input parameters for the simulations of the influence of  $s_0$  and  $s$  are given in Table 4.10. For now, a volume constraint of  $V^* = 0.1$  is used.

**Table 4.10:** Input parameters for the influence of  $s$  and  $s_0$ .

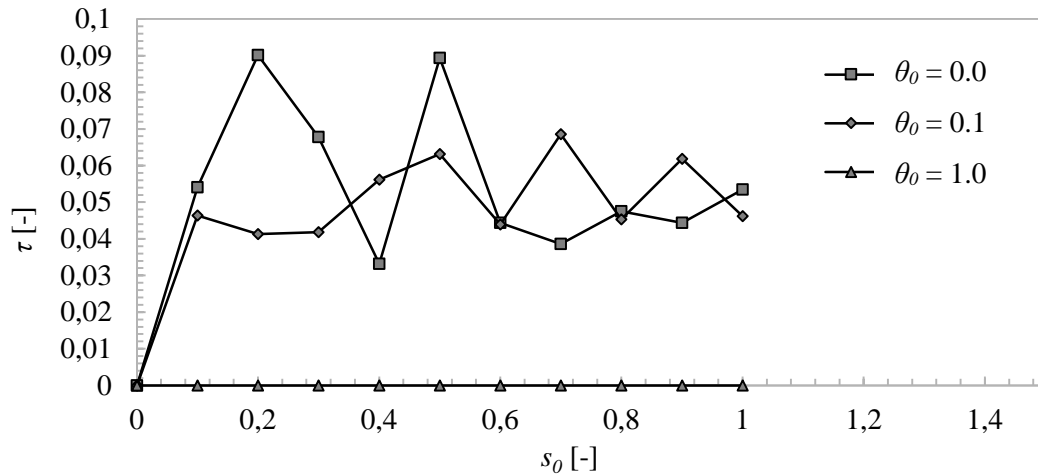
Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$I$	40 iterations
$q_H'''$	10.0 W/m <sup>3</sup>	$c/L_D$	0.05
$M \times M$	20x20	$T_\infty$	0 K
$s$	0.7		

To observe the influence of  $s_0$  in the first two iterations,  $s_0$  was ranged between -1 000 000 and 1 000 000. It was found that negative values of  $s_0$  destabilise the optimisation loop. As seen from the graph in Figure 4.3, as  $s_0$  increases, so does  $\tau$  with a destabilising effect, oscillating between high and low values.



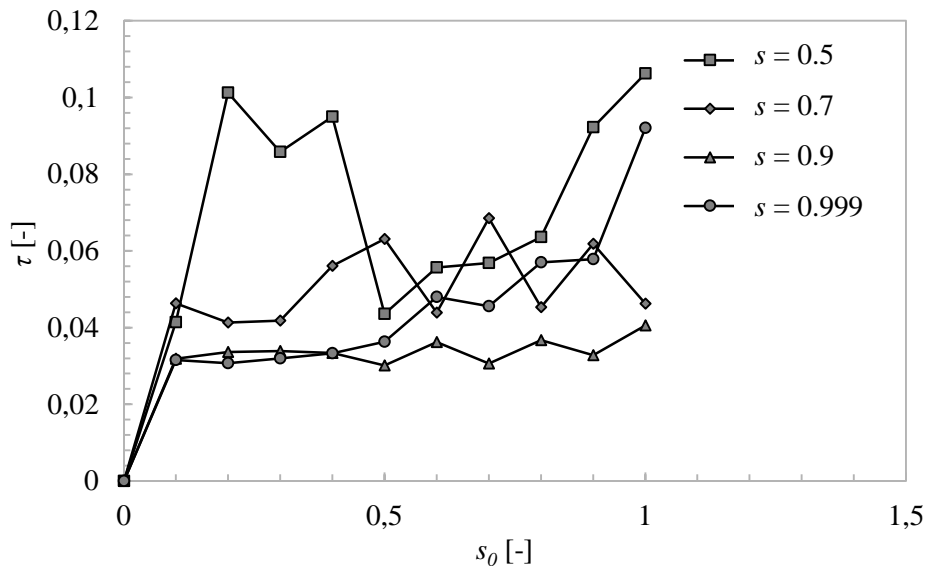
**Figure 4.3:** The influence of  $s_0$  for the two-dimensional partial Dirichlet boundary.

From Figure 4.3, it is evident that for  $s_0 \in (0: 1)$ , there is a definitive gain in the maximum temperature compared with higher values of  $s_0$ . It was decided to range  $s_0$  between 0 and 1 for different values of  $s$ . Svanberg recommends  $s = 0.7$  for the moving asymptotes, but also states the choosing of the asymptotes has to be tested more in the future. Using  $\sqrt{s}$  is recommended to make the solution more stable and conservative. The effect of the initial density matrix,  $\theta_0$ , was also tested.



**Figure 4.4:** The influence of  $s_0$  and  $\theta_0$  for  $s = 0.7$  for the two-dimensional case.

From Figure 4.4 (plotted for the default value of  $s = 0.7$ ), it is evident that by using a uniform initial density field of  $\theta_0 = 1.0$ , which is greater than the volume constraint, no feasible solution is found. For an initial guess of  $\theta$  uniformly equal to 0.1, which is equal to the volume constraint, the solution gives stable answers.  $\tau$  does vary across the range of  $s_0$ , but the solution does not seem to diverge. Other values of  $s$ , besides 0.7, were also considered as shown in Figure 4.5. As  $s$  is increased, lower  $\tau$  values are obtained, indicating that the MMA algorithm, for this particular set of input parameters, performs better. However, as  $s$  reaches unity,  $\tau$  does seem to increase for larger values of  $s_0$ . For a value of  $s = 0.9$ , the  $\tau$  value appears to be the most consistent.



**Figure 4.5:** The effect of  $s$  on the maximum temperature for different values of  $s_0$  with  $\theta_0 = 0.1$ .

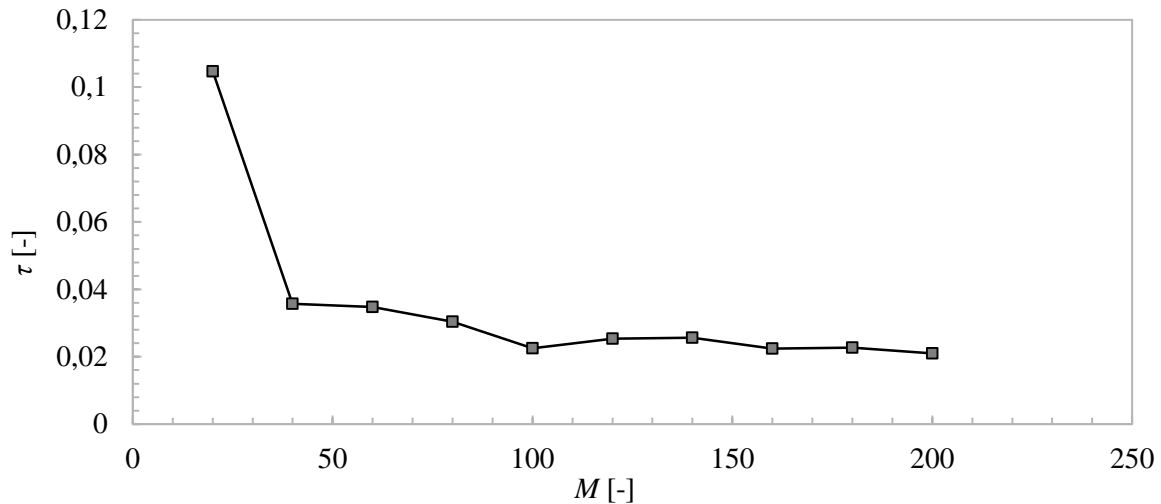
The rest of the results can be found in Appendix A.1. For the rest of the two-dimensional analysis,  $s = 0.9$  and  $s_0 = 0.1$  were used. This selection was validated routinely in the investigation.

### 4.3.3 Mesh-Dependence Study

A mesh-dependence study was done. The results are given in Figure 4.6, while the achieved architectures are given in Table 4.12. The input parameters used are given in Table 4.11. Figure 4.6 shows that there is a significant drop in the maximum temperature when the mesh density is increased from 20x20 to 40x40 nodes. Above 40x40,  $\tau$  starts to stabilise and appears to be reaching an asymptote.

**Table 4.11:** Input parameters used for the mesh dependence of the two-dimensional case.

Parameter	Value	Parameter	Value
$L$	1.0 m	Penalisation scheme	Increasing
$W$	1.0 m	$p_f$	3
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$I$	40 iterations
$q_H$	10.0 W/m <sup>3</sup>	$c/L_D$	0.05
$M \times M$	20x20 – 200x200	$s, s_0$	0.9, 0.1
$T_\infty$	0 K		


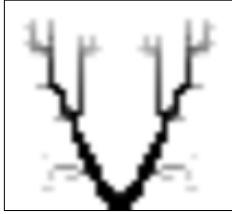
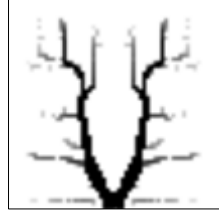
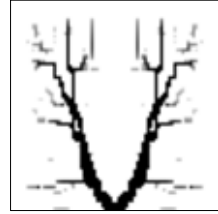
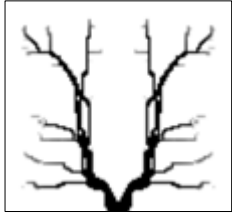
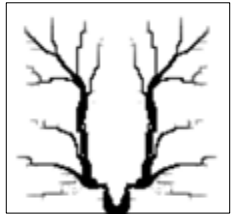
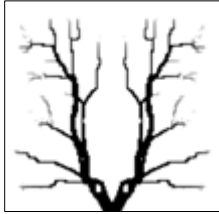
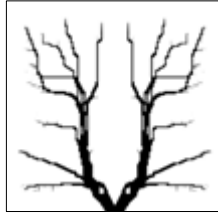
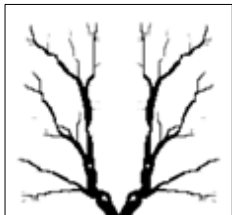
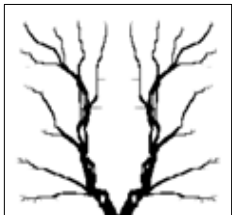


**Figure 4.6:** Mesh-dependence for the two-dimensional case.

For the two-dimensional domain,  $M = 100$  was deemed to be sufficient since the maximum temperature does not change significantly thereafter. The difference in  $\tau$  between the mesh densities of 100x100 and 160x160 is only 0.345%. The  $\tau$  value does drop when the mesh density is increased to 180x180 and 200x200, but the change was deemed negligible since the two dimensional studies are not the main focus of the study and solving a 200x200 mesh is computationally expensive.

The architectures in Table 4.12 show that the primary branches have an initial V shape, which then extend to the corners of the domain. After  $M = 100$ , the main shape of the architecture is converged, which correlates with the convergence of  $\tau$ . After this point, there are only changes in the secondary branches. As the number of elements increases, the number of secondary branches also increases. The secondary branches also grow with an upward gradient with the increase of elements compared with the straight secondary branches observed for  $M = 60$  and  $M = 80$ .

**Table 4.12:** Architectures for the two-dimensional mesh dependence.

$M$ [-]	20	40	60	80
$\tau$ [-]	$2.094 \times 10^{-1}$	$7.133 \times 10^{-2}$	$6.950 \times 10^{-2}$	$6.083 \times 10^{-2}$
Density distribution				
$M$ [-]	100	120	140	160
$\tau$ [-]	$4.503 \times 10^{-2}$	$5.067 \times 10^{-2}$	$5.119 \times 10^{-2}$	$4.488 \times 10^{-2}$
Density distribution				
$M$ [-]	180	200		
$\tau$ [-]	$4.541 \times 10^{-2}$	$4.196 \times 10^{-2}$		
Density distribution				

#### 4.3.4 Comparing Results With Other Papers

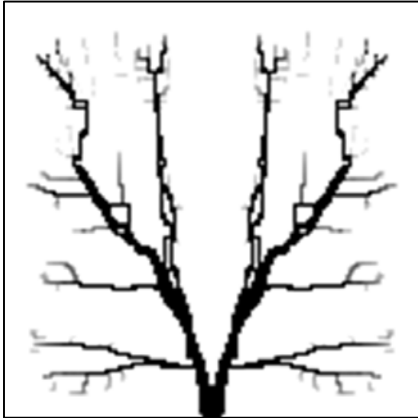
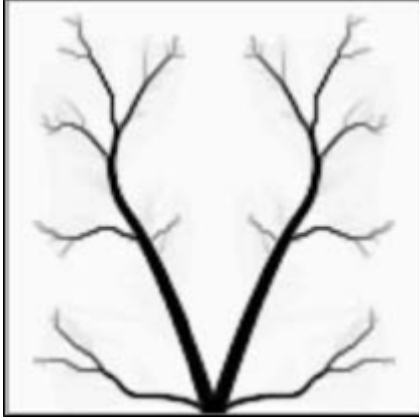
In this section, the code will be validated against the paper by Gersborg-Hansen *et al.* [8]. One case was tested with a partial Dirichlet boundary at  $V^* = 0.1$ . The input parameters are given in Table 4.13.

**Table 4.13:** Input parameters for validation of code with Gersborg-Hansen *et al.*

Parameter	Value	Parameter	Value
$L$	1.0 m	Penalisation scheme	Increasing
$W$	1.0 m	$V^*$	0.1 (10%) or 0.4 (40%)
$k_L$	0.001 W/(mK)	$I$	40 iterations
$k_H$	1.0 W/(mK)	$p$	3.0
$q_H$	0.01 W/m <sup>3</sup>	$c/L_D$	1/128
$M \times M$	128x128	$s, s_0$	0.7, 1.0
$T_\infty$	0 K		

It was assumed that Gersborg-Hansen *et al.* used 40 iterations,  $s = 0.7$  and  $s_0 = 1.0$  (default values) for the MMA since these were not specified.

**Table 4.14:** Comparison of architectures from this study and the paper by Gersborg-Hansen *et al.*

Boundary	This study	Gersborg-Hansen <i>et al.</i>
Density distribution		

As seen in Table 4.14, the results from this study and those of the paper by Gersborg-Hansen *et al.* are similar, although they do not compare exactly. Comparing the two architectures, the thickness of the branches is the same but the architecture from this study has more secondary branches. Although there are differences, the main shape of the architecture is the same.

There are a couple of reasons for these differences. In the paper they use a Reuss harmonics average for the conductivity at the interface of two elements, which is different from the harmonic mean used in this study. They also do not specify the number of iterations they used or the values for  $s$  and  $s_0$ . There is also no mention of the error used while calculating the temperature distribution or the dual objective function of the MMA. They also used FEM with a filter to obtain the converged architecture.

#### 4.4 Summary

In this chapter, the two-dimensional code was validated and tested. It was found that the temperature distribution and adjoint method correlated perfectly with the theory. The MMA was validated against a paper by Gersborg-Hansen *et al* [8]. There were differences in the architectures, but the general shape was the same.

In this chapter, a useful dimensional temperature was also derived. The effect of the asymptotes was also tested in this chapter. It was found that the values of  $s$  and  $s_0$  have a significant impact on the maximum temperature. For the two-dimensional case,  $s = 0.9$  and  $s_0 = 0.1$  gave the best performance and were used in the rest of the two-dimensional study. A mesh-dependence study was also completed and  $M = 100$  was deemed sufficient for convergence in the temperature distribution.

The architectures obtained in this chapter have a tree-like shape. There are two main branches that extend to the corners of the top of the domain. Secondary branches are also present, more so at a higher element count.

The next chapter will discuss the implementation of the third dimension using a partial Dirichlet boundary condition as well as the methodology and results for the boundary condition.

## Chapter 5      **Three-Dimensional Methodology and Results for a Partial Dirichlet Boundary**

### **5.1 Introduction**

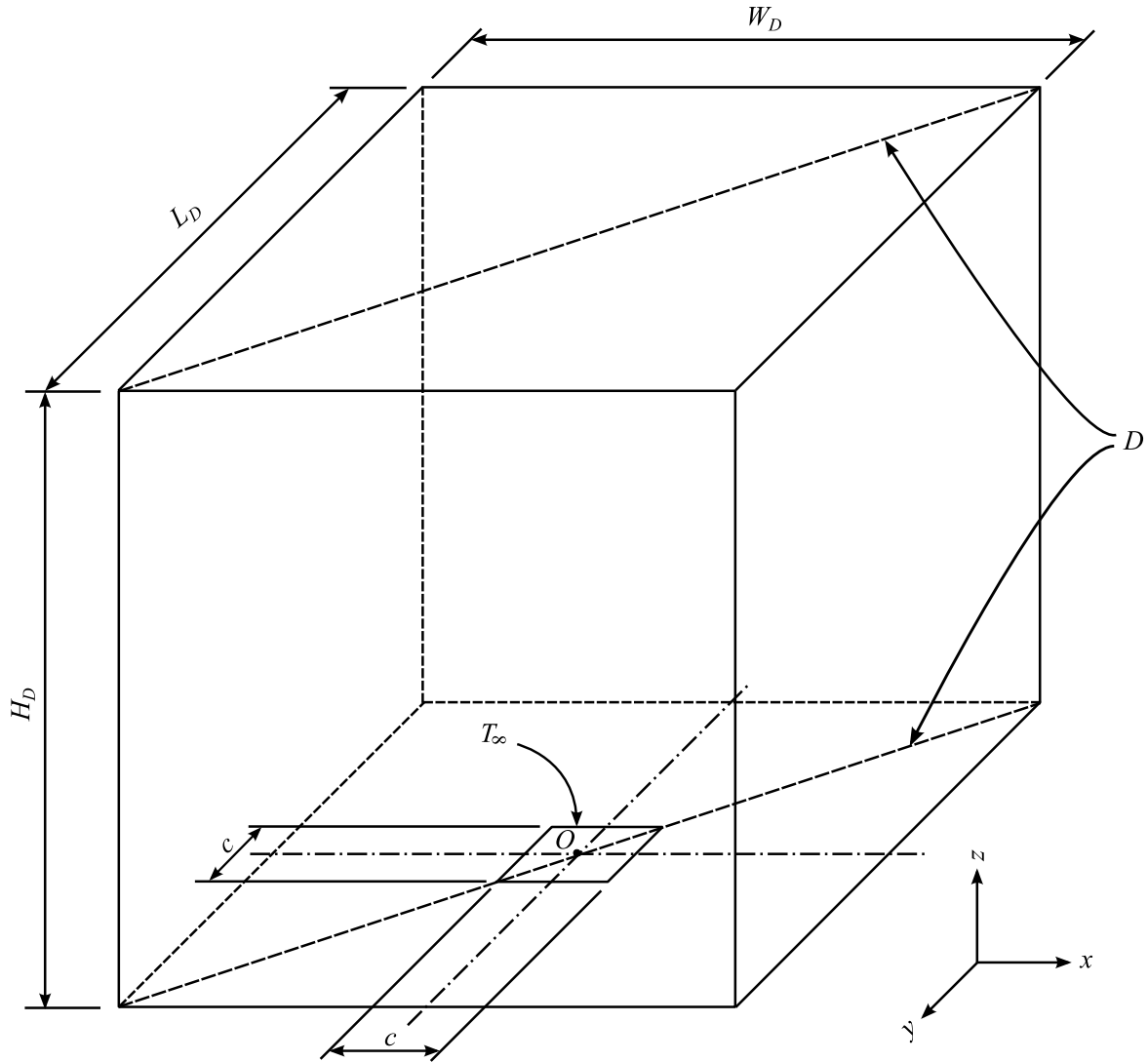
This section will discuss the implementation, methodology and results for a three-dimensional partial Dirichlet boundary. First, the effect of asymptotes will be checked, followed by mesh dependence, iteration dependence and the effect of constant vs. increasing penalisation. The effect of the initial density distribution and the size of the cold spot will also be investigated. Once these parameters have been tested, the effect of different conductivity ratios and volume constraints can be tested, which are the main parameters that can be changed in a practical situation.

### **5.2 Domain**

The domain for a partial Dirichlet boundary condition for the three-dimensional case is shown in Figure 5.1. The volume is divided into equal-sided elements, thus  $\delta_x = \delta_y = \delta_z$ . Each element again has an independent density,  $\theta_i$ , which is used to determine the conductivity and internal heat generation rate of that element. The diagonal section,  $D$ , is used for visualisation purposes and has coordinates  $(x, y, z) = ([-W_D/2, -L_D/2, 0], [-W_D/2, -L_D/2, H_D], [W_D/2, L_D/2, H_D], [W_D/2, L_D/2, 0])$  corresponding to the four corners of the section plane.

A Dirichlet boundary is where a boundary is set to a constant value, in this case, a constant temperature. For a partial Dirichlet boundary, only a small part of the boundary is set to a constant value as shown in the figure. The rest of the boundaries are selected to be adiabatic. Almost all other topology optimisation studies used a partial Dirichlet boundary. This is not necessarily a good choice of boundary condition, as will be discussed in the next chapter. It does, however, force the optimisation algorithm to produce a tree structure growing from this boundary, allowing for some control of the size and location of the main branch.

If a full Dirichlet boundary condition is used, depending on the initial choice of the material distribution, variations in the design variable sensitivities will not be seen in the  $x$ - $y$ -planes and the optimisation algorithm will in theory not be able to create a distinguishable cooling structure.



**Figure 5.1:** Domain for a partial Dirichlet boundary condition located on the bottom boundary, for a three-dimensional case.

## 5.3 Implementing the Third Dimension

### 5.3.1 Temperature Distribution

The purpose of the two-dimensional code was to build a basis from which the three-dimensional code could be developed. With this done, the model can be extended to three dimensions. This is done by adding the terms to the existing equation set to describe the third dimension. Here subscript  $t$  refers to the positive  $z$ -direction and subscript  $b$  refers to the negative  $z$ -direction. The general three-dimensional formula is thus obtained as shown in Eq. (5.1).

$$T_p a_p = T_w a_w + T_e a_e + T_n a_n + T_s a_s + T_t a_t + T_b a_b + S_u \quad (5.1)$$

The coefficients for the general three-dimensional formula are given in Table 5.1.



**Table 5.1:** Coefficients for the general three-dimensional formula for the internal nodes.

$a_w$	$a_e$	$a_n$	$a_s$	$a_t$	$a_b$	$a_p$
$\frac{k_l A_f}{\delta_x}$	$\frac{k_l A_f}{\delta_x}$	$\frac{k_l A_f}{\delta_y}$	$\frac{k_l A_f}{\delta_y}$	$\frac{k_l A_f}{\delta_z}$	$\frac{k_l A_f}{\delta_z}$	$a_w + a_e + a_n + a_s + a_t + a_b - S_p$

Since  $\delta_x = \delta_y = \delta_z = \delta$  the face area,  $A_f$ , is equal to  $\delta^2$  in all directions. This face area of  $\delta^2$  also applies to all boundaries. The source terms for different boundary conditions in three dimensions are shown below, which are unchanged from two dimensions except for the face area  $A_f$ :

*Fixed temperature:*

$$S_p = \frac{2k_p A_f}{\delta_{PN}} \quad (5.2)$$

$$S_u = \frac{2k_p A_f T_\infty}{\delta_{PN}} \quad (5.3)$$

*Heat flux:*

$$S_p = 0 \quad (5.4)$$

$$S_u = q'' A_f \quad (5.5)$$

*Adiabatic:*

$$S_p = 0 \quad (5.6)$$

$$S_u = 0 \quad (5.7)$$

*Internal heat generation:* The internal heat generation results now in a volumetric source, thus the heat generation must be multiplied with volume of each element:

$$S_u = q''' \delta^3 \quad (5.8)$$

Accept for the changes above, the thermal numerical FVM model does not change compared with the two-dimensional model. The conductivity matrix is, however, even more sparse now than with the two-dimensional model. Each row of the conductivity matrix will now have a maximum of seven non-zero entries. For a 100x100x100 mesh, there will be  $1 \times 10^{12}$  elements in the matrix with each row containing  $1 \times 10^6$  elements. This illustrates how sparse the conductivity matrix is for three dimensions and reiterates the importance of using a sparse matrix and sparse solver. For the three-dimensional case, the sparse iterative solver *lgmres* was used, since *spsolve* was not able to solve the temperature distribution in three dimensions.

### 5.3.2 Adjoint Method

The adjoint method does not change much either for three dimensions. It is only required to update the algorithm in respect of the changed layout of the conductivity matrix. The method explained in Section 3.3.2 is followed again for three dimensions. The only part that changes is the calculation of  $\partial K / \partial \theta_i$  because the addition of the third dimension now adds more coefficients that are dependent on  $\theta_i$ . For three dimensions, there are now a maximum of 19 non-zero entries in  $\partial K / \partial \theta_i$  instead of 13 as was the case in the two-dimensional domain.

### 5.3.3 Method of Moving Asymptotes

Since the MMA does not see the dimensions of the problem, it is easy to implement to three dimensions. The density matrix is a three-dimensional array for the three-dimensional case, thus only an extra logical loop is needed throughout the MMA algorithm to incorporate the additional levels of elements in the z-direction. For the three-dimensional domain, the *SQSD* was again used to solve the dual objective function.

## 5.4 Validation

### 5.4.1 Temperature Distribution

The same process as in Section 4.2.1 was used to validate the interaction of the conductivity values, converting the problem to a simple one-dimensional problem. All results correlated correctly with the theory. The general boundaries were once again validated with energy balance and all results correlated correctly.

To validate the boundaries, a three-dimensional case was set up in the commercial numerical code StarCMM+. All the important boundaries were tested, namely fixed temperature, adiabatic and heat flux. The whole volume was also heated using internal heat generation. The temperature distribution compared correctly with the StarCCM+ results within an error of  $1 \times 10^{-6}$ .

### 5.4.2 Adjoint Method

As explained in Section 4.2.2, the adjoint method can be validated with finite difference. This process was followed again for three dimensions, and all the sensitivities compared well with the finite difference results. The average difference was less than  $1 \times 10^{-6}$ .

## 5.5 Methodology

The three-dimensional optimisation routine used the same methodology as shown in Figure 3.3 with the changes outlined in Section 5.3.1 - 5.3.3. This entailed implementing the third dimension in the temperature distribution, adjoint method and MMA. Before the actual optimisation runs could begin, varying the conductivity ratio and volume constraint, the optimal running conditions (number of nodes needed, number of iterations needed, effect of penalisation, etc.) first had to be found. These conditions are outlined in Sections 5.5.2 - 5.5.7.

### 5.5.1 Dimensionless Maximum Temperature

In the two-dimensional simulations, a normalised maximum temperature was derived, which used all the practical input parameters to normalise the maximum temperature. The effect of  $k_L$ ,  $q_H$ ,  $T_\infty$  and the dimensions were again evaluated using the same tests as in Section 4.3.1. It was found that the definition of the normalised maximum temperature does not need to be changed for three dimensions, for the partial Dirichlet boundary. The normalised temperature definition is repeated in Eq. (5.9).

$$\tau = \frac{(T_{max} - T_\infty)k_L}{q_H L_D^2} \quad (5.9)$$

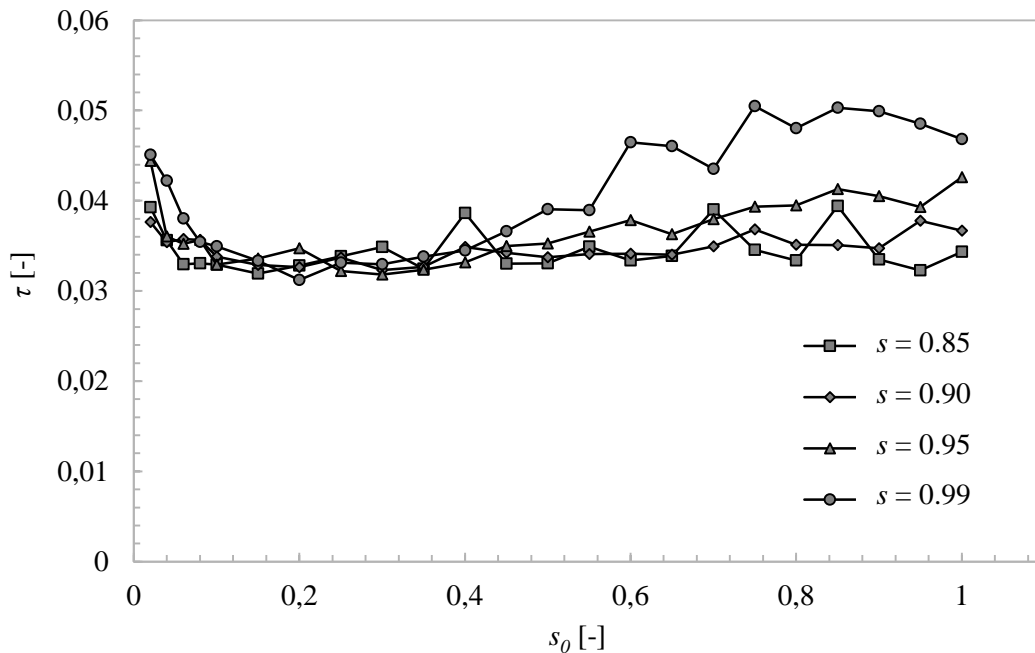
### 5.5.2 Effect of the Asymptote Parameters, $s$ and $s_0$

As was seen with the two-dimensional optimisations, the method with which the MMA asymptotes were calculated had a huge impact in some cases. Thus, for the three-dimensional domain, the effect of the asymptotes was once again checked. The value of  $s$  was considered between 0.85 and 0.99, ( $s$  must be smaller than unity). The value of  $s_0$  was ranged between 0.05 and 1.0 (the latter being the default value). The optimisation runs were done initially with a coarse mesh of 20x20 x20 elements, just to get an idea of the results.

The input parameters for these simulations are given in Table 5.2.

**Table 5.2:** Input parameter for the effect of the asymptotes for a three-dimensional domain with a partial Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	40 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.05
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M \times M \times M$	20x20x20		



**Figure 5.2:** The effect of  $s_0$  and  $s$  on  $\tau$  for a three-dimensional domain partial Dirichlet boundary.

Figure 5.2 shows the effect of  $s$  and  $s_0$  with regard to  $\tau$ .  $s = 0.85$  and  $s = 0.9$  perform well for all values of  $s_0$ , except for the lower range of  $s_0$ . For  $s = 0.95$  and  $s = 0.99$ ,  $\tau$  diverges after  $s_0 = 0.4$ . There is once again an optimal range for  $s_0$  in the region of 0.15 – 0.25, for all values of  $s$ .

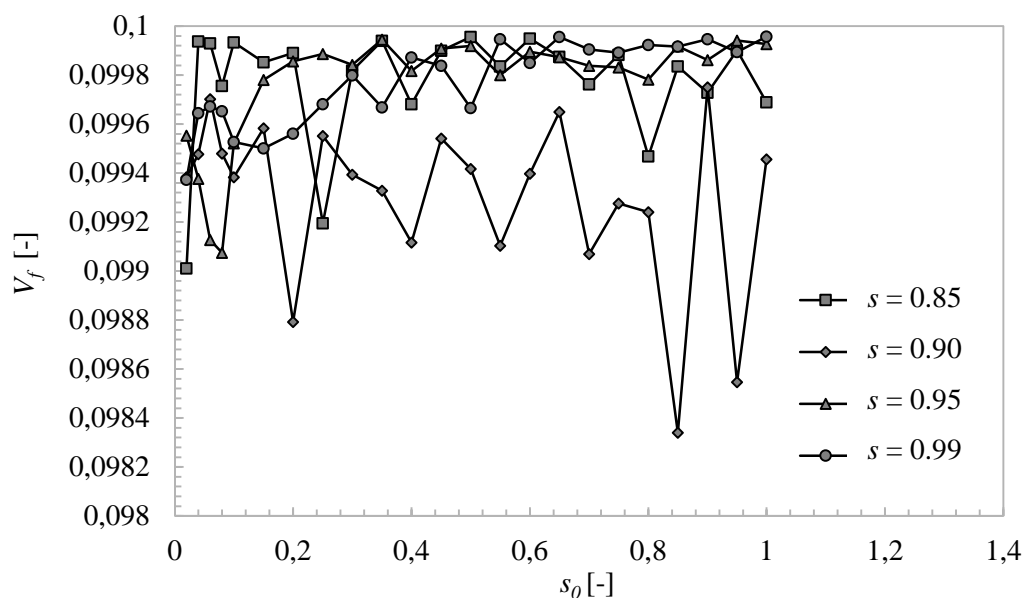
Table 5.3 shows some of the optimised architectures obtained with  $s_0$  values of 0.02, 0.15, 0.5 and 1.0 and  $s$  values of 0.85, 0.9, 0.95 and 0.99. As seen in the table, an increase in  $s$  relaxes the solution as

seen with the increase in grey densities. The problem with this is that it generates more grey areas, as seen with  $s = 0.95$  and  $0.99$ . Small values of  $s_0$  also generate excessive grey areas,  $s_0$  values of  $0.15$  give good (0-1) architectures with primary branches extending to the end of the domain, for all values of  $s$ . For higher values of  $s_0$ , the architectures are constrained and in some cases discontinuous. Although the architectures for  $s = 0.85$  are not necessarily the best performing (although not by much), they produce little grey areas and are easier to manufacture.

It is important to emphasise that the architectures obtained in Table 5.3 are shown for the diagonal section plane  $D$ . For the architectures observed, the main branches extend to the corners of the section plane, which corresponds to two opposite top corners of the three-dimensional domain. If the section plane was changed to the other diagonal as viewed from an  $x$ - $y$ -perspective, the resulting architecture would be the same because of the symmetry in the boundary conditions and thus symmetry in the architecture. In the following sections, some three-dimensional representations will be shown, although not here due to the low element count.

The effect of the asymptotes was evaluated at one set of operating conditions ( $k^*$ ,  $V^*$  etc.). The results obtained in this section could change with different values of  $k^*$  and  $V^*$ . There is room for further investigation in this area.









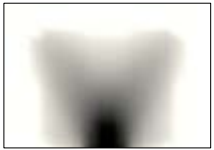







It was found that  $\tau$  was at its lowest when  $s_0$  is in the region of  $0.15$  to  $0.25$  irrespective of the value of  $s$ . It was also found that with  $s = 0.85$ , the most stable operation of the MMA algorithm was achieved irrespective of the value of  $s_0$ . This is also coincidentally close to the recommended value in the MMA paper by Svanberg, of  $\sqrt{s}$ . The value of  $s = 0.85$  was used in all further investigations unless specified otherwise. The default value of  $s_0 = 1.0$  was also included for comparative purposes. Routine rechecking of the above findings was conducted throughout the study.



**Figure 5.3:** The effect of  $s_0$  and  $s$  on the converged volume ratio for a three-dimensional domain with a partial Dirichlet boundary.

The influence of  $s_0$  and  $s$  on the achieved converged volume ratio was also checked. Figure 5.3 shows that when  $s = 0.9$ , a large scatter in the converged volume ratio was obtained over the  $s_0$  range. Less scatter was observed for the other values of  $s$ . There are minor differences, but these were relatively small.

**Table 5.3:** Architectures as viewed on the diagonal section plane  $D$  for different values of  $s$  and  $s_0$  with  $k^* = 500$  and  $V = 0.1$ .

$s$ [-]	0.85	0.85	0.85	0.85
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$3.927 \times 10^{-2}$	$3.563 \times 10^{-2}$	$3.293 \times 10^{-2}$	$3.307 \times 10^{-2}$
Density distribution				
$s$ [-]	0.9	0.9	0.9	0.9
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$3.763 \times 10^{-2}$	$3.540 \times 10^{-2}$	$3.575 \times 10^{-2}$	$3.564 \times 10^{-2}$
Density distribution				
$s$ [-]	0.95	0.95	0.95	0.95
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$4.440 \times 10^{-2}$	$3.602 \times 10^{-2}$	$3.521 \times 10^{-2}$	$3.565 \times 10^{-2}$
Density distribution				
$s$ [-]	0.99	0.99	0.99	0.99
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$4.509 \times 10^{-2}$	$4.220 \times 10^{-2}$	$3.803 \times 10^{-2}$	$3.544 \times 10^{-2}$
Density distribution				

### 5.5.3 Mesh-Dependence Study

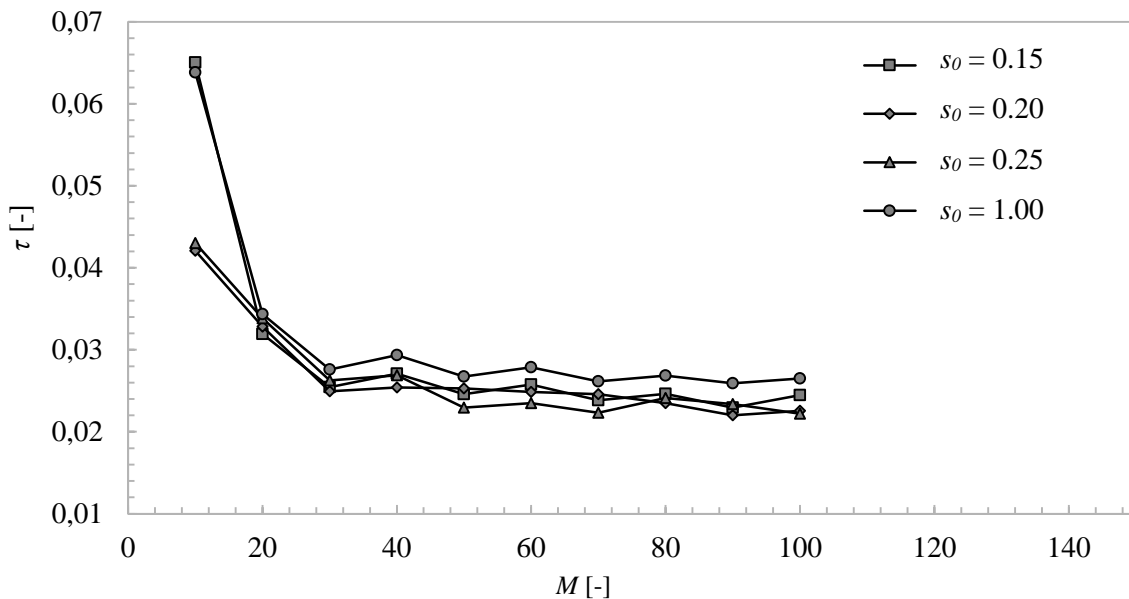
As in all CFD simulations, a mesh-dependence study is required to determine the minimum number of elements required for the optimisation scheme. For the two-dimensional domain, this was found to be 100x100 elements. For the three-dimensional domain, the mesh-dependence study was done for mesh densities ranging from 10x10x10 to 100x100x100 elements.

The optimisation runs were conducted at the same conditions and number of iterations. See Table 5.4. It can be argued that more nodes will need more iterations, but all the runs converged after 40 iterations when the increasing intensity of penalisation was used. The convergence of an optimisation run was checked by plotting  $\tau$  for each MMA iteration. The mesh-dependence study also showed that  $s = 0.85$  and  $s_0 = 0.25$  gave the best results.

**Table 5.4:** Input parameters for the mesh dependence for a three-dimensional partial Dirichlet boundary.

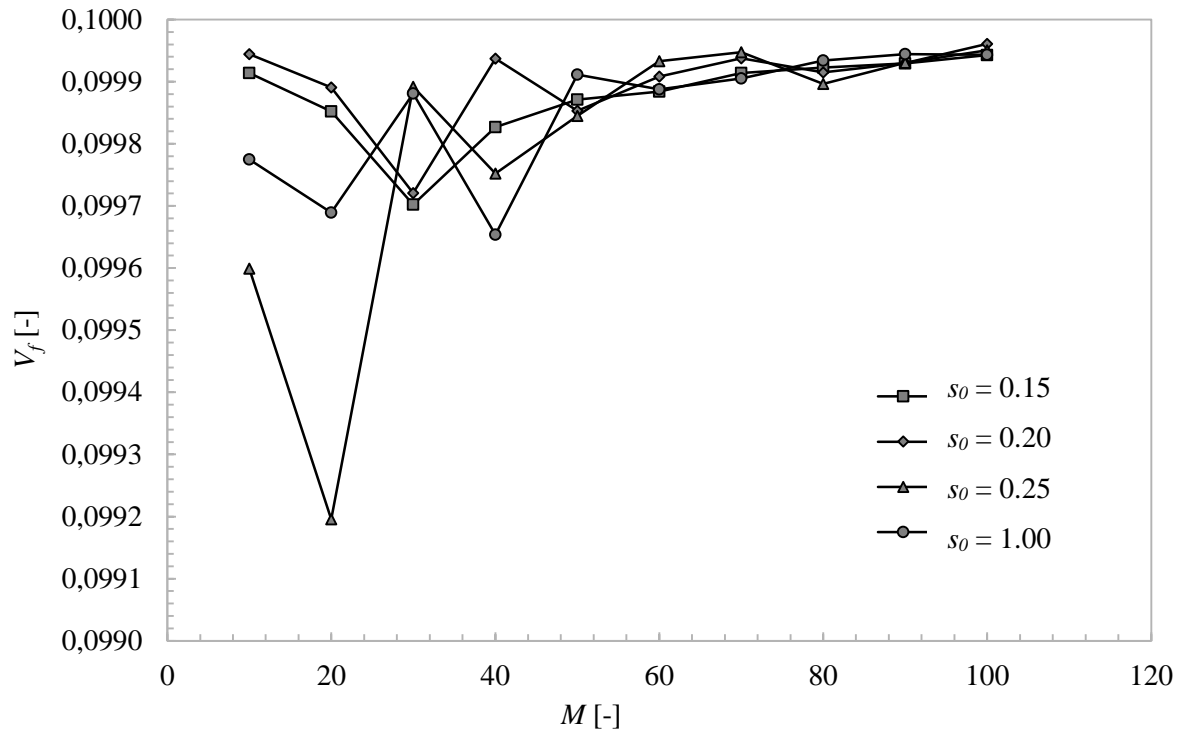
Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	40 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M \times M \times M$	10x10x10 to 100x100x100		

As seen in Figure 5.4, a grid density of higher than 50x50x50 does not yield a significantly lower maximum temperature. The difference in  $\tau$  between 50x50x50 elements and 100x100x100 elements is only 3.38% for  $s_0 = 0.25$ . This is an eight-fold increase in the total number of nodes. For the same value of  $s_0$ , 60x60x60 and 80x80x80 elements performed worse than 50x50x50 elements.



**Figure 5.4:** Mesh-dependence study for a three-dimensional domain using a partial Dirichlet boundary.

For this reason, a mesh density of 50x50x50 elements was chosen for the remainder of the three-dimensional partial Dirichlet boundary cases. It was assumed that this element count would be sufficient for different values of  $k^*$  and  $V^*$ . It is also interesting to note the huge effect of  $s_0$  and  $s$  for a grid size of 10x10 elements. Shown in Figure 5.5, the converged volume ratio approaches the constraint of 0.1 as the element size is decreased (increased mesh density). This makes sense since the smaller nodes can more accurately describe the temperature distribution and the sensitivities in the domain.



**Figure 5.5:** The converged volume ratio for the mesh-dependence study for a three-dimensional domain with a partial Dirichlet boundary.

Table 5.5 shows the converged architectures for the mesh-dependence study for  $M = 10$  to 100. Up to  $M = 40$ , there are still significant grey areas in the architecture. After  $M = 40$ , the V shape is evident in the primary branches. As explained earlier, this shape is the same for the other diagonal, which means there are four main branches extending to the top four corners of the three-dimensional domain.

As  $M$  is increased, the number of secondary branches increases, but in essence the architectures were already visible from  $M = 20$  upwards. The architectures also show that for a lower element count, the secondary branches are horizontal. As the element count is increased, the secondary branches extend upwards to the corners of the domain. The convergence in  $\tau$  seen in Figure 5.4 correlates with the convergence in the architecture after  $M = 50$  elements. There are also some discontinuous branches observed for  $M = 70$  and  $M = 90$ . The reason for this is that the branches are not in the plane of the diagonal.

**Table 5.5:** Architectures for the mesh dependence for  $s_0 = 0.15$ .











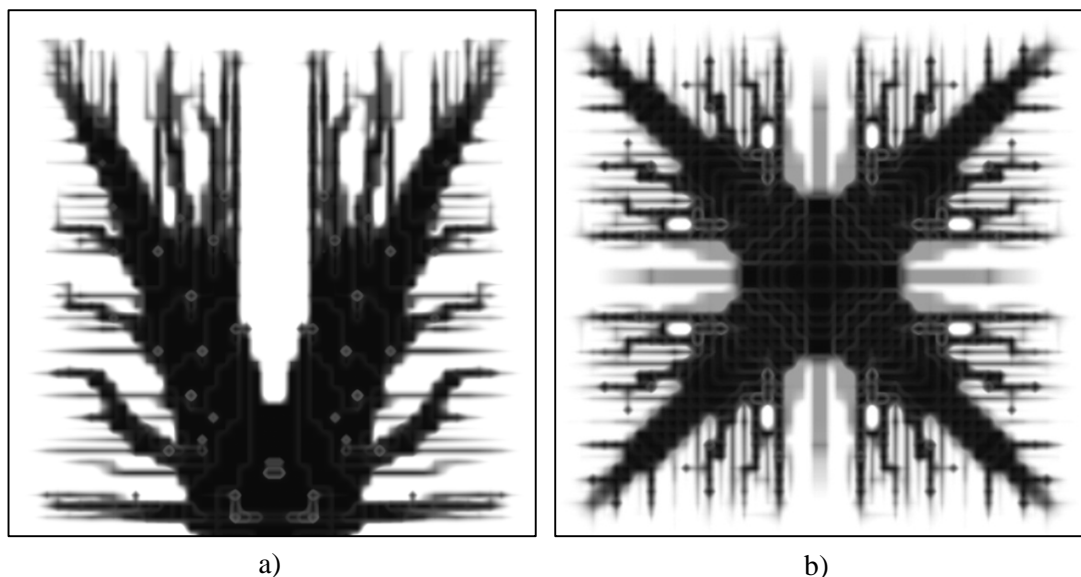
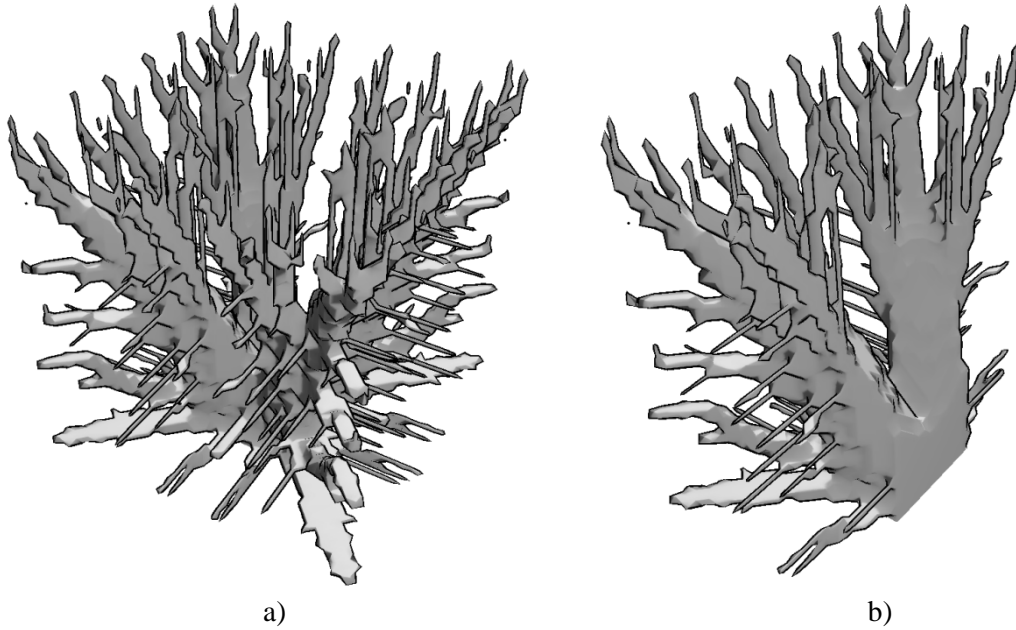
$M$ [-]	10	20	30	40
$\tau$ [-]	$6.503 \times 10^{-2}$	$3.193 \times 10^{-2}$	$2.543 \times 10^{-2}$	$2.710 \times 10^{-2}$
Density distribution				
$M$ [-]	50	60	70	80
$\tau$ [-]	$2.457 \times 10^{-2}$	$2.576 \times 10^{-2}$	$2.384 \times 10^{-2}$	$2.462 \times 10^{-2}$
Density distribution				
$M$ [-]	90	100		
$\tau$ [-]	$2.295 \times 10^{-2}$	$2.449 \times 10^{-2}$		
Density distribution				

Figure 5.6 shows a side and top view of the architecture obtained for  $M = 50$  in the mesh-dependence study. The figures better illustrate how the main branches of the architecture extend to the outer corners of the domain. As seen in the top view, the secondary branches grow normal to the sides of the domain, but the secondary branches also grow upwards when seen from the side. Figure 5.7 shows an isometric view of the architecture.



**Figure 5.6:** Shown in a) a side view of the architecture b) a top view of the architecture ( $M = 50$ ,  $s_0 = 0.15$ ).





**Figure 5.7:** An isometric view showing an isosurface<sup>2</sup> of the converged architecture for the  $M = 50$  case from the mesh-dependence study for a) full domain b) half of the domain  $y$ .

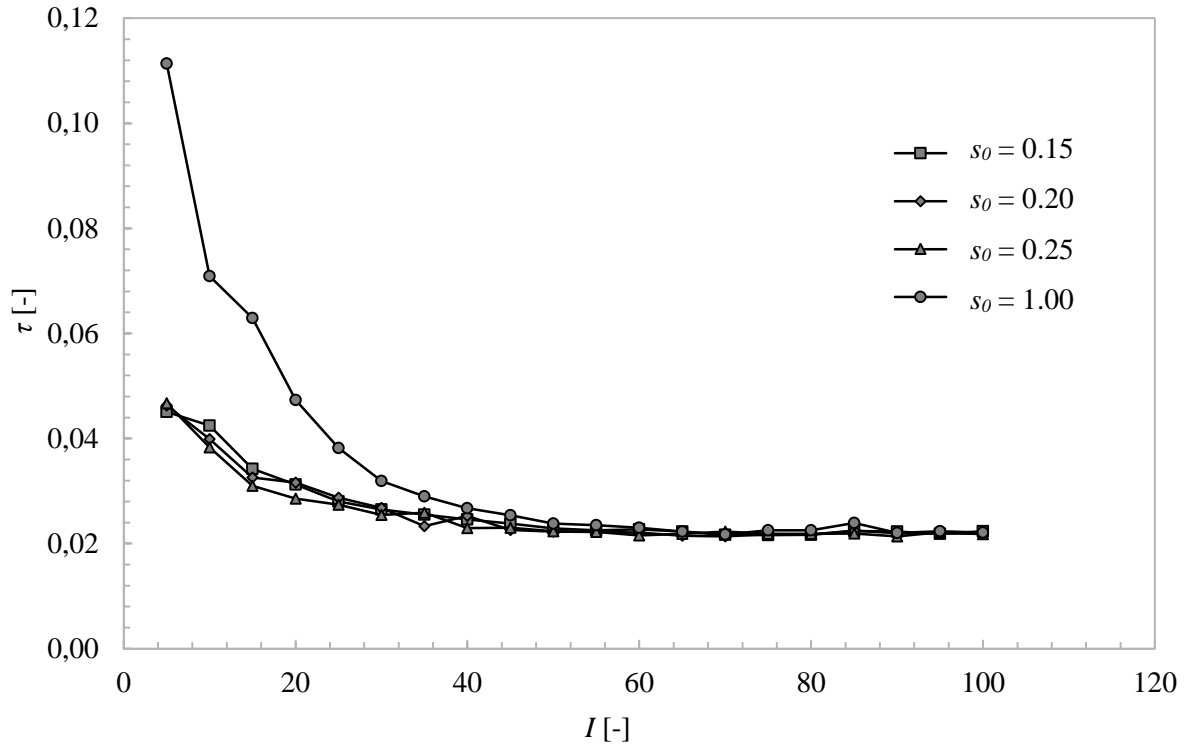
#### 5.5.4 Iteration-Dependence Study

In the mesh-dependence study, it was found that 40 MMA iterations and  $M = 50$  gave suitable results. In this section, the number of iterations is considered for  $I$  between 5 and 100 for the four cases of  $s_0$ . Converged  $\tau$  values and architectures are given in Table 5.7 and Figure 5.8 respectively. The results showed that there was a small drop in the maximum temperature as  $I$  is increased to 60 iterations. Above 60 iterations, the results oscillated slightly indicating that  $I$  does not have an influence. All values of  $s_0$  gave good results, except  $s_0 = 1.0$ . The input parameters for the iteration-dependence are shown in Table 5.6.

**Table 5.6:** Input parameters for the number of iterations for a three-dimensional partial Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	5 – 100 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M \times M \times M$	50x50x50		

<sup>2</sup> The isosurface is extracted from the three-dimensional scalar field at a specific density value using the marching cubes algorithm.



**Figure 5.8:** The effect of the number of iterations on  $\tau$  for a three-dimensional domain with a partial Dirichlet boundary.

As seen from Table 5.7, low iterations yield a poorly defined architecture. The grey areas reduce as the iterations are increased and after about 40 iterations, the grey areas are minimal, which is important for this study. There is also no real change in the architecture after  $I = 60$ , only small changes in the secondary branches. This explains the convergence of  $\tau$  as shown in Figure 5.8. The difference in  $\tau$  between  $I = 60$  and  $I = 100$  is only 1.18% for  $s_0 = 0.25$ , which is acceptable. Thus  $I = 60$  is sufficient.

As observed in the mesh-dependence study, for a low count in the iterations, the secondary branches are again horizontal, however, with an increase in the iteration count, the secondary branches again start to grow upwards to the corners of the domain.

**Table 5.7:** Architectures for the iteration-dependence study for  $s_0 = 0.15$ .











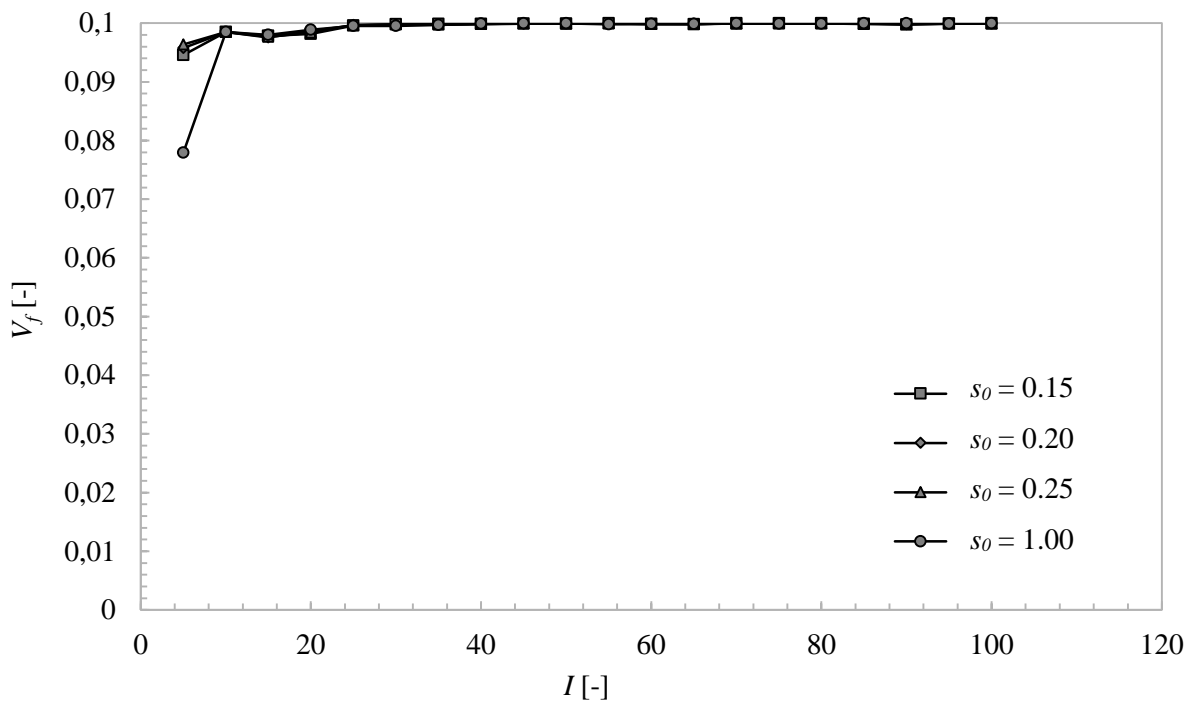
$I$ [-]	10	20	30	40
$\tau$ [-]	$4.243 \times 10^{-2}$	$3.422 \times 10^{-2}$	$3.124 \times 10^{-2}$	$2.801 \times 10^{-2}$
Density distribution				
$I$ [-]	50	60	70	80
$\tau$ [-]	$2.290 \times 10^{-2}$	$2.252 \times 10^{-2}$	$2.268 \times 10^{-2}$	$2.225 \times 10^{-2}$
Density distribution				
$I$ [-]	90	100		
$\tau$ [-]	$2.231 \times 10^{-2}$	$2.186 \times 10^{-2}$		
Density distribution				

Figure 5.9 shows the effect of the iterations on the converged volume ratio. This curve also shows a nice convergence with the volume ratio, as expected with the increase in the iterations.

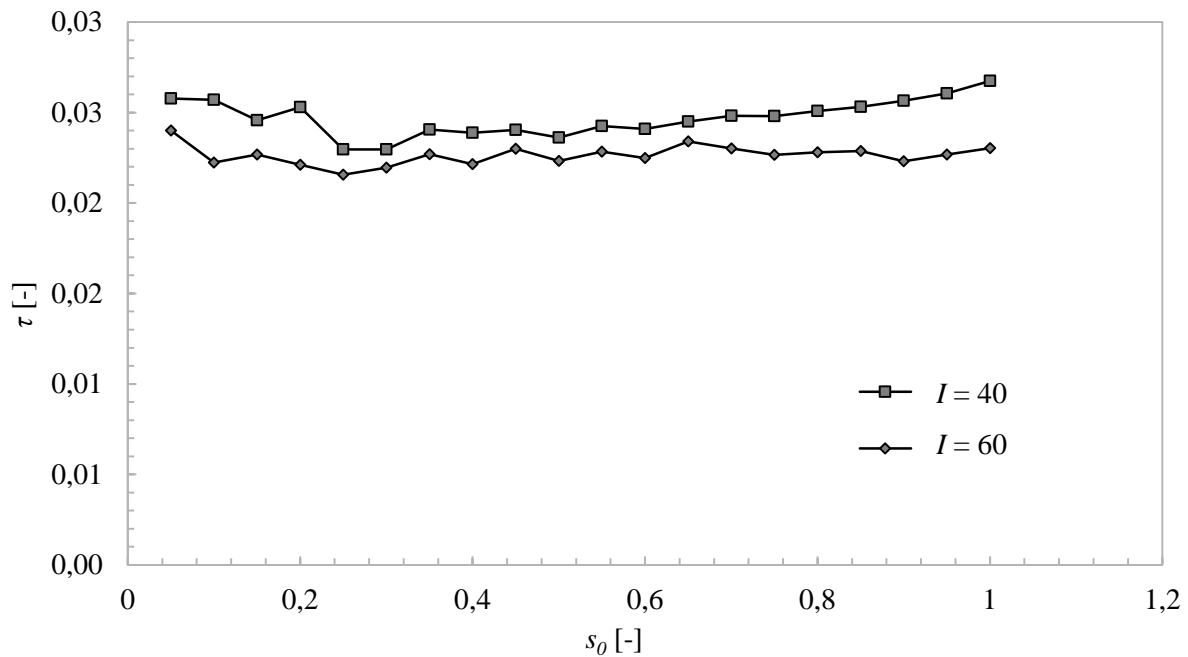


**Figure 5.9:** The effect on the volume ratio due to the number of iterations for a three-dimensional domain with a partial Dirichlet boundary.

**Table 5.8:** Input parameters for effect of asymptotes for a three-dimensional domain with a partial Dirichlet boundary.

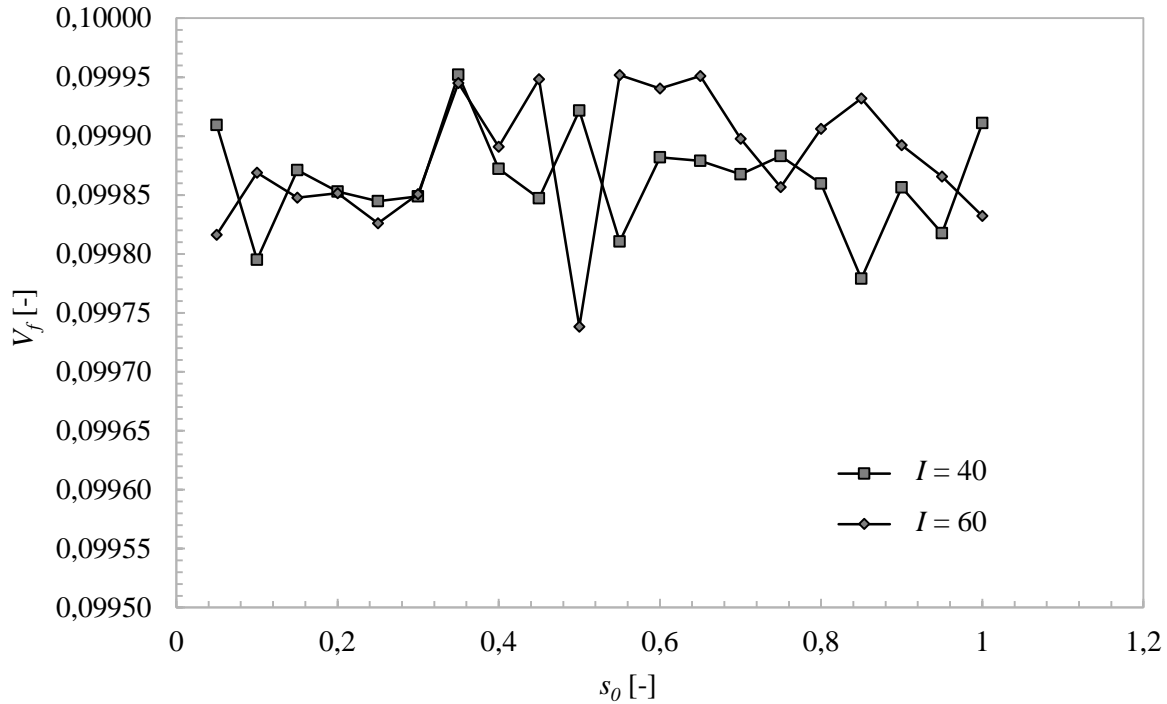
Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	40, 60
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M_x M_x M_x$	50x50x50		

Since the ideal mesh size and number of iterations needed were now known, the influence of  $s$  and  $s_0$  was rechecked with  $M = 50$  and  $I$  either 40 or 60. As seen in Figure 5.10, the optimal  $s_0$  values still lay between 0.15 and 0.25, but for 60 iterations, the algorithm operation is more stable than with 40 iterations. This can be attributed to a smaller increment in the penalisation level from one MMA iteration to the next. The input parameters are shown in Table 5.8.



**Figure 5.10:** The effect of the  $s_0$  and  $I$  on  $\tau$  for a three-dimensional domain with a partial Dirichlet boundary.

The influence of the converged volume is not as clear as can be seen in Figure 5.11. The values exhibit the same level of scatter for both 40 and 60 iteration cases, but because the  $V^*$  scale is very small, this is not a problem.



**Figure 5.11:** The effect of the asymptotes and  $I$  on the volume ratio for a three-dimensional domain with a partial Dirichlet boundary.

The results for  $s$  and  $s_0$  obtained using  $I = 60$  reaffirm the choice for the  $s_0$  range chosen in Section 5.5.2.

### 5.5.5 Constant Penalisation vs. Incremental Increasing Penalisation

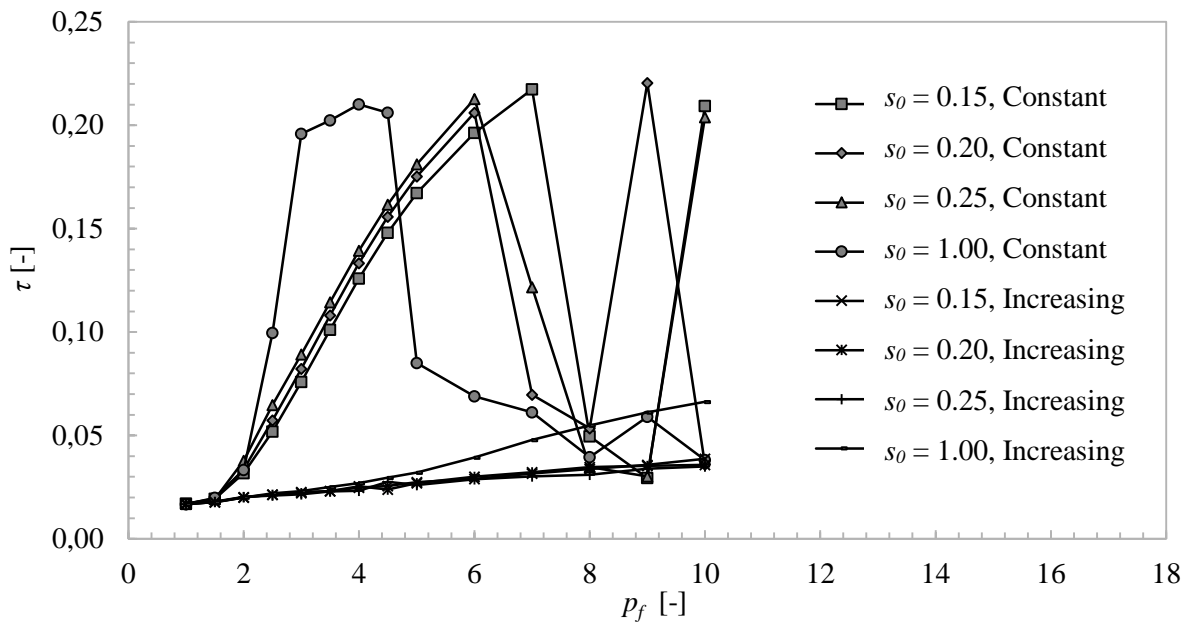
Penalisation is used to give better 0-1 solutions and minimise the amount of grey areas (composite material). There are two kinds of penalisations, constant and incremental increasing penalisation. With constant penalisation, the sensitivities required in the MMA algorithm are penalised in each iteration by the same value of the penalisation factor. In incremental increasing penalisation, the penalisation factor is started at 1.0 (no penalisation) for the first iteration, and uniformly increased to its final value at the last MMA iteration.

In most other cases, penalisation factors of 2 or 3 are used [8], [9]. To get a better idea of the effect of penalisation, the penalisation factor was checked for values between 1 and 5 in increments of 0.5 and between 5 and 10 in increments of 1. The optimal penalisation factor would most probably lie in the region of 2 and 5, which is the reason for the smaller increments.

As seen in Figure 5.12, an increase in  $p$  generally results in an increase in  $\tau$ . The results for the increasing penalisation scheme are almost identical for all values of the  $s_0$ , except for  $s_0 = 1.0$ , which exhibits a steeper increase. The results for the increasing penalisation scheme are as expected, where the maximum temperature increases with an increase in the penalisation factor. This is due to the black/white architecture of the higher penalisation factors. The penalisation removes the grey areas, which actually helps considerably with the decrease in the temperature distribution. The input parameters are shown in Table 5.9.

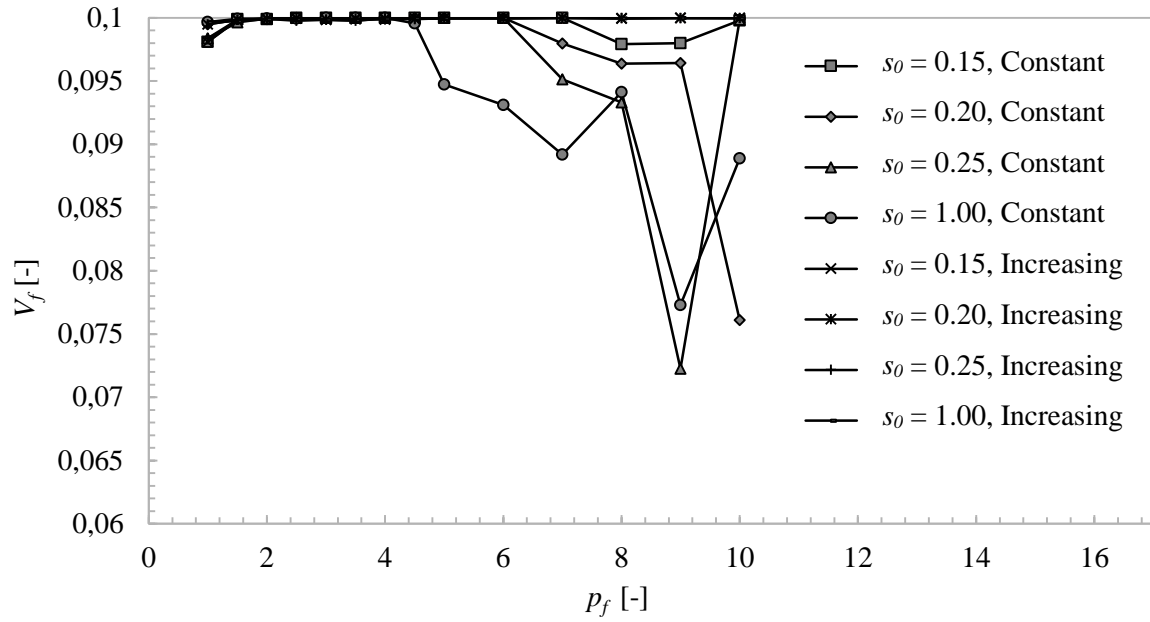
**Table 5.9:** Input parameters for constant and incremental increasing penalisation for a three-dimensional domain with a partial Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing and constant
$W_D$	1.0 m	$p_f$	1 - 10
$H_D$	1.0 m	$l$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M_x M_x M$	50x50x50		



**Figure 5.12:** The effect of constant and increasing penalisation on  $\tau$  for a three-dimensional domain with a partial Dirichlet boundary.

The trends in the  $\tau$  results for the constant penalisation scheme are not as promising, and significantly higher  $\tau$  values are observed when compared with the incremental increasing penalisation scheme. All values for  $s_0$  show an increase in the maximum temperature with an increase in  $p$ , but for constant penalisation, the gradient of the increase is much higher. Also, after  $p_f = 5$ , the constant penalisation scheme results in  $\tau$  starting to diverge.



**Figure 5.13:** The effect of constant and increasing penalisation on the volume ratio for a three-dimensional domain with a partial Dirichlet boundary.

As shown in Figure 5.13, the volume constraint is well satisfied for penalisation factors between 2.5 and 5. The increasing scheme cases all followed this pattern and the volume constraint was better satisfied with an increase in  $p$ . The constant penalisation gave similar results up to  $p_f = 5$ , after which the solution became unstable as shown in Figure 5.13. Looking at the results, it is evident that incremental increasing penalisation is superior to constant penalisation. It was decided to use  $p_f = 3$  for the rest of the results since it gave a good combination of the results between maximum temperature and volume constraint.

Table 5.10 shows the architectures for different constant penalisation factors. Up to  $p_f = 2$ , there are still some grey areas in the volume. After  $p_f = 2.5$ , the grey areas diminish, but as seen from the architectures, constant penalisation is not a valid method for topology optimisation. For  $p_f = 1.5$ , there is still a reasonable tree structure. After  $p_f = 2$ , the architectures shrink into a three-dimensional hemispheric form, which continues to shrink as  $p$  increases. For high  $p$  values, there is no discernible structure anymore. This concentration of the material around the cold spot is the reason for the high maximum temperature of the high constant penalisation factors.

**Table 5.10:** Architectures as seen on the diagonal section  $D$  for constant penalisation with  $s_0 = 0.15$ .











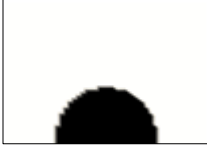

















$p_f$ [-]	1	1.5	2.0	2.5
$\tau$ [-]	$1.704 \times 10^{-2}$	$1.932 \times 10^{-2}$	$3.181 \times 10^{-2}$	$5.197 \times 10^{-2}$
Density distribution				
$p_f$ [-]	3.0	3.5	4.0	4.5
$\tau$ [-]	$7.584 \times 10^{-2}$	$1.011 \times 10^{-1}$	$1.259 \times 10^{-1}$	$1.480 \times 10^{-1}$
Density distribution				
$p_f$ [-]	5.0	6.0	7.0	8.0
$\tau$ [-]	$1.672 \times 10^{-1}$	$1.963 \times 10^{-1}$	$2.174 \times 10^{-1}$	$4.957 \times 10^{-2}$
Density distribution				
$p_f$ [-]	9.0	10.0		
$\tau$ [-]	$2.945 \times 10^{-2}$	$2.094 \times 10^{-1}$		
Density distribution				

Table 5.11 shows converged architectures for increasing penalisation factors. There are again grey areas up to  $p = 2$ , after which the grey areas start to diminish. From  $p_f = 2.5$  to  $p_f = 5$ , there is no real change in the architecture, only in the maximum temperature, which increases with  $p$ . The difference is seen in the secondary branches, as  $p$  is increased, the number of secondary branches increases. From  $p_f = 6$  to  $p_f = 10$ , the distance between the main branch tips and the corners of the domain increases and the main branches also grow in terms of width. These architectures start to resemble the architectures from constant penalisation after  $p_f = 2$  as shown in Table 5.10.



**Table 5.11:** Architectures as seen on the diagonal section  $D$  for increasing penalisation with  $s_0 = 0.15$  for a three-dimensional domain with a partial Dirichlet boundary.

$p_f$ [-]	1	1.5	2.0	2.5
$\tau$ [-]	$1.704 \times 10^{-2}$	$1.774 \times 10^{-2}$	$2.001 \times 10^{-2}$	$2.140 \times 10^{-2}$
Density distribution				
$p_f$ [-]	3.0	3.5	4.0	4.5
$\tau$ [-]	$2.268 \times 10^{-2}$	$2.308 \times 10^{-2}$	$2.483 \times 10^{-2}$	$2.558 \times 10^{-2}$
Density distribution				
$p_f$ [-]	5.0	6.0	7.0	8.0
$\tau$ [-]	$2.746 \times 10^{-2}$	$2.893 \times 10^{-2}$	$3.159 \times 10^{-2}$	$3.377 \times 10^{-2}$
Density distribution				
$p_f$ [-]	9.0	10.0		
$\tau$ [-]	$3.572 \times 10^{-2}$	$3.867 \times 10^{-2}$		
Density distribution				

## 5.5.6 Initial Density Distribution

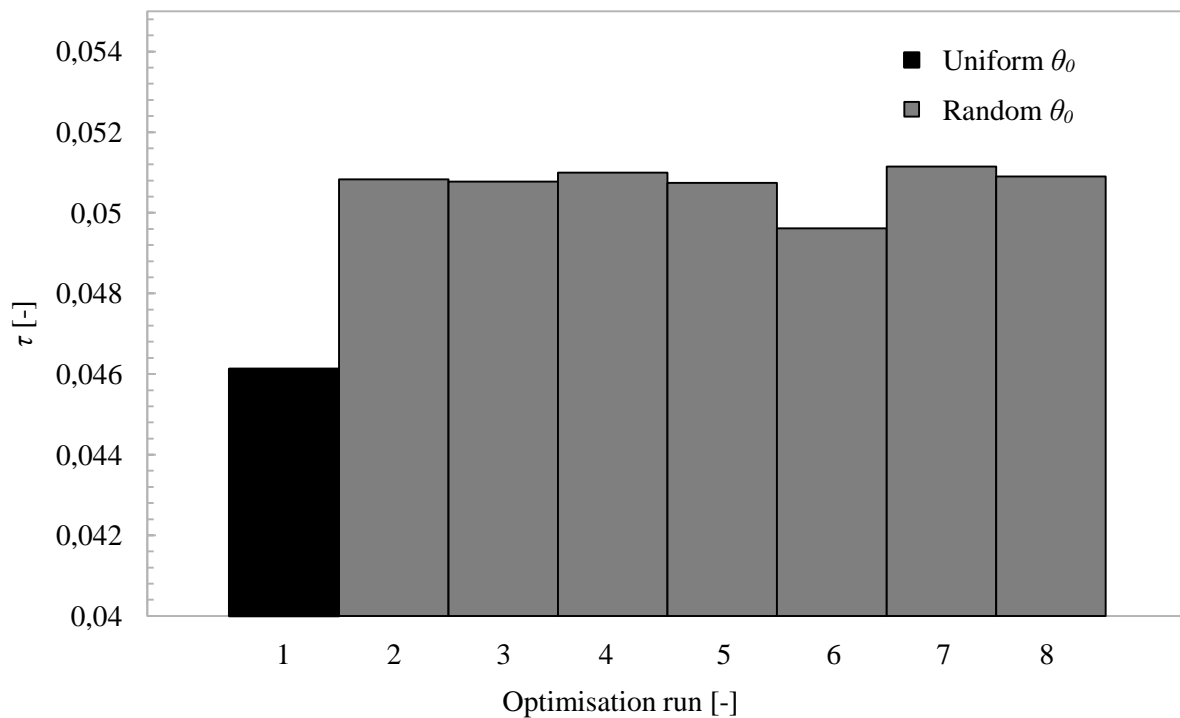
One of the inputs of the MMA requires an initial guess to the variable that is being optimised. This density distribution is updated after each MMA iteration with the solution of the previous iteration. This section investigates the effect of this initial density distribution.

### 5.5.6.1 Random Initial Density Distribution

It was already found that an initial guess for the density distribution that exceeds the volume constraint does not work. To determine how sensitive the algorithm is to an initial guess, a random initial density matrix, which still complies with the volume constraint, was used. The input parameters are shown in Table 5.12.

**Table 5.12:** Input parameters for the random initial density distribution using a three-dimensional domain with a partial Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$l$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.05 (5%)
$k_H$	1000 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M_x M_x M$	50x50x50		







**Figure 5.14:** The effect on  $\tau$  when using a random initial density distribution (2 – 8) compared with an even initial density distribution (1).

Figure 5.14 shows that using a random initial guess for the density distribution is detrimental to the maximum temperature. Runs 2 – 8 show the results for a random initial density distribution and Run 1 shows the result for an even density distribution. There is a maximum increase of 10% in  $\tau$  when using a random initial guess for the density distribution.

The average maximum dimensionless temperature is  $5.072 \times 10^{-2}$  with a minimum of  $4.961 \times 10^{-2}$  and a maximum of  $5.115 \times 10^{-2}$ . There is thus a variation from the average of 0.85% and 2.2% for the maximum and minimum respectively. This is a small difference but the real difference is seen in the density distribution as indicated in Table 5.13. The density distribution is no longer symmetrical, which means the algorithm is very sensitive to the initial guess. Note that the distributions are plotted for diagonal  $D$  only. Disconnected portions are due to branches cutting through the diagonal.

**Table 5.13:** Architectures for an even distribution (1) and random distribution (2, 7-8) as seen on the diagonal slice  $D$  for a three-dimensional domain with a partial Dirichlet boundary

Run [-]	1	2	7	8
$\tau$ [-]	$4.612 \times 10^{-2}$	$5.083 \times 10^{-2}$	$5.115 \times 10^{-2}$	$5.090 \times 10^{-2}$
Density distribution				

### 5.5.6.2 $p = 1$ Solution

As seen in the previous section, when using a random initial guess, the MMA is very sensitive to the given initial density distribution. Section 5.5.5 showed that using  $p = 1$ , gives the lowest maximum temperature, but produces only a slightly distinguishable tree structure with mostly grey areas, which would be very difficult to manufacture, as shown in Table 5.10. The purpose of this section is to use the converged architecture (with  $p = 1$  and constant penalisation) as an initial density distribution for an optimisation run with incremental penalisation and  $p = 3$ . The input parameters are given in Table 5.14.


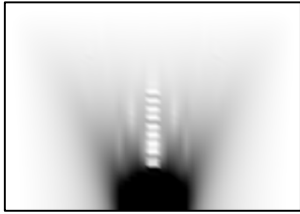
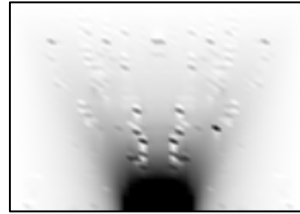






**Table 5.14:** Input parameters for  $p = 1$  initial distribution.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing and constant
$W_D$	1.0 m	$p_f$	1 or 3
$H_D$	1.0 m	$I$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.1 (10%)
$k_H$	100.0, 200.0, 1000.0 and 2000.0 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M_x M_x M$	50x50x50		

Table 5.15 shows converged architectures with different initial density distributions and penalisation factors. The first row shows an initial density distribution with constant penalisation ( $p = 1$ ). There are naturally many grey densities due to the penalisation factor of 1. This architecture produces the lowest maximum temperature, but is not practical to manufacture. The second row uses the converged densities from the first row as an initial guess for the density distribution in the MMA. Using this initial distribution with increasing penalisation and  $p_f = 3$  produces a tree structure.

The third row uses a uniform density distribution and increasing penalisation with  $p_f = 3$ , as done with the tests in the previous sections. This also produces a tree-like structure as seen in the table. Comparing the second and third row, which both use  $p_f = 3$  but different initial density distributions, the results are similar. The shape of the main branches is similar, but for the results of a uniform initial density distribution, the main branches are thicker. The number of secondary branches is the same for both cases, although there are small differences in the shape of the branches. On average, using the solution of  $p_f = 1$  for an initial density distribution, produces a maximum temperature of 3% lower, compared with using a uniform initial density distribution.

**Table 5.15:** Architectures for different initial density distributions for a three-dimensional domain with a partial Dirichlet boundary.

Initial density distribution is uniform with $p_f = 1$ (constant)			
$k^*$ [-]	50	100	500
$\tau$ [-]	$1.443 \times 10^{-1}$	$7.747 \times 10^{-2}$	$1.729 \times 10^{-2}$
Density distribution			
Initial density distribution is above solution with $p_f = 3$ (increasing)			
$k^*$ [-]	50	100	500
$\tau$ [-]	$1.669 \times 10^{-1}$	$9.196 \times 10^{-2}$	$2.188 \times 10^{-2}$
Density distribution			
Initial density distribution is uniform with $p_f = 3$ (increasing)			
$k^*$ [-]	50	100	500
$\tau$ [-]	$1.705 \times 10^{-1}$	$9.492 \times 10^{-2}$	$2.268 \times 10^{-2}$
Density distribution			

The method for the initial density distribution was not used in this study since the gain in thermal performance is minimal, the architecture does not change and it doubles the computational time.

### 5.5.7 Effect of Cold Spot Size




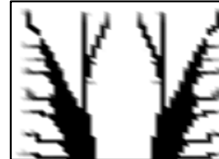
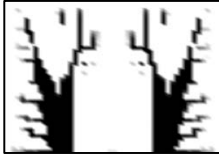

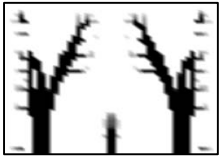



In this chapter, for all the optimisation runs, an arbitrary value for the width of the cold spot was chosen as  $c/L_D = 0.1$ , thus 10% of the width of the bottom boundary. As a reference, Gersborg-Hansen *et al.* used a width of 0.78125%, in their two-dimensional investigation [8]. The purpose of this subsection is to find out what effect the size of the cold spot has on  $\tau$  and the final architecture.

Table 5.16 shows converged architectures for a range of values for  $c/L_D$ . As the size of  $c/L_D$  is increased,  $\tau$  decreases up to  $c/L_D = 0.6$ . The reason for the decrease in  $\tau$  is that the bigger cold spot can now extract more heat from the domain. Up to  $c/L_D = 0.2$ , the architecture maintains the V shape with secondary branches extruding from the main branches with an upward angle. After  $c/L_D = 0.2$ ,

the V shape architecture breaks down and splits into two separate main branches. As  $c/L_D$  increases, the distance between the main branches increases and the width of the main branches decreases.

For  $c/L_D = 1.0$ , the whole bottom boundary is now a cold spot. If this boundary is applied, there is no variation in the sensitivities in an  $x$ - $y$ -plane. For this reason, the architecture shown in the table for  $c/L_D = 1.0$  is obtained. All the densities in an  $x$ - $y$ -plane are equal and the density decreases as  $z$  increases.

**Table 5.16:** Converged architectures showing the effect of the base width  $c/L_D$  for a three-dimensional domain with a partial Dirichlet boundary.

$c/L_D$ [-]	0.1	0.2	0.3	0.4
$\tau$ [-]	$2.294 \times 10^{-2}$	$1.861 \times 10^{-2}$	$1.731 \times 10^{-2}$	$1.651 \times 10^{-2}$
Density distribution				
$c/L_D$ [-]	0.5	0.6	0.7	0.8
$\tau$ [-]	$1.622 \times 10^{-2}$	$1.478 \times 10^{-2}$	$1.519 \times 10^{-2}$	$1.555 \times 10^{-2}$
Density distribution				
$c/L_D$ [-]	0.9	1.0		
$\tau$ [-]	$1.792 \times 10^{-2}$	$8.764 \times 10^{-2}$		
Density distribution				

## 5.6 Results for Different Conductivity Ratios and Volume Constraints

The purpose of the optimisation runs in Section 5.5 was to find the most suitable conditions for the optimisation in the three-dimensional domain with a partial Dirichlet boundary. Two most important aspects that need to be studied is the volume constraint and the conductivity ratio, since these are the factors that can be adjusted in a practical situation. The conductivity ratio was ranged between 5 and 3 000 and the volume constraint was ranged between 0.05 and 0.3. For this section,  $c/L_D$  is selected to be 0.1. The input parameters are shown in Table 5.17.

**Table 5.17:** Input parameters for conductivity and volume simulations for a three-dimensional partial Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$l$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.05 – 0.3 (5% - 30%)
$k_H$	10 – 6000 W/(mK)	$c/L_D$	0.1
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M_x M_x M$	50x50x50		

Table 5.18 shows results obtained for different conductivity ratios for the same volume constraint of 0.1. As the conductivity ratio is increased, the maximum temperature drops, which is expected since a higher-conductive material can transport the heat generated more effectively to the cold boundary. The architecture also shows that as the conductivity ratio is increased, the size of the main and side branches decreases in terms of width. The base of the tree also decreases in size with an increase in the conductivity ratio. The architecture for  $k^* = 3\,000$  does look as if it is discontinuous, but this is not true. The problem is that some of the side branches are in a plane that is not visible from the diagonal slice. The architecture are similar in nature to architectures obtained in two-dimensional studies [8–10], especially for  $k^*$  values greater than 500.

**Table 5.18:** Comparison of the conductivity ratios for  $V^* = 0.1$  and  $s_0 = 0.15$  for a three-dimensional domain with a partial Dirichlet boundary.







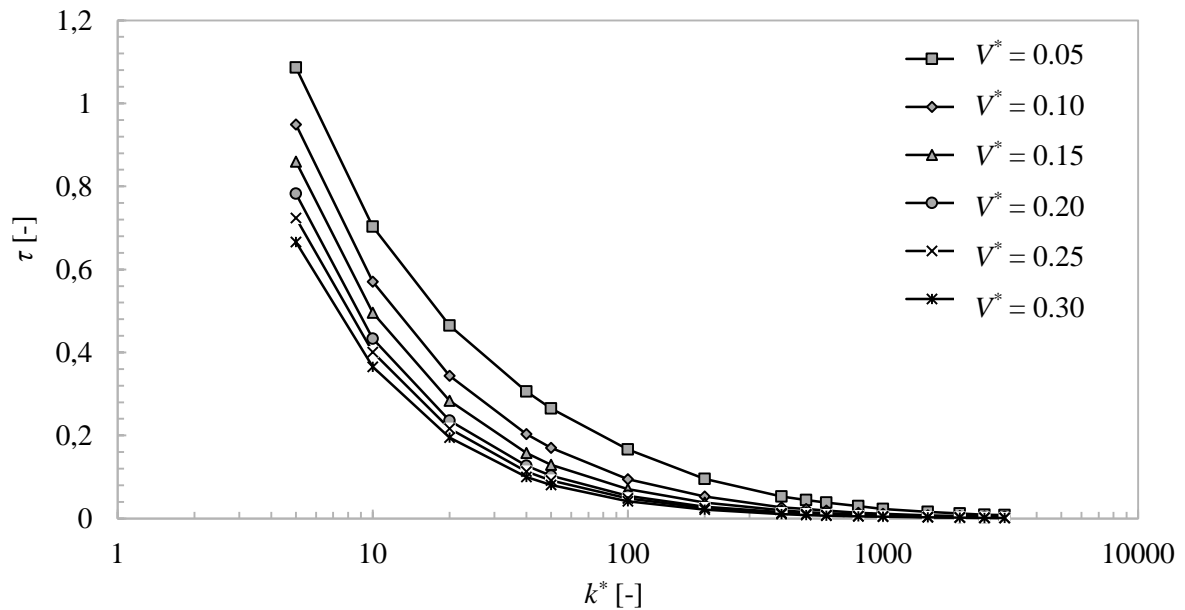
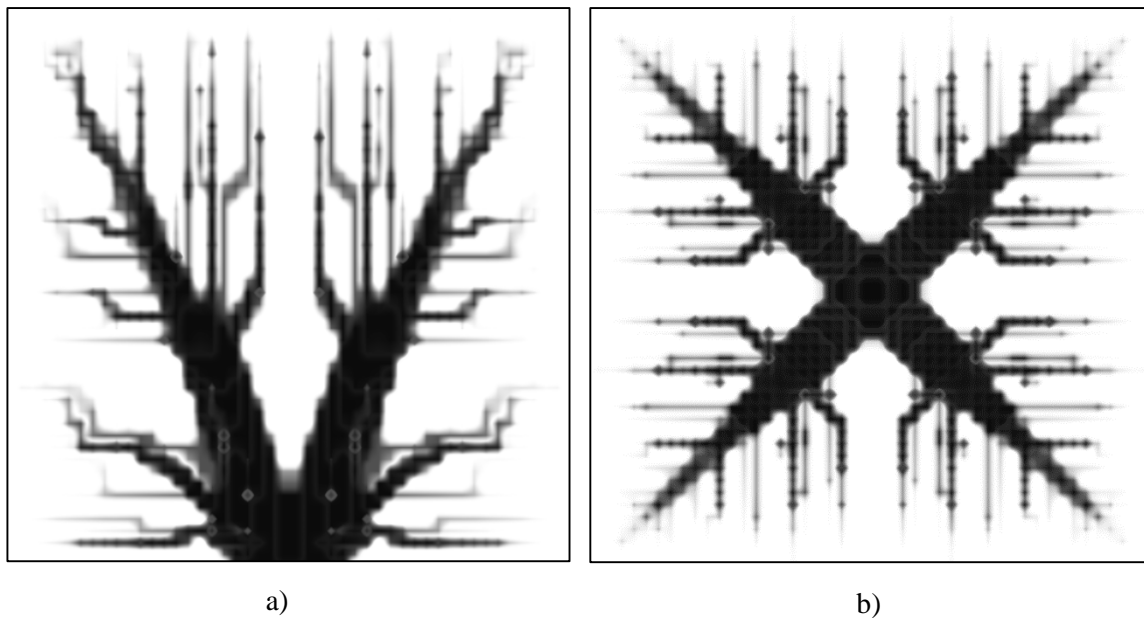
$k^* [-]$	5	50	500
$\tau [-]$	$9.487 \times 10^{-1}$	$1.704 \times 10^{-1}$	$2.268 \times 10^{-2}$
Density distribution			
$k^* [-]$	1000	2000	3000
$\tau [-]$	$1.145 \times 10^{-2}$	$6.312 \times 10^{-3}$	$4.090 \times 10^{-3}$
Density distribution			

Figure 5.15 shows a complete comparison of all volume constraints from 0.05 to 0.3 for all conductivity ratios, for a  $s_0$  value of 0.15. For other  $s_0$  values, the same pattern is obtained, but are not shown here. As seen in the graph, there is a drop in the maximum temperature for all values of the conductivity ratio, if the volume constraint is increased. From  $V^* = 0.05$  to 0.1, there is a substantial gain in maximum temperature, but this gain decreases as the volume constraint is increased. The volume constraint also has a greater effect on the low conductivity range, but a much smaller effect on the high conductivity range. The rest of the results can be found in Appendix B.1.1.

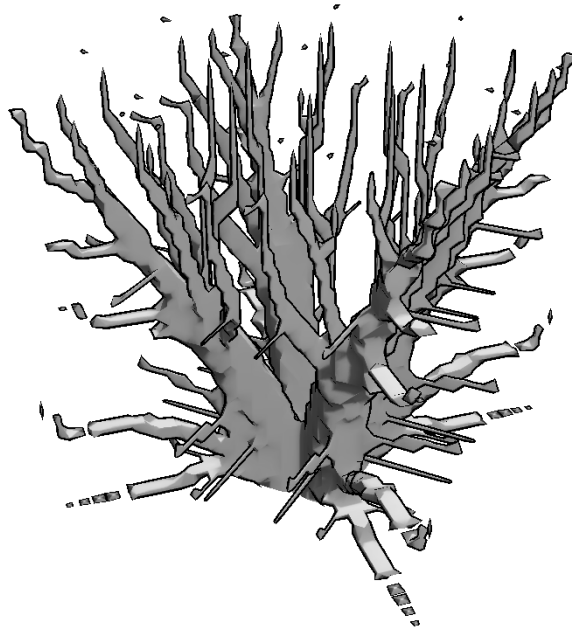


**Figure 5.15:** Comparison of all volume constraints to all conductivity ratios with  $s_0 = 0.15$  for a three-dimensional domain with a partial Dirichlet boundary.

Figure 5.16 shows the architecture for a side and top view for  $k^* = 2\,000$  and  $V^* = 0.05$ . The figures better illustrate the thinner main branches when the conductivity ratio is increased. From a top and side perspective, the main branches again extend to the corners of the domain. As seen in the top view, the secondary branches grow at a normal angle to the sides of the domain but again at an upward angle as seen from the side view. Figure 5.17 shows an isometric view of the architecture.



**Figure 5.16:** The architecture for  $k^* = 2\,000$  and  $V^* = 0.05$  a) side view b) top view.



**Figure 5.17:** Isometric view showing an iso-surface of the architecture obtained for  $k^* = 2000$  and  $V^* = 0.05$ .

Table 5.19 shows the architectures for different volume constraints at the same conductivity ratio of  $k^* = 500$ . The effect of the increase in the volume constraint is clearly seen with an increase in the size of the main and side branches of the trees. A volume above 0.15 is not very practical since space is normally the constraint in small electronics. All the volumes still show a V shape with the main branches extending to the corners of the domain.

**Table 5.19:** Architectures for the different volume constraints at  $k^* = 500$  and  $s_0 = 0.15$  for a three-dimensional domain with a partial Dirichlet boundary.

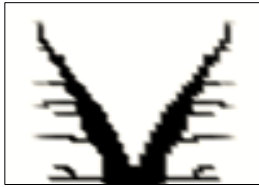





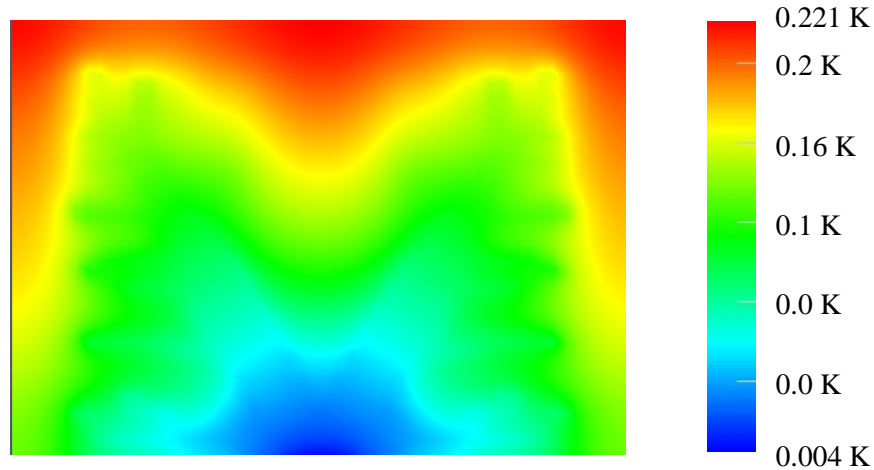
$V^* [-]$	0.05	0.1	0.15
$\tau [-]$	$4.420 \times 10^{-2}$	$2.268 \times 10^{-2}$	$1.538 \times 10^{-2}$
Density distribution			
$V^* [-]$	0.2	0.25	0.3
$\tau [-]$	$1.192 \times 10^{-2}$	$1.091 \times 10^{-2}$	$8.458 \times 10^{-3}$
Density distribution			

Figure 5.18 shows the temperature distribution for  $k^* = 500$  and  $V^* = 0.05$ . The architecture extracts the heat generated from the domain effectively. The cold spot can clearly be seen on the bottom boundary. In this figure, red indicates hot and blue indicates cold.





**Figure 5.18:** Temperature distribution for  $k^* = 500$  for a three-dimensional domain with a partial Dirichlet boundary.

## 5.7 Summary

The partial Dirichlet boundary was the first boundary condition tested for the three-dimensional domain. The effect of the asymptotes was again similar to that of the two-dimensional domain, with an optimal range for  $s_0$  between 0.15 – 0.25. The mesh-dependence study showed that fewer elements for the characteristic length are needed for convergence, but the iteration study showed that more iterations are needed compared with the two-dimensional cases. The study of the penalisation schemes showed that constant penalisation is not effective and  $p_f = 3$  with increasing penalisation produces a good architecture and maximum temperature. This indicated that the MMA algorithm is to some extent history dependent and does not necessarily reach the global minimum of  $g_0$ , but rather local minimums. Using a random initial guess for the density distribution is also detrimental to the final architecture and the maximum temperature. Using a solution from  $p_f = 1$  with constant penalisation decreased the maximum temperature by 3% compared with the case where a uniform initial material distribution is used. The architecture was unchanged and the additional computational time made this initial guess unfeasible.

After the optimisation runs, it can be concluded that increasing the conductivity ratio decreases the maximum temperature. An increase in the volume constraint also produced a lower maximum temperature, but this performance increase lowers as the volume constraint is increased. The architectures obtained are similar to the architectures obtained in two dimensions, with the main branches extending to the corners of the domain. It can be concluded that topology optimisation can be used to find optimal architectures for heat conduction in a three-dimensional domain using a partial Dirichlet boundary.

In the next chapter, the full Dirichlet boundary will be investigated. The same methodology used in this current chapter will be used to investigate this new boundary condition. For this boundary condition, an initial base is needed to give a variation in the  $x$ - $y$ -plane of sensitivities otherwise the MMA will not be able to generate a tree structure. This size and placement of this structure will be investigated in detail. In the next chapter, the penalisation factor and initial guess will not be investigated again. The values obtained in this chapter will be used.

## Chapter 6      **Three-Dimensional Methodology and Results for the Full Dirichlet Boundary**

### 6.1 Introduction

The boundary condition used in Section 5.2 is not practical, because with a heat sink utilisation, the whole bottom of the boundary will probably be at a specific temperature, also known as a full Dirichlet boundary. This will result in a lower more realistic maximum temperature. However, if a full Dirichlet boundary condition is considered with the code used in prior sections, it will give sensitivities that are identical for all the elements in each  $x$ - $y$ -plane. From this, the MMA algorithm will not produce a tree-like structure and will only lead to a uniform decreasing density distribution from the bottom up as shown in Figure 6.1. This problem can be avoided by giving the domain an initial base definition for the tree seed. This is discussed in the next section.



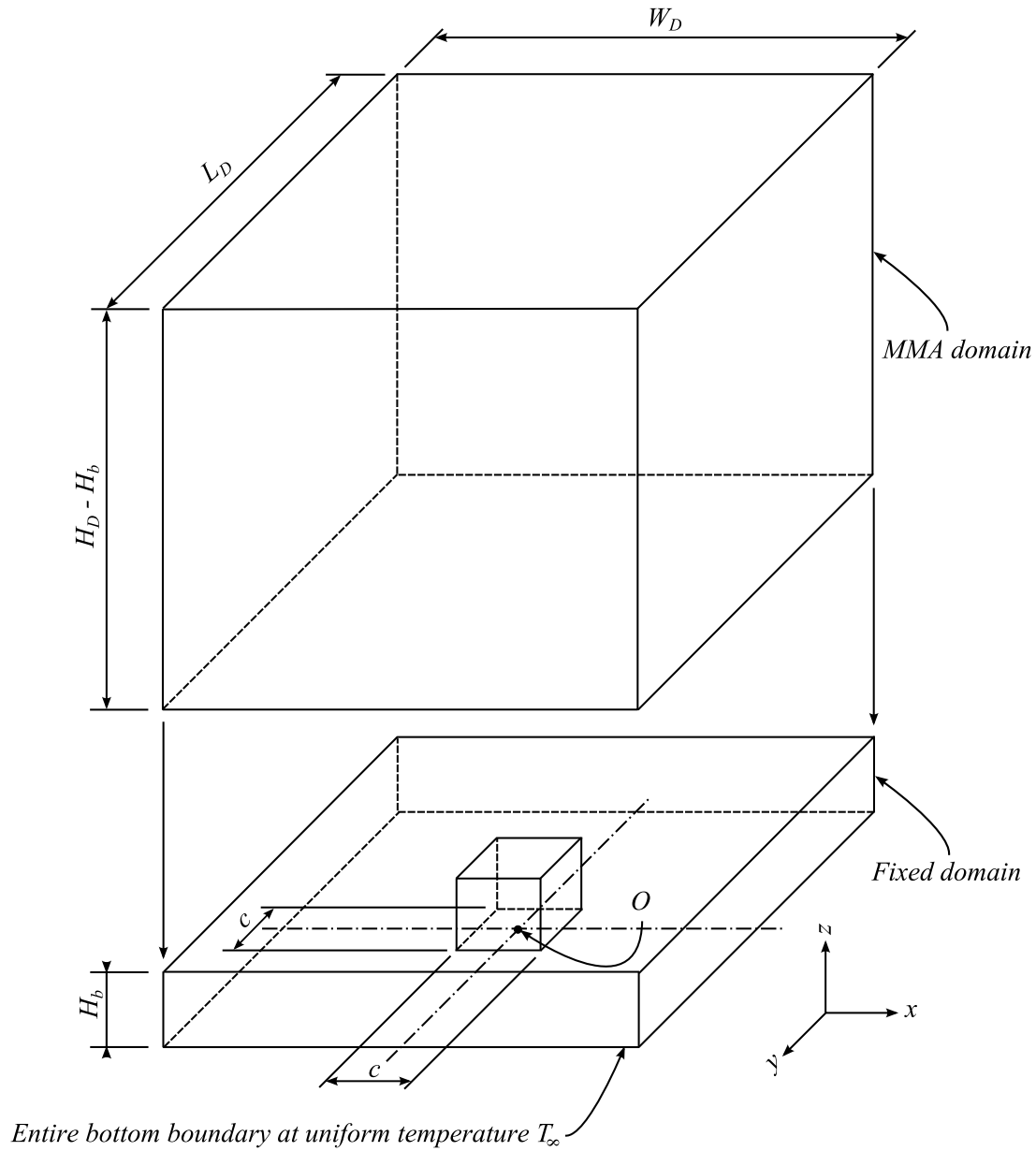
**Figure 6.1:** A converged architecture for a full Dirichlet boundary.

This chapter will first consider the size of the initial base seed placed in the centre of the bottom base using the parameters obtained in the previous chapter (element count, iteration count, etc.). Once this is established, the mesh-dependence study and iteration study will be checked again. For this chapter, penalisation will not be checked again, instead  $p_f = 3$  with the increasing scheme will be used. With this established, the effect of the placement of the initial base will be checked. Once the optimal placement has been found, the optimisation runs for different conductivity and volume constraints will be performed.

### 6.2 Domain

The only way to achieve a tree-like structure is when there is variation in the sensitivities in the  $x$ - and  $y$ -directions as with the previous chapters. The problem with the identical sensitivities can be solved

by supplying the domain with an initial base of a tree structure, similar to the bases obtained in Section 5.6. As shown in Figure 6.2, the elements in the initial base structure will have a density of 1 while the surrounding elements will have a density of 0. Thus all elements up to the height of the initial base will be fixed and will not be influenced by the optimisation routine. The MMA will thus only loop over the elements above the fixed defined subdomain. This defined subdomain will ensure that there is variation in the sensitivities, allowing a tree-like structure to form.



**Figure 6.2:** Domain for the full Dirichlet boundary for the three-dimensional case.

## 6.3 Methodology

### 6.3.1 Dimensionless Maximum Temperature

The dimensionless maximum temperature was again evaluated for the full Dirichlet boundary, due to the different nature of the boundary and the fixed base. It was found that the definition of the dimensionless temperature did not have to be updated. It is repeated here for convenience.

$$\tau = \frac{(T_{max} - T_{\infty})k_L}{q_H L_D^2} \quad (6.1)$$

### 6.3.2 The Effect of the Fixed Subdomain Height

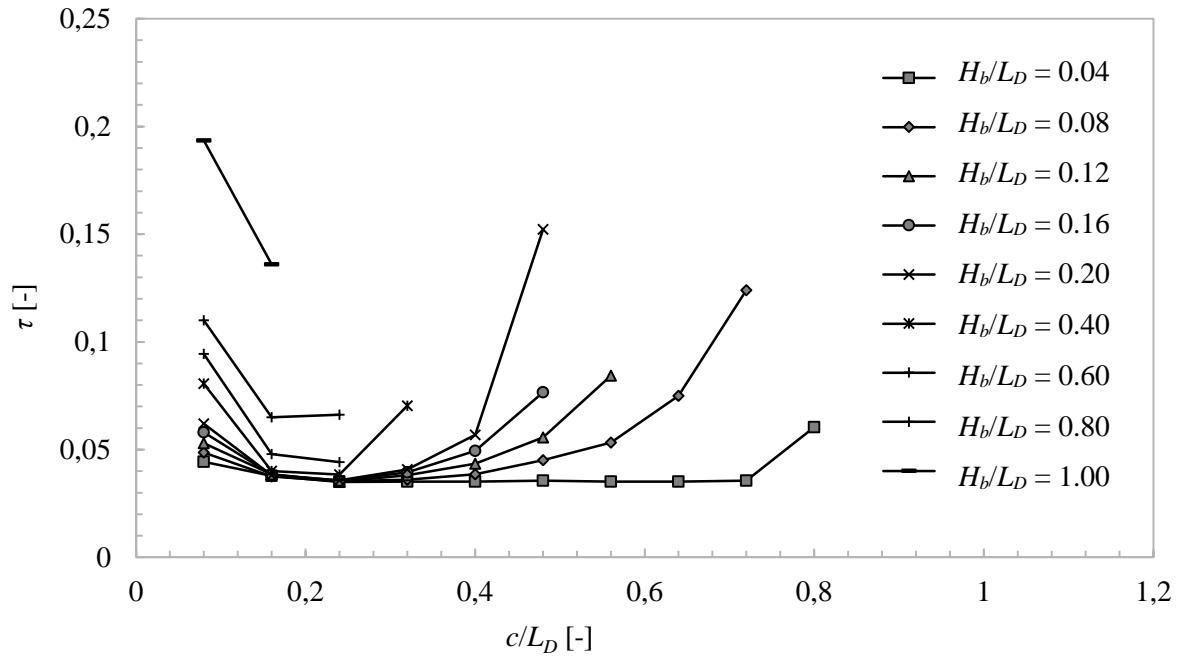
It is important to determine what impact the choice of  $c$  and  $H_B$  has on the converged architecture, and whether there is an optimal choice for these values. To test this, various optimisation runs were conducted for varying sizes of the base width and base height. Initially, it was assumed that optimal placement of the initial base was in the middle of the bottom domain. This was later confirmed in Section 6.3.6. The input parameters are given in Table 6.1.

It was found that the *SQSD*, used in the previous simulations to find the root, did not work quite as well in these simulations. It did not find the roots of the functions of the MMA subproblem accurately. The bisection method was then tried, which also uses the derivative of the objective function. The bisection method gave the correct root values.

**Table 6.1:** Input parameters for the effect of the base size for a three-dimensional full Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.05 (5%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.08 – 0.8
$q_H$	10.0 W/m <sup>3</sup>	$T_{\infty}$	0 K
$M \times M \times M$	50x50x50	$H_b/L_D$	0.04 – 1.0

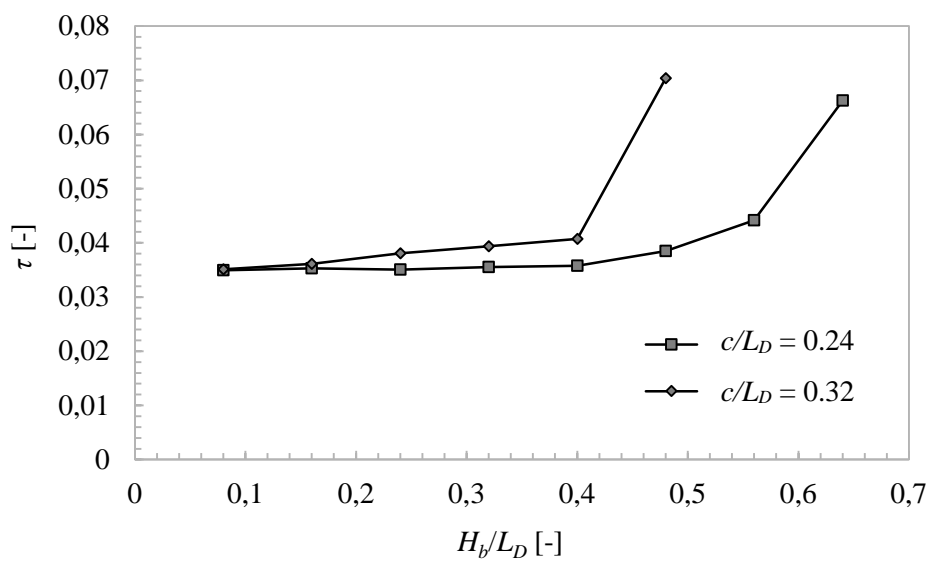
As seen in Figure 6.3,  $c$  and  $H_b$  have a significant influence on the converged  $\tau$ . There is an optimal width for the base of the tree, around  $c/L_D = 0.24$  for  $H_b/L_D < 0.3$ . As  $c/L_D$  approaches 0 and 1,  $\tau$  increases. The effect of  $H_b$  is more apparent in the lower and higher values of  $c$ . When  $c/L_D = 0.24$ , the effect is less, but a smaller height does give a better maximum temperature as shown in Figure 6.4. Only those combinations of  $c$  and  $H_b$  that did not violate the volume constraint, and which allowed sufficient freedom for the MMA algorithm, were considered.



**Figure 6.3:** The effect of the base width and height on the maximum temperature for  $k^* = 500$  for a three-dimensional domain with a full Dirichlet boundary.

When  $c/L_D$  is too small or too large (0.04 and 0.8), there is again very little variation in the  $x$ - $y$ -plane sensitivities, which makes it difficult for the MMA to generate a tree structure.

Figure 6.4 shows the results for  $c/L_D = 0.24$  and  $c/L_D = 0.32$  for all heights of the base. As the base height is increased, the maximum temperature increases. From this, it can be concluded that when the MMA algorithm is given more freedom (i.e. a smaller base height), lower converged maximum temperatures are obtained. In terms of the volume, the volume constraint is satisfied for all accounts of  $c$  and  $H_b$ .



**Figure 6.4:** Influence of  $H_b/L_D$  for  $c/L_D = 0.24$  and  $0.32$ .

**Table 6.2:** Comparison of architectures for different heights using  $k^* = 500$ ,  $c/L_D = 0.24$  and  $V^* = 0.05$  for a three-dimensional full Dirichlet boundary.


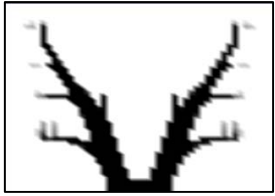
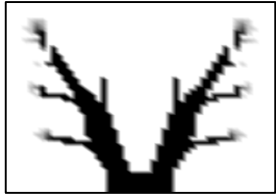


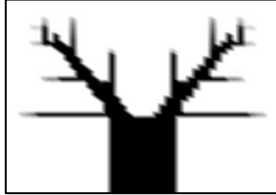
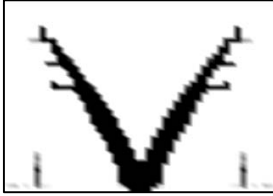
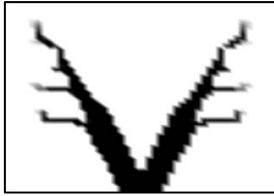


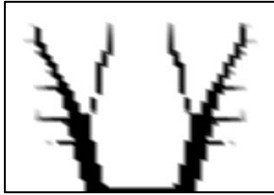
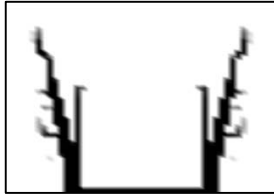



$H_b/L_D$ [-]	0.04	0.08	0.12
$\tau$ [-]	$3.495 \times 10^{-2}$	$3.530 \times 10^{-2}$	$3.506 \times 10^{-2}$
Density distribution			
$H_b/L_D$ [-]	0.16	0.20	0.40
$\tau$ [-]	$3.552 \times 10^{-2}$	$3.575 \times 10^{-2}$	$3.847 \times 10^{-2}$
Density distribution			

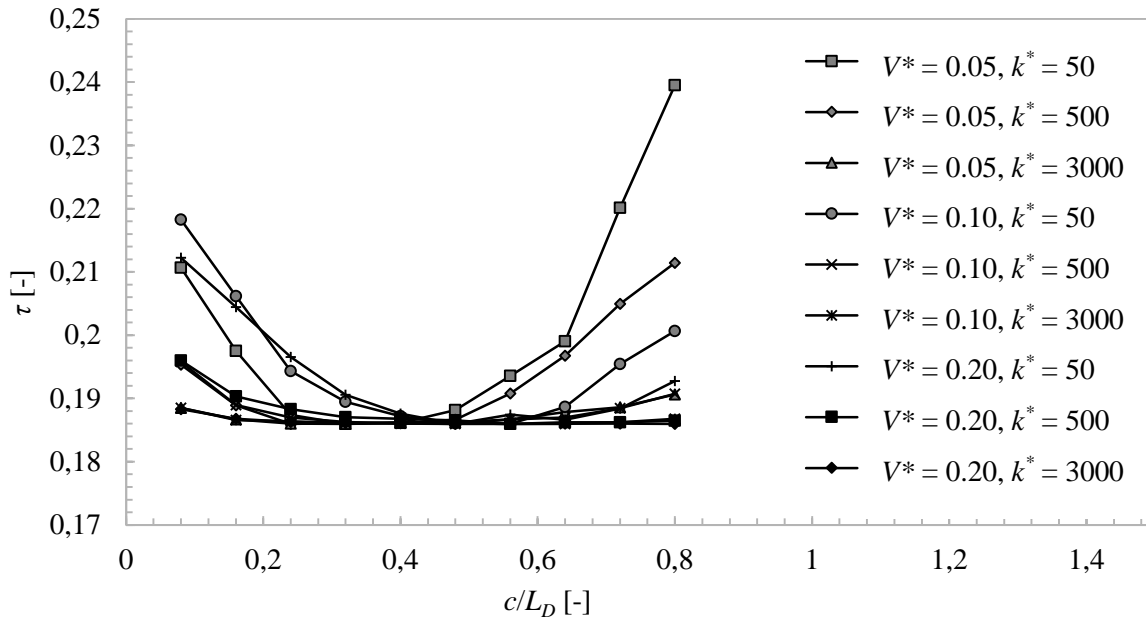
Table 6.2 shows the different converged architectures for a subdomain with  $c/L_D = 0.24$  at different heights of the subdomain. All architectures are similar, again with the V shape for the main branches extending to the corners of the domain. When the height of the subdomain,  $H_b/L_D$ , is above 0.2, there is a huge restriction in the domain, which does not give the MMA much to work with. As the height of the base is increased, the width of the main branches and the number of secondary branches decrease. This is due to the base consuming more of the volume ratio, allowing less volume to be allocated by the MMA.

Table 6.3 shows converged architectures for a range of  $c/L_D$ , for a base height of  $H_b/L_D = 0.04$ . Up to  $c/L_D = 0.56$ , the main branches again form a V shape. As the width is increased further, the main branches start to grow vertically to the corners of the domain. As the width of the base is increased, the width of the main branches and the number of secondary branches again decrease. Above  $c/L_D = 0.56$ , the architecture is less defined in terms of secondary branches. This is due to a lack of variation in the sensitivities in the  $x$ - $y$ -plane.

**Table 6.3:** Comparison of architectures for different values of  $c/L_D$  using  $k^* = 500$ ,  $H_b/L_D = 0.04$  and  $V^* = 0.05$  for a three-dimensional full Dirichlet boundary.

$c/L_D$ [-]	0.08	0.16	0.24
$\tau$ [-]	$4.431 \times 10^{-2}$	$3.787 \times 10^{-2}$	$3.495 \times 10^{-2}$
Density distribution			
$c/L_D$ [-]	0.32	0.40	0.56
$\tau$ [-]	$3.514 \times 10^{-2}$	$3.516 \times 10^{-2}$	$3.514 \times 10^{-2}$
Density distribution			
$c/L_D$ [-]	0.64	0.72	0.80
$\tau$ [-]	$3.516 \times 10^{-2}$	$3.564 \times 10^{-2}$	$6.041 \times 10^{-2}$
Density distribution			

The optimal value for  $c/L_D$  and  $H_b/L_D$  obtained earlier in this subsection is based on one conductivity and volume constraint.  $H_b/L_D$  of 0.04 is clearly superior, but there is variation in  $\tau$  for different values of  $c/L_D$ . At  $H_b/L_D = 0.04$ , different values of  $c/L_D$  were tested for a range of volume and conductivity ratios. The results are shown in Figure 6.5. The results are normalised around the optimum of each volume and conductivity set. The figure shows that there is a global optimum around  $c/L_D = 0.4$ . The chosen value for  $c/L_D$  of 0.24 also performs well for all cases, except for  $k^* = 50$ .

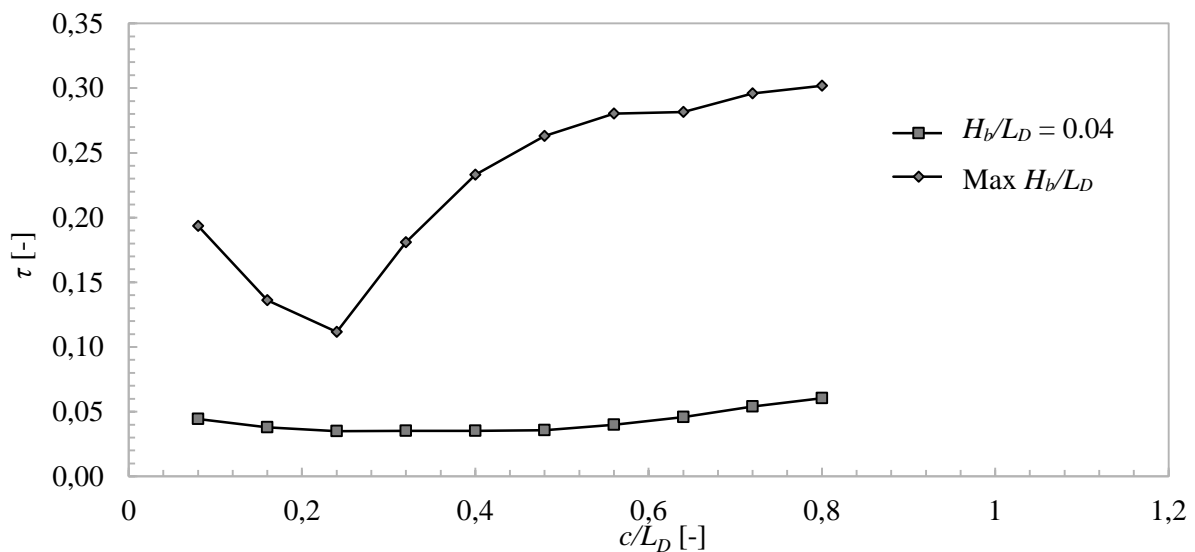


**Figure 6.5:** The effect on  $\tau$  for different  $k^*$  and  $V^*$  ratios for  $H_b/L_D = 0.04$  and different values of the seed width,  $c/L_D$ .

Another aspect that was investigated is the comparison of converged architectures as obtained with the MMA algorithm with solid bases (which still satisfy the volume constraint). This means that there is a solid base with width  $c$  and height  $H_b$ , where  $H_b$  is the maximum allowed height for the specific width, which satisfies the volume constraint. Thus the following equation is satisfied:

$$\frac{c^2 H_b}{L_D^3} = V^* \quad (6.2)$$

Figure 6.6 shows the comparison. Giving the domain a small base and letting the MMA solve the density distribution is clearly more effective than solid blocks. From a manufacturing and electronic design perspective, such simplistic block-cooling structures would be superior, but thermally inferior.



**Figure 6.6:** Comparison of converged architectures with  $H_b/L_D = 0.04$  and solid blocks with maximum heights.



Table 6.4 shows some converged architectures for the results from Figure 6.6. In terms of maximum temperature, allowing the MMA to optimise the domain is clearly superior compared with a solid block.

**Table 6.4:** Architectures for MMA optimised architectures and solid blocks

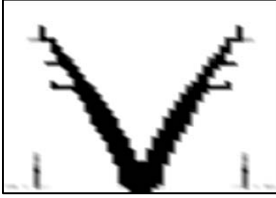
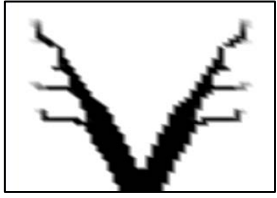
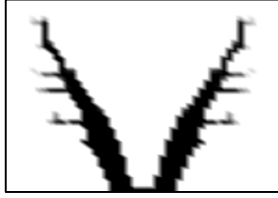

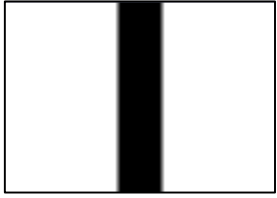

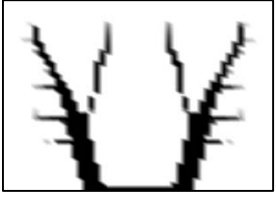
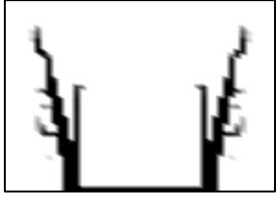

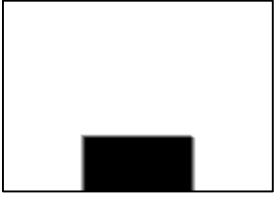


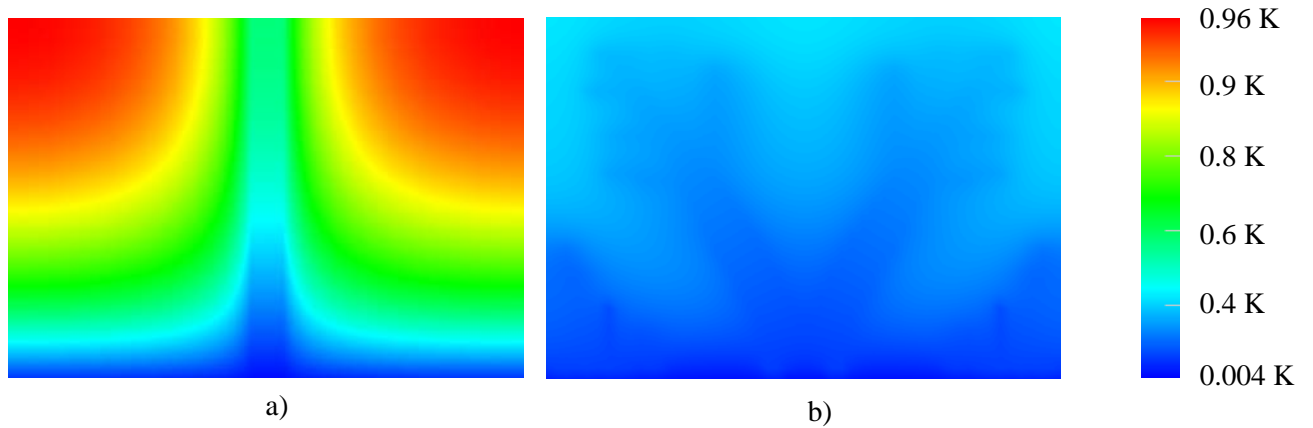
$c/L_D$ [-]	0.08	0.16	0.24
$H_b/L_D$ [-]	0.04	0.04	0.04
$\tau$ [-]	$4.432 \times 10^{-2}$	$3.787 \times 10^{-2}$	$3.495 \times 10^{-2}$
Density distribution			
$c/L_D$ [-]	0.08	0.16	0.24
$H_b/L_D$ [-]	1.0	1.0	0.86
$\tau$ [-]	$1.935 \times 10^{-1}$	$1.362 \times 10^{-1}$	$1.116 \times 10^{-1}$
Density distribution			
$c/L_D$ [-]	0.4	0.56	0.80
$H_b/L_D$ [-]	0.04	0.04	0.04
$\tau$ [-]	$3.517 \times 10^{-2}$	$3.564 \times 10^{-2}$	$3.978 \times 10^{-2}$
Density distribution			
$c/L_D$ [-]	0.4	0.56	0.80
$H_b/L_D$ [-]	0.3	0.14	0.06
$\tau$ [-]	$2.331 \times 10^{-1}$	$2.630 \times 10^{-1}$	$2.803 \times 10^{-1}$
Density distribution			

Figure 6.7 shows the temperature distribution for two architectures, a solid block on the left and a converged MMA architecture on the right. As seen in the figure, the MMA architecture distributes the heat more evenly over the domain. The MMA converged architecture also provides a maximum temperature of 4.5 times lower than the temperature of the solid block. This shows the effectiveness of using the topology optimisation. For both figures, red is hot and blue is cold and both use the same scale.



**Figure 6.7:** Temperature distribution for a) the solid block ( $c/L_D = 0.08$ ,  $H_b/L_D = 1.0$ ) and b) the MMA converged architecture ( $c/L_D = 0.08$ ,  $H_b/L_D = 0.04$ ).

### 6.3.3 Effect of the Asymptote Parameters, $s$ and $s_0$

As was done with the partial Dirichlet boundary condition, the effect of the MMA  $s$  and  $s_0$  values was checked. This is done to verify that the previously selected values are still correct. A small element count (20x20x20) was used to test most of the  $s$  and  $s_0$  values. The results are similar to the previous boundary condition. For the maximum temperature,  $s = 0.85$  still gives the most stable results over all the values of  $s_0$ . For all values of  $s$ , there is still an optimum point for  $s_0$  in the range of 0.15 - 0.25. The input parameters are shown in Table 6.5.

**Table 6.5:** The input parameters for the effect of the asymptotes.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.05 (5%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.24
$q_H$	10.0 W/m <sup>3</sup>	$T_\infty$	0 K
$M \times M \times M$	20x20x20	$H_b/L_D$	0.04

The results are found in Appendix B.2.1.

### 6.3.4 Mesh-Dependence Study

For the partial Dirichlet boundary condition in the previous chapter, an element count of 50x50x50 was sufficient for the maximum temperature to converge. The mesh-dependence was checked for the full Dirichlet boundary condition. For  $M > 50$  elements,  $\tau$  does not change significantly. The results can be found in Appendix B.2.2. There is a slight reduction when  $M$  is increased to 100 elements, but

this is not enough to justify the additional simulation time needed. For  $s_0 = 0.15$ , there is a difference of 5.12% in  $\tau$  when comparing the 50x50x50 and 100x100x100 element cases. The input parameters are shown in Table 6.6.

**Table 6.6:** Input parameters for the mesh dependence of a three-dimensional domain with a full Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$I$	60 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.05 (5%)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.24
$q_H$	10.0 W/m <sup>3</sup>	$H_b/L_D$	0.04
$M \times M \times M$	20x20x20 – 100x100x100	$T_\infty$	0 K

Table 6.7 shows the converged architectures for the mesh dependence of a full Dirichlet boundary. As seen in the table, after 50x50x50 elements, the main shape of the architecture is established. After this, only extra secondary branches start to appear and the grey areas diminish. There are also much less secondary branches compared with the architectures achieved using the partial Dirichlet boundary. The secondary branches also only appear near the end of the main branch and not at the base of the tree.

**Table 6.7:** Converged architectures as seen on the diagonal section plane  $D$  for the mesh dependence with  $s_0 = 0.15$ .









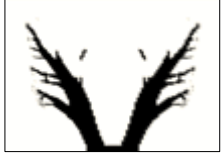
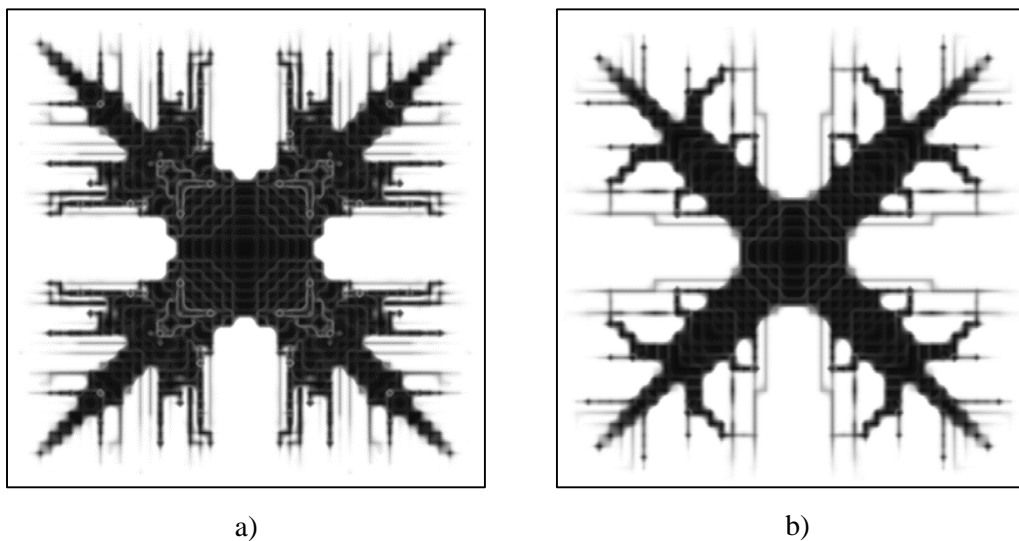
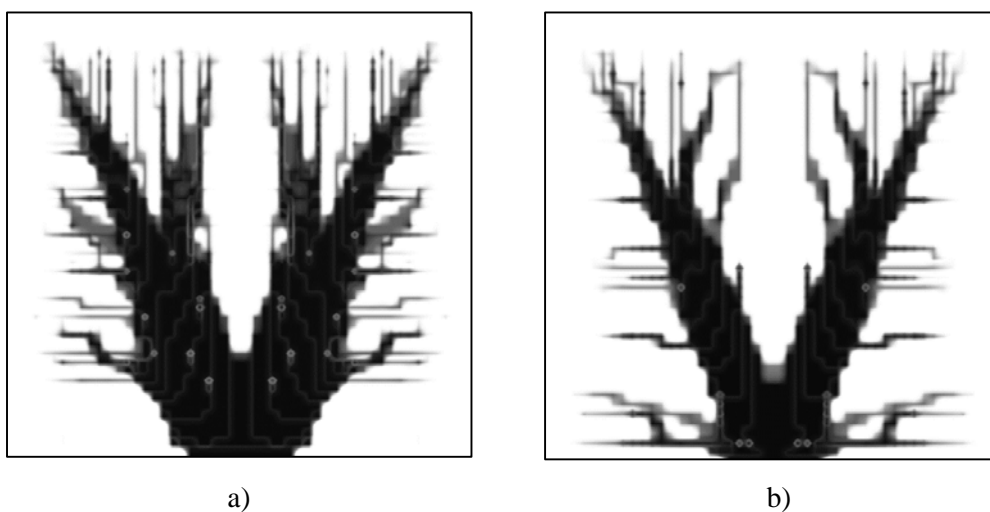
$M$ [-]	20	30	40	50
$\tau$ [-]	$4.697 \times 10^{-2}$	$4.151 \times 10^{-2}$	$4.080 \times 10^{-2}$	$3.930 \times 10^{-2}$
Density distribution				
$M$ [-]	60	70	80	90
$\tau$ [-]	$3.847 \times 10^{-2}$	$3.930 \times 10^{-2}$	$3.645 \times 10^{-2}$	$3.774 \times 10^{-2}$
Density distribution				
$M$ [-]	100			
$\tau$ [-]	$3.734 \times 10^{-2}$			
Density distribution				

Figure 6.8 shows a top view for converged architectures for both a partial and full Dirichlet boundary condition. The similarities are clearly seen here, with the main branches extending the corners of the domain. The main branches are thinner though for the full Dirichlet boundary compared with the partial boundary. The base in the middle is also smaller for the full Dirichlet boundary. In terms of the secondary branches, there are fewer branches for the full Dirichlet boundary compared with the partial boundary.

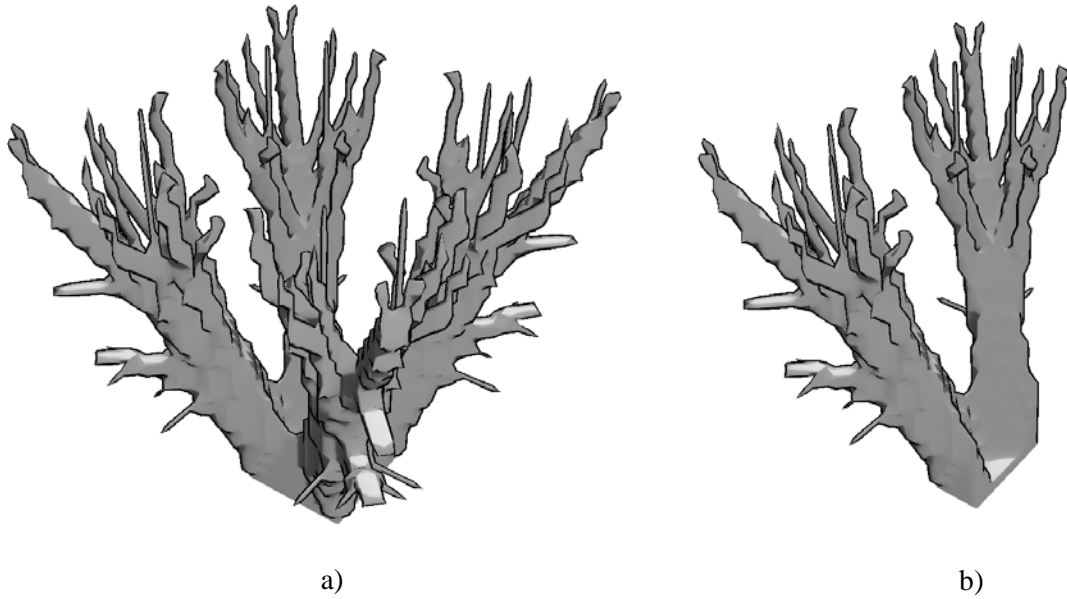
Figure 6.9 shows a side view for converged architectures for both a partial and full Dirichlet boundary. This view better illustrates the decreased width of the main branches of the full Dirichlet boundary. The V shape is still evident in both architectures. For the full Dirichlet boundary, there are fewer secondary branches, but secondary branches grow lower down the main branch compared with the partial boundary. Figure 6.10 shows an isometric view of the architecture obtained with the full Dirichlet boundary.



**Figure 6.8:** A top view of a converged architecture for a) a partial Dirichlet boundary and b) a full Dirichlet boundary ( $k^* = 500$ ,  $V^* = 0.05$ ).



**Figure 6.9:** A side view of a converged architecture for a) a partial Dirichlet boundary and b) a full Dirichlet boundary ( $k^* = 500$ ,  $V^* = 0.05$ ).



**Figure 6.10:** An isometric view showing the converged architecture for a full Dirichlet boundary a) a full domain and b) half the domain ( $k^* = 500$ ,  $V^* = 0.05$ ).

### 6.3.5 Iteration-Dependence Study

For the full Dirichlet boundary condition, the influence of the number of iterations was rechecked. This test was done for a node count of  $50 \times 50 \times 50$ , as described in the previous section. The input parameters are shown in Table 6.8.

**Table 6.8:** Input parameters for the iteration dependence of a three-dimensional full Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p$	3
$H_D$	1.0 m	$I$	10 – 100 iterations
$k_L$	2.0 W/(mK)	$V^*$	0.05 (5 %)
$k_H$	1000.0 W/(mK)	$c/L_D$	0.24
$q_H$	10.0 W/m <sup>3</sup>	$H_b/L_D$	0.04
$M \times M \times M$	$50 \times 50 \times 50$	$T_\infty$	0 K

It was found that for an iteration count above 60, little reduction in  $\tau$  was observed. This is similar to what was observed in the previous chapter. The difference in  $\tau$  between  $I = 60$  and 100 is 6.4%. This might be deemed significant, but at the higher computational cost, running at  $I = 100$  was not justifiable. The difference from 60 to 70 iterations is only 0.018%, which is acceptable. The results also started to oscillate after  $I = 60$ . Thus  $I = 60$  was chosen for the rest of the optimisation runs for the full Dirichlet boundary. The results are shown in Appendix B.2.3.

**Table 6.9:** Converged architectures as seen on the diagonal section plane  $D$  for the iteration-dependence.









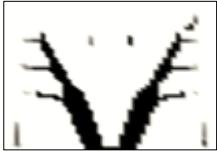

$I$ [-]	10	20	30	40
$\tau$ [-]	$7.464 \times 10^{-2}$	$5.555 \times 10^{-2}$	$4.359 \times 10^{-2}$	$3.930 \times 10^{-2}$
Density distribution				
$I$ [-]	50	60	70	80
$\tau$ [-]	$3.730 \times 10^{-2}$	$3.559 \times 10^{-2}$	$3.559 \times 10^{-2}$	$3.531 \times 10^{-2}$
Density distribution				
$I$ [-]	90	100		
$\tau$ [-]	$3.595 \times 10^{-2}$	$3.335 \times 10^{-2}$		
Density distribution				

Table 6.9 shows the architectures for the iteration study. The architectures follow the same trend as with the partial Dirichlet boundary. For  $I < 40$ , there are still noticeable grey areas in the final solution. After  $I = 40$ , the grey areas are minimal and there is no real difference in the main branches of the architectures.

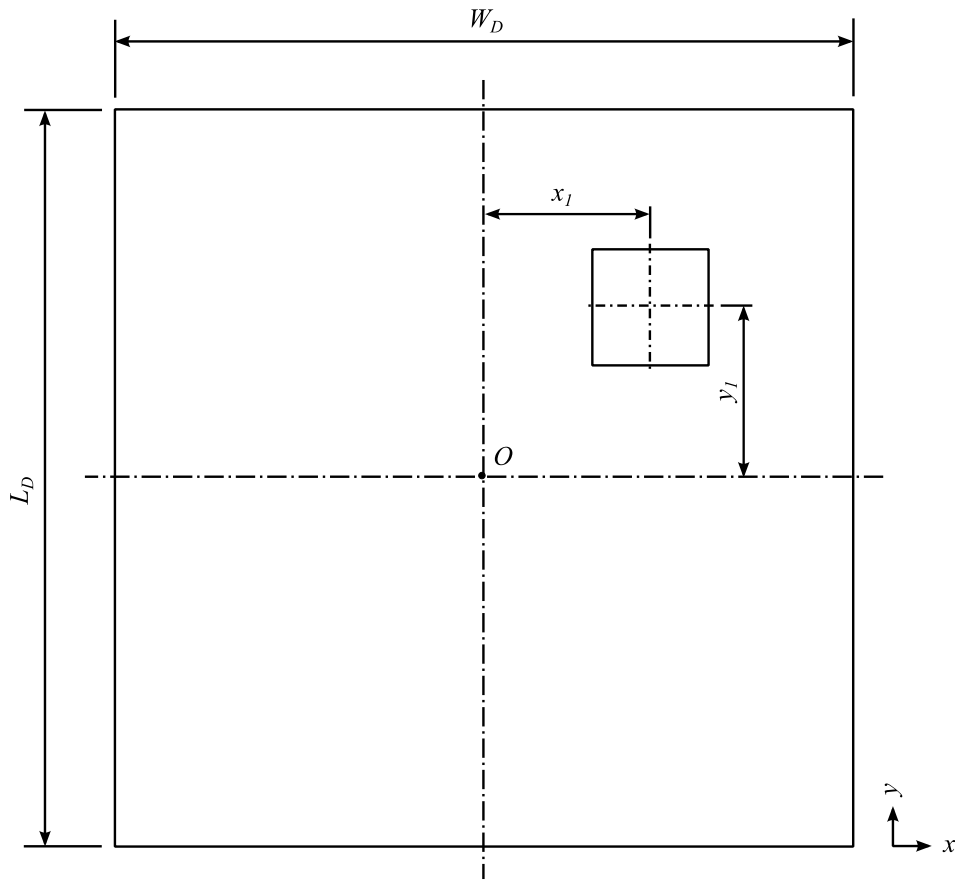
With the increase in  $I$ , there are minor differences in the secondary branches. Compared with the partial Dirichlet boundary, the secondary branches are now mostly horizontal. The number of secondary branches stays mostly unchanged after  $I = 50$ . There is also a small dissymmetry for  $I = 80$  and  $I = 90$ . This could be because of small numerical errors.

### 6.3.6 Placement of Base Structure

When there is only one base tree, it is probable that the optimal placement of the base would be in the middle of the lower boundary such that the domain is symmetrical. Despite this, it would be interesting to see what the effect is of moving the base. The width  $c$  and height  $H_b$  for the base found in Section 6.3.2 were used for this sub investigation. Since the thermal response would be symmetrical in terms of the boundary conditions and the dimensions, the base of the tree was only moved in one of the quadrants as shown in Figure 6.11, where  $x_1/L_D$  and  $y_1/L_D$  were varied each from 0.0 to 0.38. The input parameters are shown in Table 6.10.

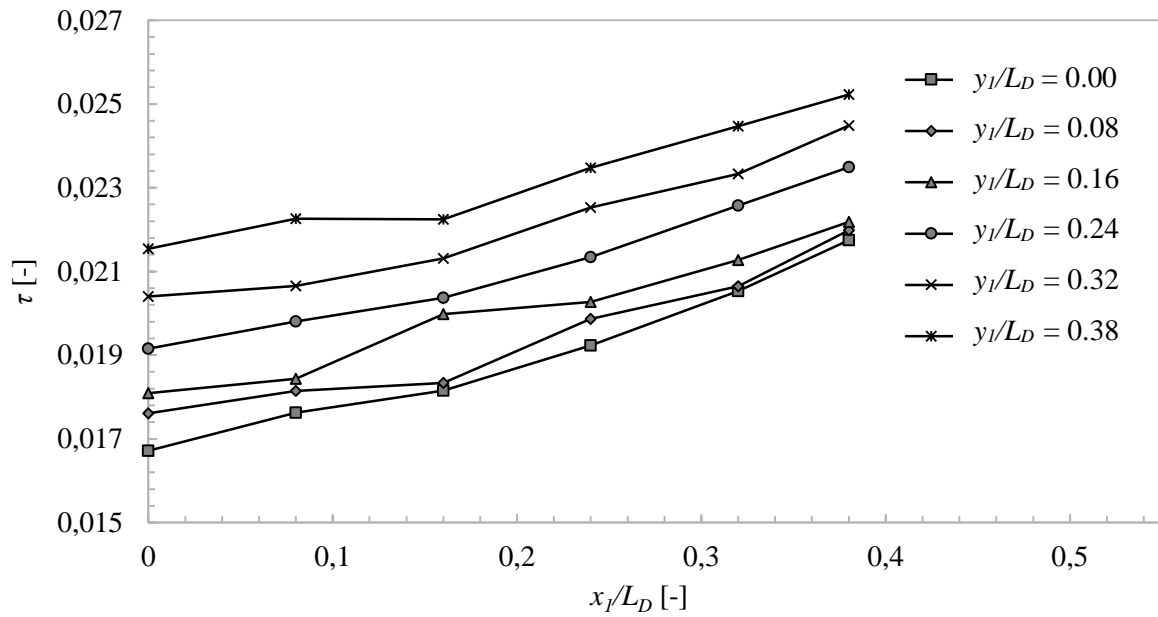
**Table 6.10:** Input parameters for the placement of the base structure for a three-dimensional domain with a full Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	Penalisation scheme	Increasing
$W_D$	1.0 m	$p_f$	3
$H_D$	1.0 m	$l$	60 iterations
$k_L$	2.0 W/(mK)	$c/L_D, H_b/L_D$	0.24, 0.04
$k_H$	1000.0 W/(mK)	$M \times M \times M$	50x50x50
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.0 – 0.38
$V^*$	0.05 (5 %)	$T_\infty$	0 K



**Figure 6.11:** The parameters used for moving the base of the tree.

Figure 6.12 shows the effect of the position of the initial base tree on the maximum temperature. It is clear that the optimal placement of the base tree is in the middle of the bottom boundary ( $x_1/L_D = 0$  and  $y_1/L_D = 0$ ), which makes sense due to the symmetric nature of the problem. Since the boundaries are symmetrical, the heat spreading will be symmetrical throughout the domain. Thus the cooling should be optimal if the maximum absolute distance from the base to any point on the edge of the domain is minimal. This would occur when the base is in the centre of the lower boundary of the domain.

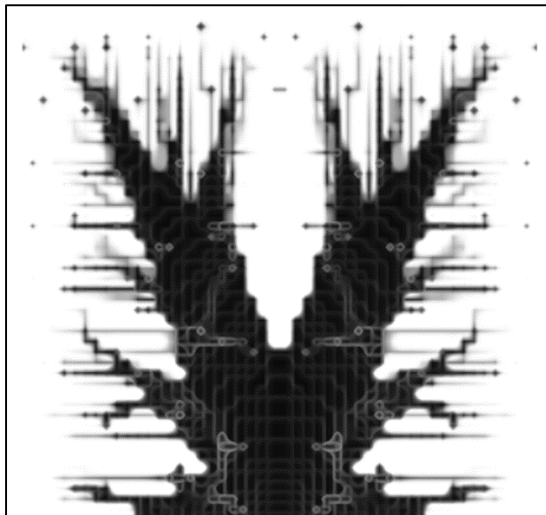


**Figure 6.12:** The effect on  $\tau$  due to the position of the initial base tree for a three-dimensional domain with a full Dirichlet boundary.

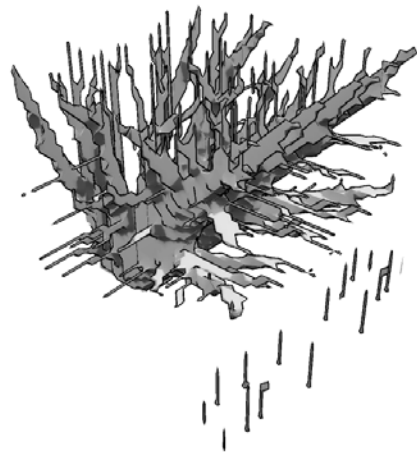
Moving the base horizontally to the edge of the domain, results in a 26% increase in the maximum temperature compared with the base in the middle of the bottom boundary. Moving the base to the corner of the bottom boundary results in the highest maximum temperature and a 40% increase in the maximum temperature, compared with the base in the middle of the bottom boundary.

Figure 6.13 shows different views for a converged architecture with the initial base on the edge of the bottom boundary ( $x_1/L_D = 0.38$ ,  $y_1/L_D = 0.0$ ). As seen in the top view, there are now only two main branches, extending to the opposite corners of the domain. These branches are also much thicker compared with the initial block being in the centre of the bottom boundary. These branches need to be thicker to extract the heat produced in the part of the domain where the initial base is not present.

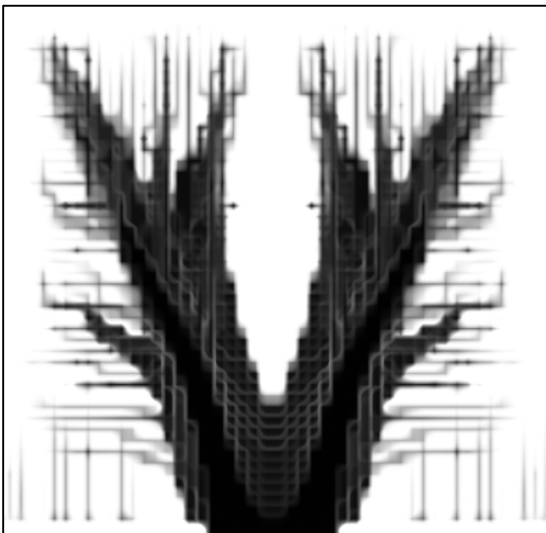




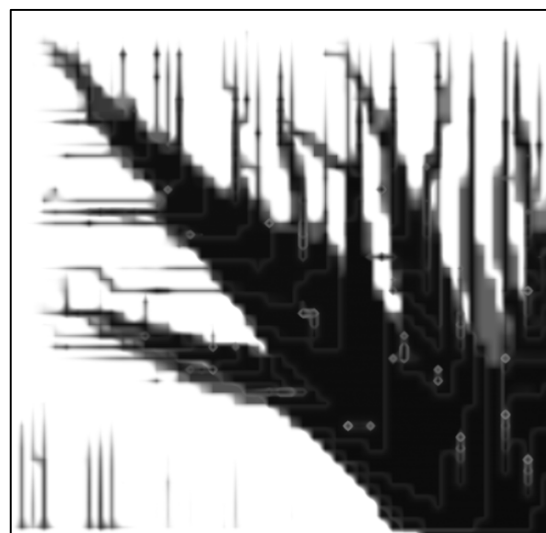
Top view



Isometric



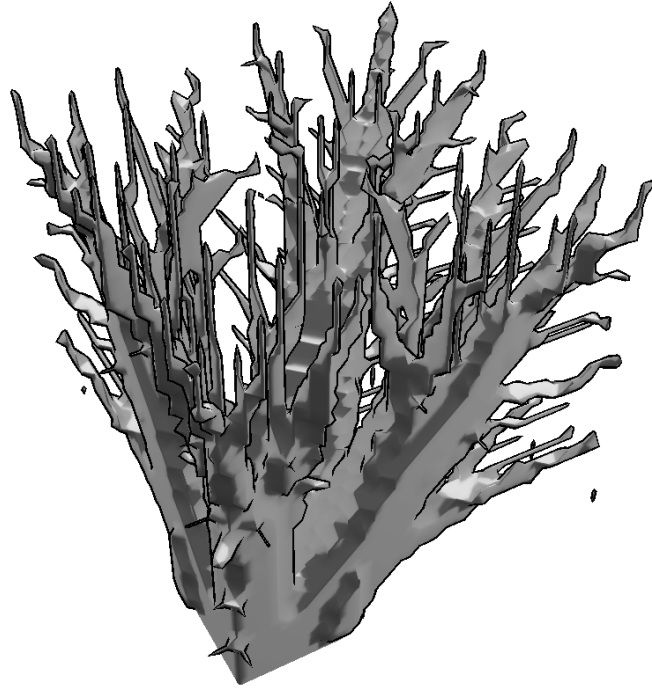
Front view



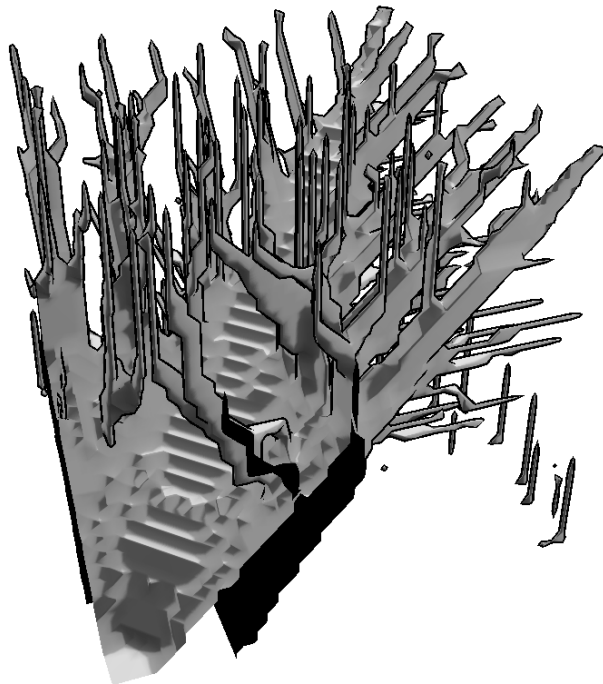
Side view

**Figure 6.13:** A converged architecture with the initial block on the edge of the bottom boundary ( $x_1/L_D = 0.38$ ,  $y_1/L_D = 0.0$ ).

Figure 6.14 shows an isometric view of a converged architecture with the initial base at  $x_1/L_D = 0.26$  and  $y_1/L_D = 0.26$ . Since the initial base is not on the edge of the bottom boundary, there are again now four main branches. The width of the branch correlates with the distance to the corner of the domain to which the branch grows.



**Figure 6.14:** Isometric view for the architecture obtained with the initial block just off-centre in one of the quarters of the bottom boundary ( $x_1/L_D = 0.26$ ,  $y_1/L_D = 0.26$ ).



**Figure 6.15:** Isometric view for the architecture obtained with the initial block in the corner of the bottom boundary ( $x_1/L_D = 0.38$ ,  $y_1/L_D = 0.38$ ).

Figure 6.15 shows an isometric view of a converged architecture with the initial base at  $x_1/L_D = 0.38$  and  $y_1/L_D = 0.38$ . With the initial block now in the corner of the domain, there is no branch extending

to that respective top corner. There is also only one clear main branch. This main branch is much thicker now, compared with the main branch in Figure 6.14. There are some secondary branches, which extend in a normal direction to the edges across from the corner where the initial base is placed.

## 6.4 Results for Different Conductivity Ratios and Volume Constraints

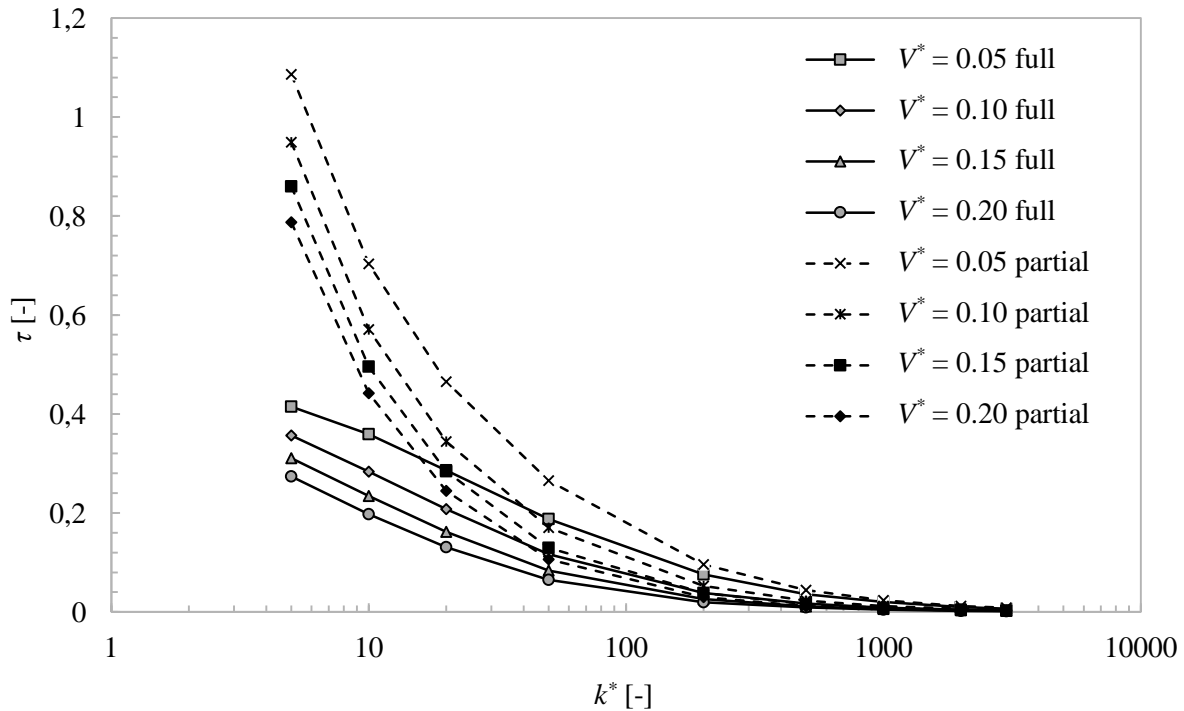
As with the previous boundary condition, an important part of this study is the effect of the conductivity ratio and the volume constraint. These parameters were once again extensively investigated. The input parameters are given in Table 6.11. Only the optimal placement of the base ( $x_1/L_D = 0$  and  $y_1/L_D = 0$ ) is considered here.

**Table 6.11:** Input parameters for practical boundary condition for a three-dimensional domain with a full Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$V^*$	0.05 – 0.2 (5% - 20%)
$k_L$	2.0 W/(mK)	$c/L_D, H_b/L_D$	0.24, 0.04
$k_H$	10 - 6000 W/(mK)	$M_x M_x M$	50x50x50
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.0, 0.0
Penalisation scheme	Increasing	$T_\infty$	0 K

Based on the experience obtained from the previous sections of this study, a reduced number of conductivity ratios was considered here. For the partial Dirichlet boundary condition, the volume constraint was tested up to  $V^* = 0.3$ , but this was not practical. For the full Dirichlet boundary condition, the volume constraint was tested up to  $V^* = 0.2$ . For each volume constraint, the same trend was observed, namely that there was a decrease in the maximum temperature with an increase in the conductivity ratio.

Figure 6.16 shows the effect of the volume constraint for all conductivity ratios for both the partial (dashed lines) and full Dirichlet boundary (solid lines). As with the partial Dirichlet boundary, there is again a large reduction in the maximum temperature when increasing the volume from 0.05 to 0.1, but smaller reduction of  $\tau$  from  $V^* = 0.1$  to  $V^* = 0.15$  and  $V^* = 0.2$ . Comparing the same volume constraints for the partial and full Dirichlet boundary, the full Dirichlet boundary obtains a lower maximum temperature for all values of  $k^*$ . This is again due to the larger temperature boundary that allows more heat to be extracted. The reduction between the partial and the full Dirichlet boundary was 89.6% (for  $k^* = 5$ ) and 34.5% (for  $k^* > 200$ ). The reduction dropped to 21.5% at  $k^* = 3\ 000$ . This was for the  $V^* = 0.05$  case.



**Figure 6.16:** The graph showing a comparison of all the volume constraints for a  $s_0 = 0.15$  for a three-dimensional domain with a full Dirichlet boundary.

The graph for the full Dirichlet boundary follows a similar pattern as with the partial Dirichlet boundary condition. There is, however, a slight S shape to the graph. For all values of  $k^*$ ,  $\tau$  is lower for the full Dirichlet boundary compared with the partial boundary. This is due to the size of the isothermal uniform temperature boundary of the full Dirichlet boundary, allowing more heat to be extracted through the bigger boundary compared with the smaller boundary of the partial Dirichlet boundary. The overall shape still shows that an increase in conductivity ratio yields a better maximum temperature for a full Dirichlet boundary. The volume constraint is satisfied for all values of  $k^*$ . The rest of the results can be found in Appendix B.2.4.

**Table 6.12:** Comparison of the conductivity ratio for  $V^* = 0.1$  and  $s_0 = 0.15$  for a three-dimensional full Dirichlet boundary ( $c/L_D = 0.24$ ,  $H_b/L_D = 0.04$ ,  $x_l/L_D = 0.0$  and  $y_l/L_D = 0.0$ ).

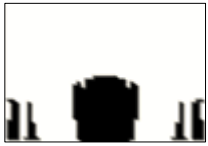





$k^* [-]$	5	50	500
$\tau [-]$	$3.567 \times 10^{-1}$	$1.164 \times 10^{-1}$	$1.693 \times 10^{-2}$
Density distribution			
$k^* [-]$	1000	2000	3000
$\tau [-]$	$8.839 \times 10^{-3}$	$4.618 \times 10^{-3}$	$3.293 \times 10^{-3}$
Density distribution			

Table 6.12 shows the converged architectures for different conductivity ratios. With the low conductivity ratios, there is only a small blob structure in the middle of the domain with some trees starting to form on the edges. These trees form because of the full Dirichlet boundary. Once again, as the conductivity ratio is increased, the number of side branches increases and the width of the main branches decreases. Interestingly in this case, there is little difference in the architectures for the high-conductivity ratios ( $k^* = 1\,000 - 3\,000$ ).

Table 6.13 shows the converged architectures for different volume constraints at  $k^* = 500$  for the partial and full Dirichlet boundary. For the full Dirichlet boundary, the results are similar to the partial Dirichlet boundary. When the volume constraint is increased, the width of the main branches increases as well as the number of side branches. The partial boundary has more secondary branches and a wider base at large volume constraint.

**Table 6.13:** Architectures for the different volume constraints at  $k^* = 500$  and  $s_0 = 0.15$  for a three-dimensional domain with a full Dirichlet boundary and a partial Dirichlet boundary.









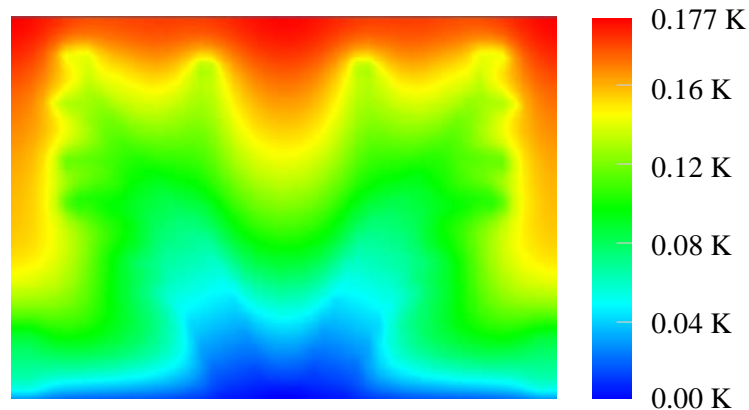
Full Dirichlet boundary				
$V^* [-]$	0.05	0.1	0.15	0.2
$\tau [-]$	$3.559 \times 10^{-2}$	$1.693 \times 10^{-2}$	$1.126 \times 10^{-2}$	$8.855 \times 10^{-3}$
Density distribution				
Partial Dirichlet boundary				
$V^* [-]$	0.05	0.1	0.15	0.2
$\tau [-]$	$4.420 \times 10^{-2}$	$2.268 \times 10^{-2}$	$1.538 \times 10^{-2}$	$1.192 \times 10^{-2}$
Density distribution				

Figure 6.17 shows the temperature distribution for a case with  $k^* = 500$  and  $V^* = 0.05$ . The full Dirichlet boundary is clearly visible here, with a low temperature seen on the whole bottom boundary. The architecture effectively extracts the heat generated in the volume.



**Figure 6.17:** Temperature distribution for  $k^* = 500$  for a three-dimensional domain with a full Dirichlet boundary.

## 6.5 Summary

The purpose of this chapter was to investigate a full Dirichlet boundary, which represented a more realistic boundary condition. To obtain tree-like structures, the domain had to be supplied with an initial base defined in a fixed subdomain, otherwise there would be no variation in the sensitivities, which would result in an even density distribution. It was found that an initial base with  $c/L_D = 0.24$  was optimal for any height of the base below  $H_b/L_D = 0.2$ . A test was done to find the optimal placement of the base structure and it was found that this was in the middle of the bottom boundary, as initially assumed.

The effect of  $s$  and  $s_0$  was again checked and it was found that the values used for the partial Dirichlet boundary were still sufficient. The number of nodes and iterations needed were again checked and it was found that the values used in the partial Dirichlet boundary were again sufficient.

With the tests explained above done, the optimisation runs were conducted for different conductivity ratios and volume constraints. The results for the full Dirichlet boundary are similar to the partial boundary, again showing that an increase in the volume or conductivity ratio decreases the maximum temperature. However, there is a marked decrease in the  $\tau$  values. The architectures obtained are very similar to the partial Dirichlet boundary, once again with a V shape for the main branches extending to the four top corners of the domain. There are, however, less secondary branches present for the full Dirichlet boundary. It can be concluded that it is possible to simulate a full Dirichlet boundary if an initial base is supplied to the domain.

## Chapter 7      **Three-Dimensional Methodology and Results for Multiple Bases**

### **7.1 Introduction**

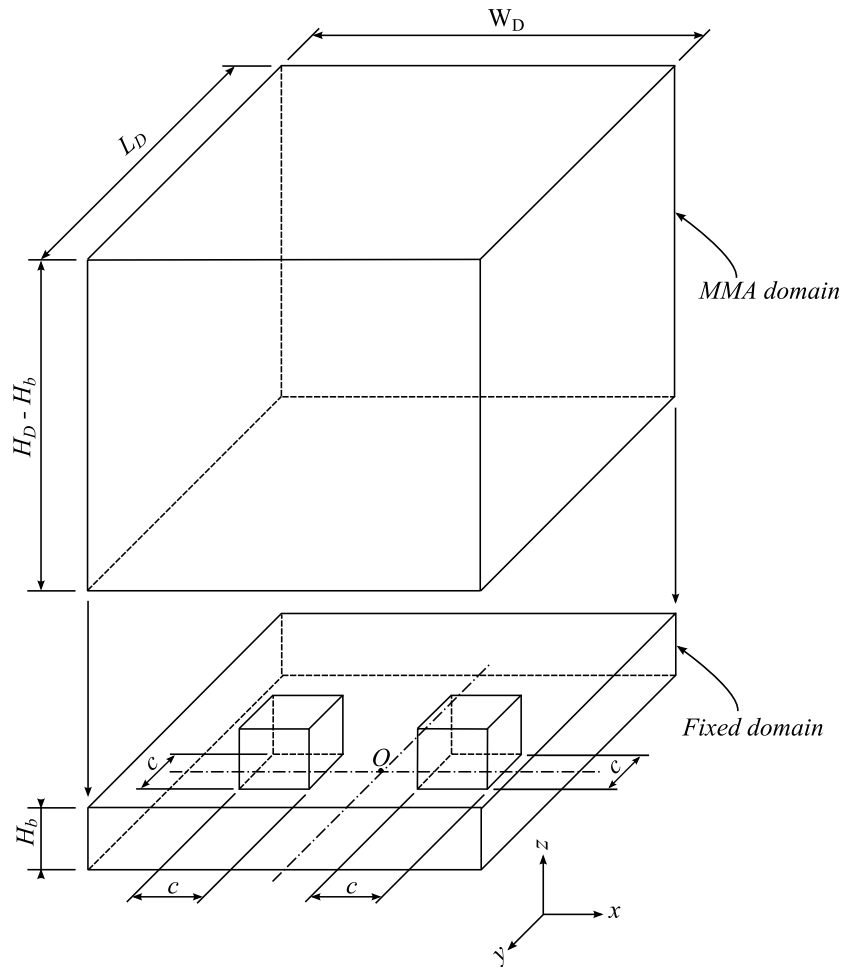
To further investigate the full Dirichlet boundary, more than one base can also be considered, where more than one seed location is provided. This is the logical next step, as it will give the MMA a chance to better distribute the material over the whole volume. For a start, two base structures are considered and are later-on extended to four. In this chapter, these conditions are investigated in a similar method as used in the previous chapter. First, an optimal placement of the blocks is assumed and the effect of the size of the blocks is tested. Once the optimal size is established, the blocks are moved around in the domain to find the optimal placement. With the optimal size and placement known, optimisation runs for the conductivity and volume constraint will be completed. For this chapter, only  $s = 0.85$  and  $s_0 = 0.15$  are considered.

### **7.2 Two Base Structures**

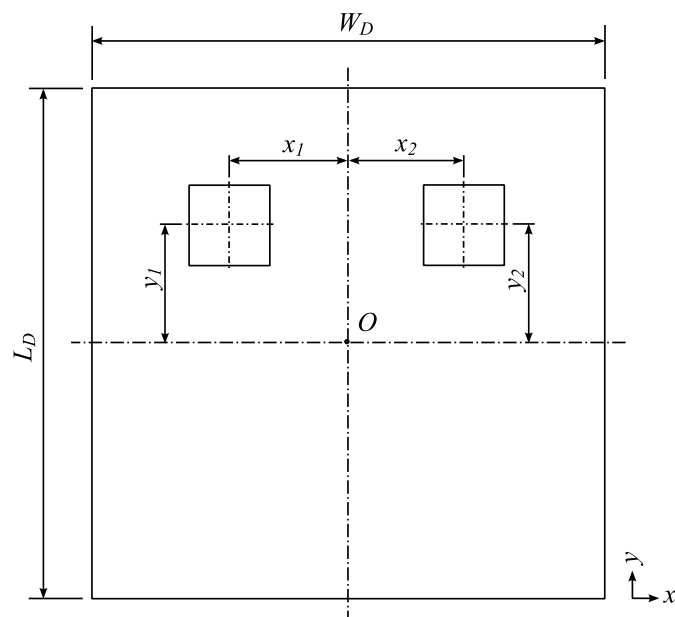
#### **7.2.1 Domain**

The domain for two base structures is shown in Figure 7.1. The same concept is used as with one initial base. Once again, the whole bottom boundary is set to a constant temperature, thus a full Dirichlet boundary. All other boundaries are adiabatic. The initial bases are identical in height and width and are placed according to the dimensions shown in Figure 7.2.

Initially, it is assumed that the optimal placement for the two base structures is in the centre of the respective half domain that the base occupies. Thus,  $y_1/L_D = y_2/L_D = 0.26$  and  $x_1/L_D = x_2/L_D = 0.0$  (or  $y_1/L_D = y_2/L_D = 0.0$  and  $x_1/L_D = x_2/L_D = 0.26$ ). This was later confirmed in Section 7.2.3.



**Figure 7.1:** Domain for two base structures for a three-dimensional domain with a full Dirichlet boundary.



**Figure 7.2:** The placement of the base structures.



## 7.2.2 Size of the Base Structures

As done in Section 6.3.2, the effect of two identically sized base structures is now considered. The height and width of the base are tested for a number of values. In this case, the maximum relative width was set as  $c/L_D = 0.48$ , otherwise the two blocks will merge into one rectangle. The height and width of the base are defined as in Figure 7.1 and Figure 7.2. The input parameters are shown in Table 7.1.

**Table 7.1:** Input parameters for the size of the two base structures for a three-dimensional domain with a full Dirichlet boundary.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$V^*$	0.05 (5%)
$k_L$	2.0 W/(mK)	$c/L_D$	0.04 – 0.48
$k_H$	1000.0 W/(mK)	$H_b/L_D$	0.04 – 1.0
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.26, 0.0
$M \times M \times M$	50x50x50	$x_2/L_D, y_2/L_D$	0.26, 0.0
Penalisation scheme	Increasing	$T_\infty$	0 K

The results follow the same trend as with one base for  $c$  and  $H_b$ . The optimum relative width in this case is  $c/L_D = 0.16$  for each base. This means that using two bases with  $c/L_D = 0.16$  would use twice the available volume of the constraint compared with using one initial base. An  $H_b/L_D$  value of 0.04 again gives the lowest maximum temperature for all values of  $c/L_D$ . Around the optimum  $c/L_D = 0.16$ , all values of  $H_b/L_D$  below 0.4 perform similarly. All converged volume ratios were close to the volume constraint. The results are shown in Appendix B.3.1.

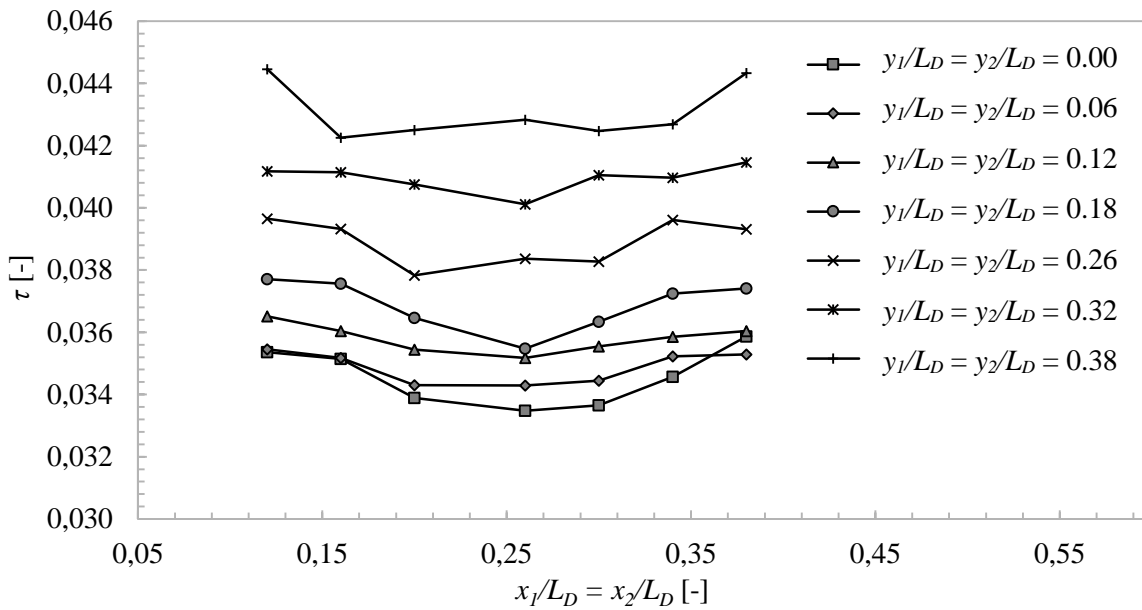
The optimum in this case was found at one conductivity and volume constraint. As done with one initial base, the optimum height of  $H_b/L_D = 0.04$  was tested at different conductivity and volume constraints for all values of  $c/L_D$ . The optimum found earlier performs significantly worse at low values of the conductivity ratio ( $k^* = 50$ ). For this reason, it was decided to use  $c/L_D = 0.24$  since it performs well for all conductivity ratios and volume constraints. Once again, the smallest value of  $H_b/L_D$  is used to give the MMA the maximum amount of freedom to solve. The results are shown in Appendix B.3.1.

## 7.2.3 Placement of the Base Structures

Using the height and width of the base structures obtained in the previous section of  $H_b/L_D = 0.04$  and  $c/L_D = 0.24$ , the optimal placement of the base structures can be studied. Since the problem is symmetrical, it was decided to keep the placement of the structures symmetrical as well. Two possible scenarios for the placement of the structures exist, moving both in the same half of the domain and moving the structures in different halves of the domain. The first scenario to be considered is when both structures are moved in the same half of the domain, as shown in Figure 7.2. To keep the positions symmetrical,  $x_1 = x_2$  and  $y_1 = y_2$ . The input parameters are shown in Table 7.2.

**Table 7.2:** Input parameters for the placement of the two base structures.

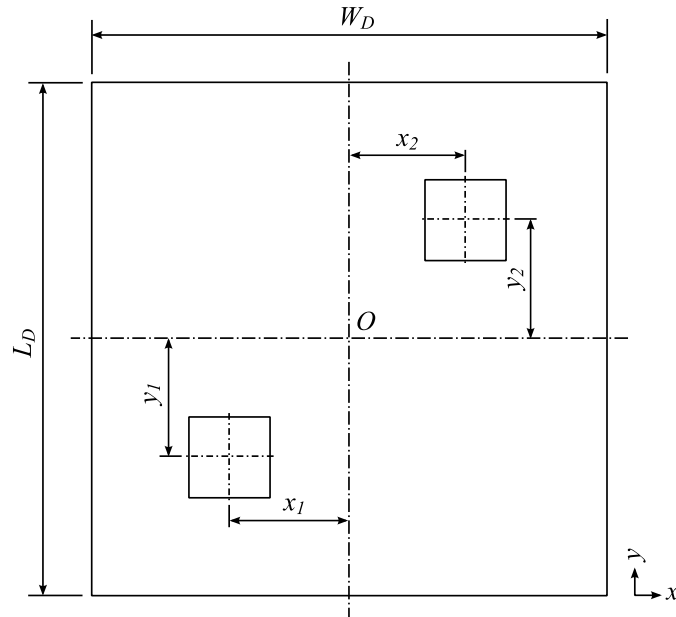
Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$T_\infty$	0 K
$k_L$	2.0 W/(mK)	$c/L_D, H_b/L_D$	0.24, 0.04
$k_H$	1000.0 W/(mK)	$M \times M \times M$	50x50x50
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D = x_2/L_D$	0.0 - 0.038
$V^*$	0.05 (5%)	$y_1/L_D = y_2/L_D$	0.0 - 0.038
Penalisation scheme	Increasing		



**Figure 7.3:** The effect on  $\tau$  when moving the two base structures in the same half of the domain.

Figure 7.3 shows that the optimal placement of the two base structures is in the centre of the halves of the domain, thus when  $x_1/L_D = x_2/L_D = 0.26$  and  $y_1/L_D = y_2/L_D = 0.0$  (or  $x_1/L_D = x_2/L_D = 0.0$  and  $y_1/L_D = y_2/L_D = 0.26$ ). This makes sense, since moving the block higher in the domain (increasing  $y_1/L_D$  and  $y_2/L_D$ ) forces the tree structures to grow asymmetrically. On one side of the domain, a higher temperature is obtained since the thermal path in the tree is longer. This again makes sense when viewing the placement problem in terms of the distances to the edge of the domain. The  $\tau$  value was, however, not as sensitive to the placement as was the case with the single base. If the domain is split in two, the optimum should be when the maximum absolute distance from a base to any edge of its half domain is minimal.

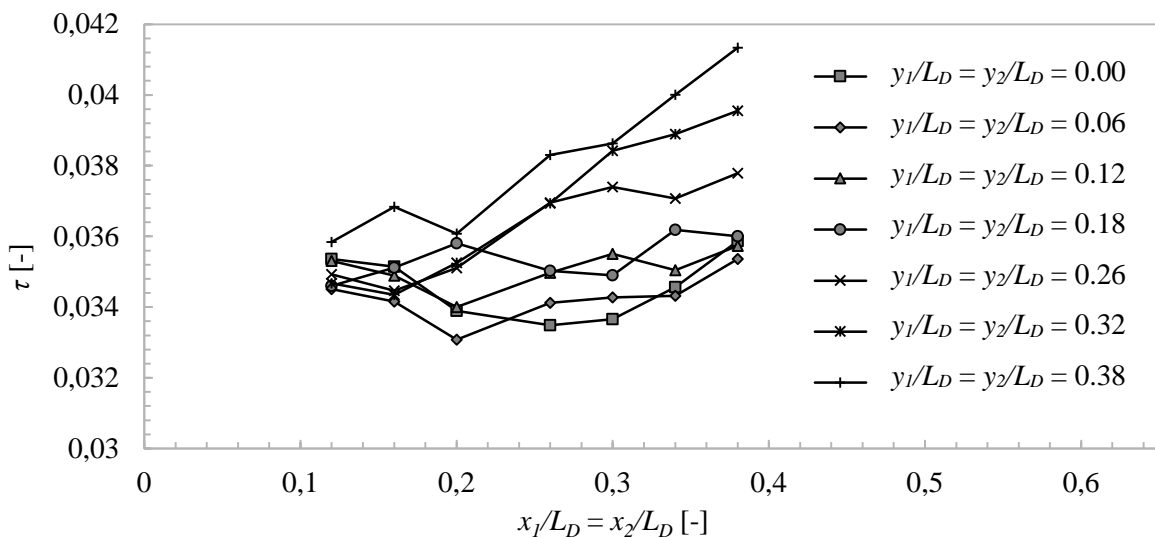
The other possible scenario for the placement of the structures is by moving them again symmetrically, but in opposite quadrants of the domain. This is represented in Figure 7.4. The input parameters for these simulations are the same as in Table 7.2.



**Figure 7.4:** Domain for moving two base structures in different halves of the domain.

Figure 7.5 shows the effect on  $\tau$  when moving the two base structures in opposite quadrants of the domain. The results show that the optimum is at  $x_1/L_D = x_2/L_D = 0.2$  and  $y_1/L_D = y_2/L_D = 0.06$  (or  $x_1/L_D = x_2/L_D = 0.2$  and  $y_1/L_D = y_2/L_D = 0.06$ ). The results show that as the blocks are moved further away from the origin, the temperature decreases up to about  $x_1/L_D = x_2/L_D = 0.2$ , after which the temperature starts to increase again.

When moving the blocks in the same half of the domain, the optimum is at  $x_1/L_D = x_2/L_D = 0.26$  and  $y_1/L_D = y_2/L_D = 0.0$ , again seen on this graph. The optimum for moving the blocks in opposite quadrants of the domain shown in Figure 7.5 performs only 1.3% better than the optimum found in Figure 7.3. The two optimum points also represent approximately the same base placement. It was decided to use the optimum from Figure 7.3, since the architecture is symmetrical and would be easier to manufacture.



**Figure 7.5:** The effect on  $\tau$  when moving the two base structures in the different halves of the bottom domain.

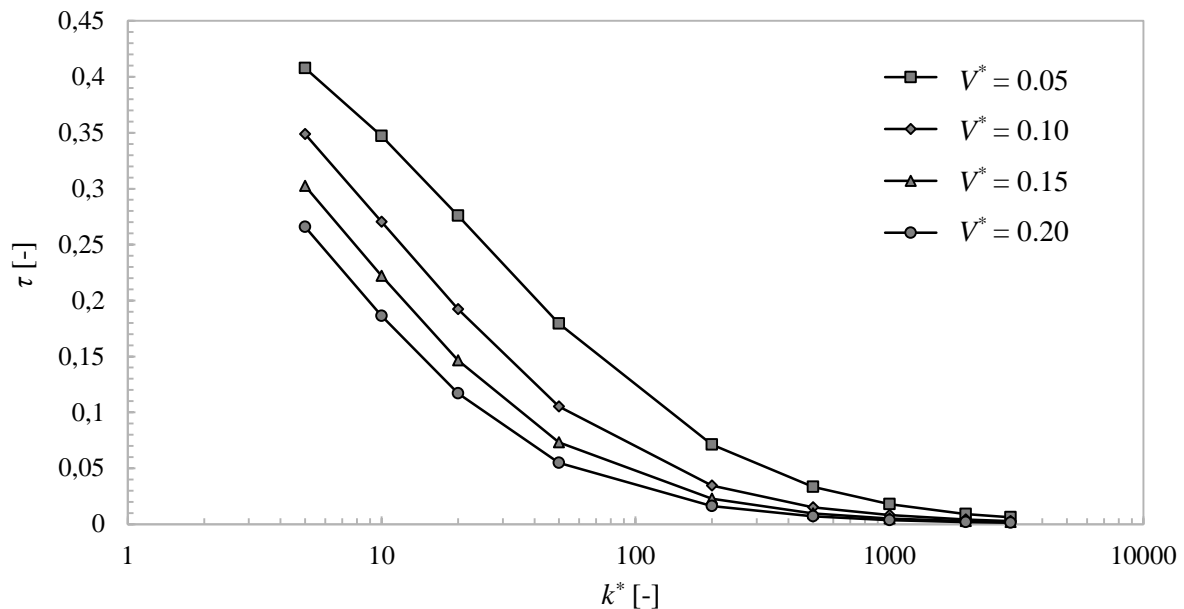
## 7.2.4 Results for Different Conductivity Ratios and Volume Constraints

With the optimum placement of the blocks now known, the optimisation runs could be done for varying volume constraints and conductivity ratios. The same intervals were used as before, with  $k^*$  ranging from 5 – 3 000 and  $V^*$  from 0.05 – 0.2. The purpose of these runs would be to compare the architectures and thermal performance to those obtained with one block. The input parameters used are given in Table 7.3.

**Table 7.3:** Input parameters for the simulations of varying  $k^*$  and  $V^*$ .

Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$V^*$	0.05 – 0.2 (5% - 20%)
$k_L$	2.0 W/(mK)	$c/L_D, H_b/L_D$	0.24, 0.04
$k_H$	10.0 – 6000.0 W/(mK)	$M \times M \times M$	50x50x50
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.26, 0.0
$T_\infty$	0 K	$x_2/L_D, y_2/L_D$	0.26, 0.0
Penalisation scheme	Increasing		

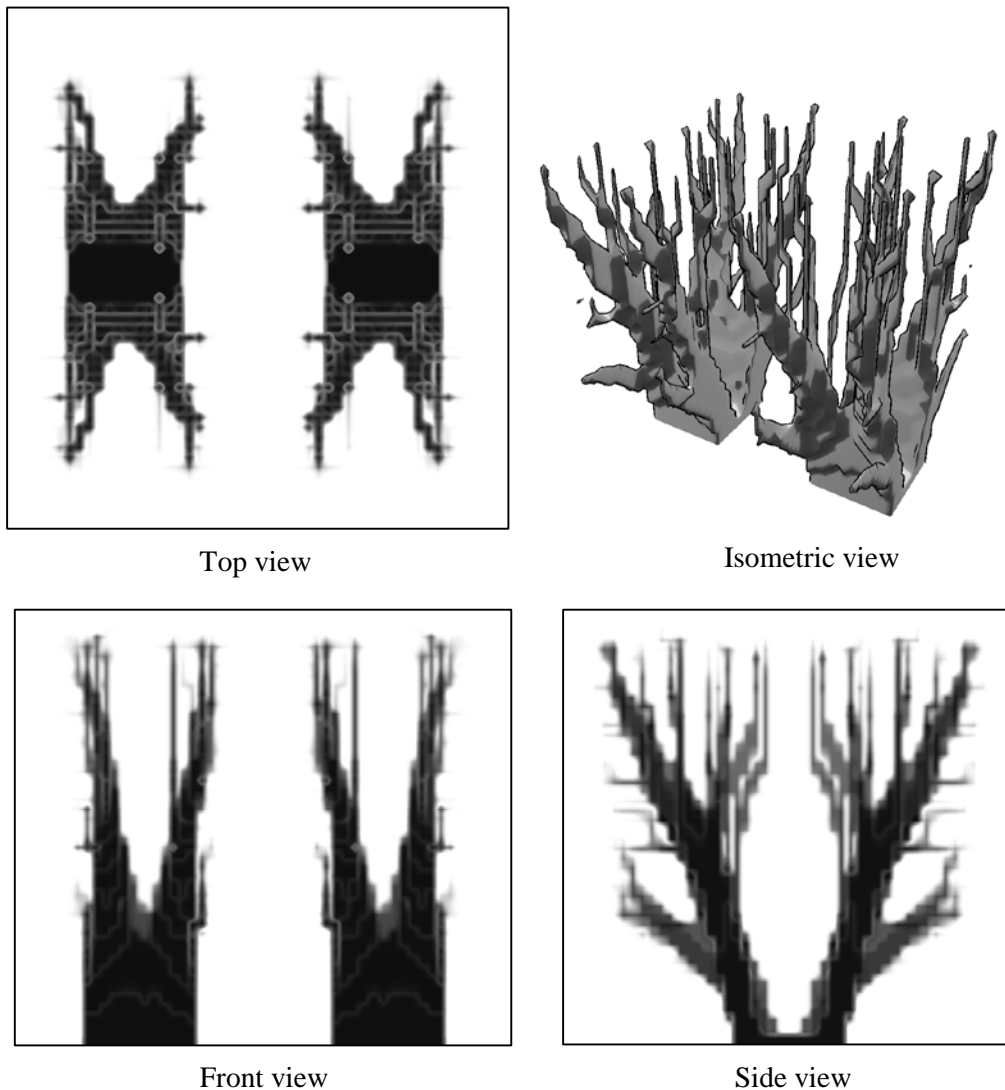
Figure 7.6 shows a comparison of all the volume constraints using two base structures. The results show that an increase in the volume constraint will decrease the maximum temperature, as expected. The results follow the same trend as with one base structure. The gain from the increase in the volume constraint again decreases as the conductivity ratio is increased. The plot of the volume constraint is not shown, since the volume constraint is satisfied within a reasonable margin for all cases. These results will be compared with one and four bases in Section 7.4.



**Figure 7.6:** Comparison of all the volume constraints for a three-dimensional domain with a full Dirichlet boundary and two initial bases.

Figure 7.7 shows a front, top, side and isometric view of a converged architecture with two initial bases. Each tree has four branches, as with one initial base. The primary branches are now much thinner, since the same volume constraint still applies. The primary branches grow normal to the sides

of the domain, as seen in the top view, different from a single base where the primary branches grow towards the corners of the domain. Secondary branches are only observed in the side view.

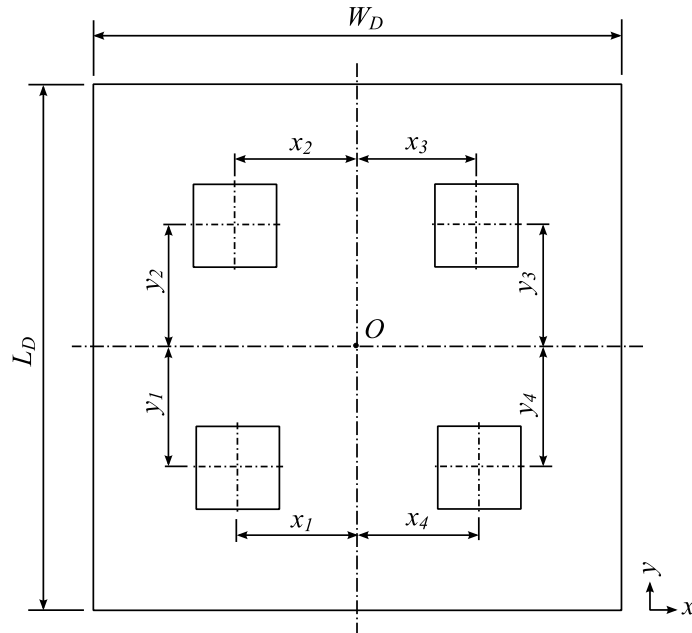


**Figure 7.7:** A converged architecture with two initial bases,  $k^* = 500$  and  $V^* = 0.05$ .

## 7.3 Four Base Structures

### 7.3.1 Domain

Figure 7.8 shows the domain when using four initial base structures. The concept is the same as in Figure 7.1, only now using four bases. The same methodology as with two bases will be followed in this section, namely find the optimal size and placement of the structures and then do the optimisation runs for different conductivity ratios and volume constraints.



**Figure 7.8:** The placement of the four base structures.

### 7.3.2 Size of the Base Structures

For the size of the base structures, it is initially assumed that the optimal placement of the initial base is just off-centre in each quadrant. This small deviation from the centre is due to the node size and placement. The input parameters are shown in Table 7.4.

**Table 7.4:** Input parameters for the size of the base structure, using four bases.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$V^*$	0.05 (5%)
$k_L$	2.0 W/(mK)	$c/L_D$	0.04 – 0.48
$k_H$	1000.0 W/(mK)	$H_b/L_D$	0.04 – 1.0
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.26, 0.26
$M \times M \times M$	50, 50, 50	$x_2/L_D, y_2/L_D$	0.26, 0.26
$T_\infty$	0 K	$x_3/L_D, y_3/L_D$	0.26, 0.26
Penalisation scheme	Increasing	$x_4/L_D, y_4/L_D$	0.26, 0.26

The results follow the same pattern as with one and two bases. The lowest value of  $H_b/L_D$  again performs the best for all values of  $c/L_D$ . The optimum with four initial bases is  $c/L_D = 0.12$  and  $H_b/L_D = 0.04$ . The lowest value of  $H_b/L_D$  is once again chosen to give the MMA the maximum amount of freedom. The results are shown in Appendix B.4.1.

The effect of  $c/L_D$  for different conductivity ratios and volume constraints and  $H_b/L_D = 0.04$  was again checked. The results show that the optimum dimensions for the bases chosen before ( $c/L_D = 0.12$  and  $H_b/L_D$ ) are sufficient.

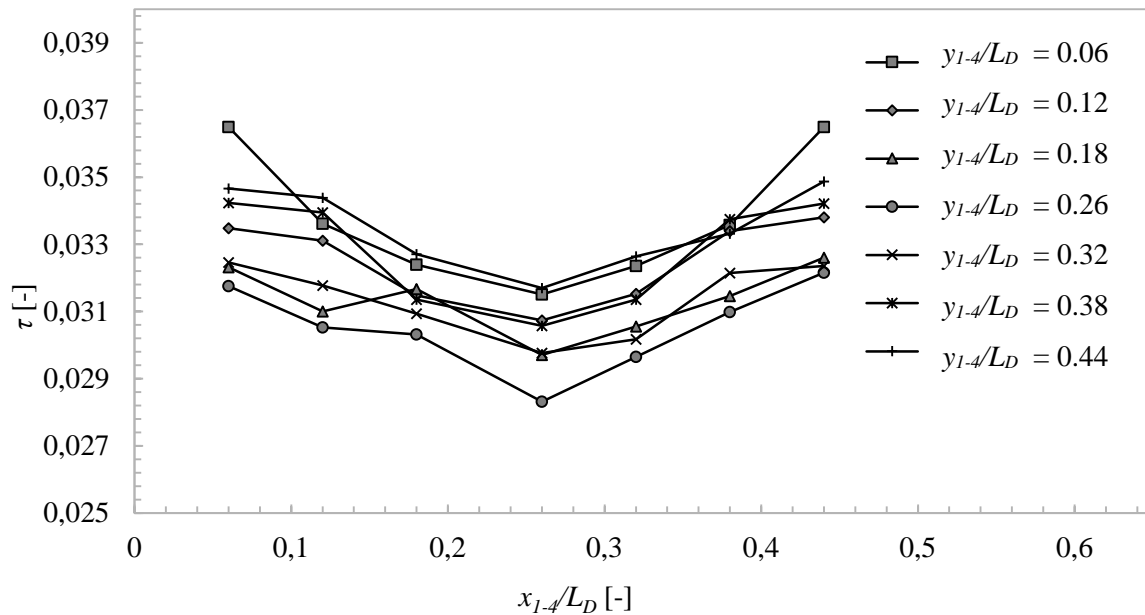
### 7.3.3 Placement of the Base Structures

In the beginning of this subsection, it was assumed that the optimum placement of the bases was just off-centre of each quadrant. To validate this assumption, the bases were moved symmetrically in the domain as shown in Figure 7.8. The input parameters are shown in Table 7.5.

**Table 7.5:** Input parameters for the placement of the base structures, using four initial bases.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$V^*$	0.05 (5%)
$k_L$	2.0 W/(mK)	$c/L_D$	0.12
$k_H$	1000.0 W/(mK)	$H_b/L_D$	0.04
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.06 - 0.44
$M \times M \times M$	50, 50, 50	$x_2/L_D, y_2/L_D$	0.06 - 0.44
$T_\infty$	0 K	$x_3/L_D, y_3/L_D$	0.06 - 0.44
Penalisation scheme	Increasing	$x_4/L_D, y_4/L_D$	0.06 - 0.44

Figure 7.9 shows the effect of the placement of the base structures, using four initial bases. The graph shows that the initial assumption of the placement of the base structures was correct. The optimal placement is at  $y_{1-4}/L_D = 0.26$  and  $x_{1-4}/L_D = 0.26$ . The graph shows that the maximum temperature increases as the block is moved further away from the centre of the quadrant.



**Figure 7.9:** The effect on  $\tau$  when moving the bases symmetrically in the domain.

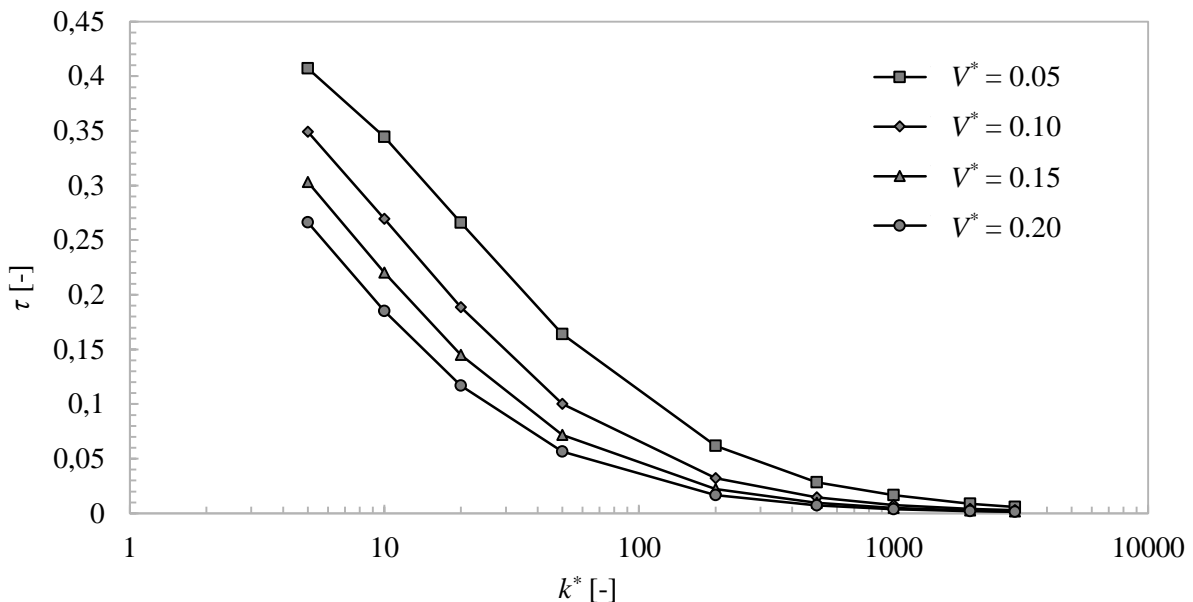
### 7.3.4 Results for Different Conductivity Ratios and Volume Constraints

With the optimum placement of the blocks now known, the optimisation runs could be done for varying volume constraints and conductivity ratios. The same intervals were used as before, with  $k^*$  ranging from 5 – 3 000 and  $V^*$  from 0.05 – 0.2. The purpose of these runs would be to compare the

architectures and thermal performance with those obtained with one and two blocks. The input parameters used are given in Table 7.6.

**Table 7.6:** Input parameters for different conductivity ratios and volume constraints for a three-dimensional domain with a full Dirichlet boundary, using four initial bases.

Parameter	Value	Parameter	Value
$L_D$	1.0 m	$p_f$	3
$W_D$	1.0 m	$I$	60 iterations
$H_D$	1.0 m	$V^*$	0.05 (5%)
$k_L$	2.0 W/(mK)	$c/L_D$	0.12
$k_H$	10.0 - 6000.0 W/(mK)	$H_b/L_D$	0.04
$q_H$	10.0 W/m <sup>3</sup>	$x_1/L_D, y_1/L_D$	0.26, 0.26
$M_x M_y M_z$	50, 50, 50	$x_2/L_D, y_2/L_D$	0.26, 0.26
$T_\infty$	0 K	$x_3/L_D, y_3/L_D$	0.26, 0.26
Penalisation scheme	Increasing	$x_4/L_D, y_4/L_D$	0.26, 0.26



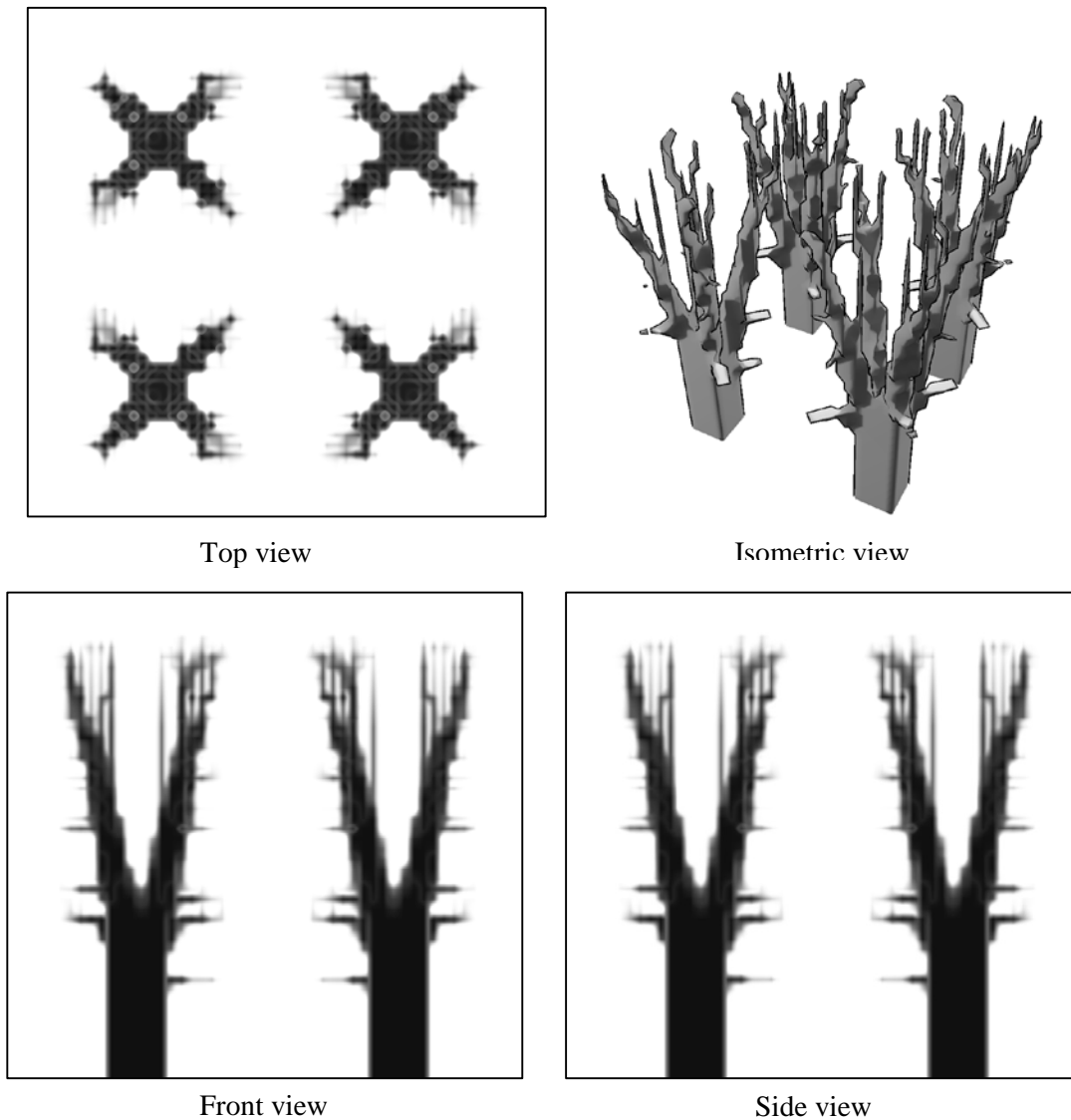
**Figure 7.10:** Comparison of all the volume constraints for a three-dimensional domain with a full Dirichlet boundary and four initial bases.

Figure 7.10 shows a comparison of all the volume constraints using four base structures. The results show that an increase in the volume constraint will decrease the maximum temperature, as expected. The results follow the same trend as with one and two base structures. The gain from the increase in the volume constraint again decreases as the conductivity ratio is increased. The plot of the volume constraint is not shown, since the volume constraint is satisfied within a reasonable margin for all cases. These results will be compared with those of one and two bases in Section 7.4.

Figure 7.11 shows a front, side, top and isometric view of a converged architecture, using four initial bases. Each tree has a cross-shape, as seen in the top view, similar to when one base was used. The main branches extend to the corners of each tree's respective quarter domain. This is also observed in the front view. There were secondary branches observed, growing parallel to the bottom boundary, but much less than when using one and two bases.



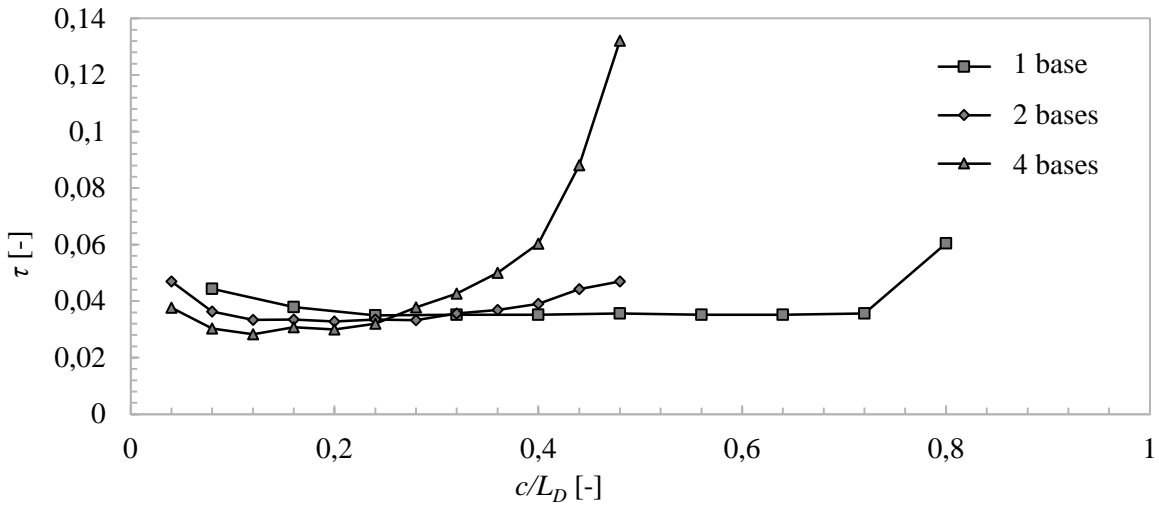
The initial growth of the trees is interesting. Although the fixed domain height,  $H_b/L_D$ , is only limited to 0.04, the architecture shows that the primary and secondary branches only appear after  $H_b/L_D > 0.25$ . This was not observed with one or two bases.



**Figure 7.11:** A converged architecture using four initial bases, with  $k^* = 500$  and  $V^* = 0.05$ .

#### 7.4 Comparison of Thermal Results for Different Number of Initial Bases

Figure 7.12 shows a comparison of the thermal performance of one, two and four initial bases for  $H_b/L_D = 0.4$ , for different values of  $c/L_D$ . Two and four bases perform better than one base, for  $c/L_D$  values less than 0.24. After this point, four bases perform the worst followed by two bases and one base, which performs the best. This is due to the area taken on the bottom boundary. After that specific point, two and four bases cover most of the bottom boundary, raising the issue again of no variation in the sensitivity distribution in the  $x$ - $y$ -plane.

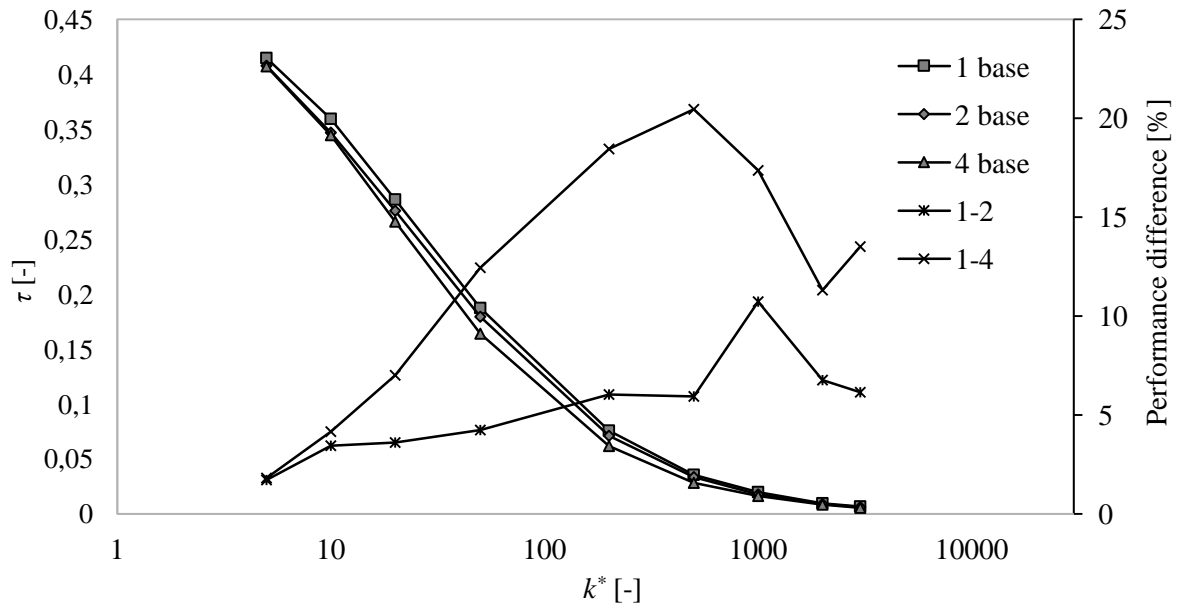


**Figure 7.12:** Comparison of the size of the base structure for different values of  $c/L_D$  and  $H/L_D = 0.04$ , for one, two and four bases.

Figure 7.13 shows a comparison of one, two and four bases for an arbitrarily chosen  $V^* = 0.05$ . From the figure, it is evident that two bases perform better in terms of temperature than one base, and four bases perform better than two. The figure also shows the percentage difference between one and two bases, two and four bases as well as the difference between one and four bases.

$$Performance\ difference(\%) = \frac{|\tau_1 - \tau_2|}{\tau_1} * 100 \quad (7.1)$$

Here subscript 1 refers to the single-seed case and subscript 2 to the two-seed case (or four-seed case). For low conductivity values ( $k^* \leq 20$ ), the performance difference when using two or four bases compared to one is less significant. In the mid-conductivity range, two and four bases perform significantly better than one base. This effect is again less in the higher ranges of the conductivity ratio ( $k^* \geq 1000$ ). Overall, using more than one base performs better for all conductivity ratios compared with one initial base. A decrease of 20.4 % in the maximum temperature was observed for  $k^* = 500$  when four bases are used compared to only one base.



**Figure 7.13:** A thermal performance comparison of one, two and four bases with the effect on  $\tau$  for  $V^* = 0.05$  and varying values of  $k^*$  for a three-dimensional domain with full Dirichlet boundary. The graph also shows the % difference between one and two bases and one and four bases.

## 7.5 Summary

In this chapter, multiple initial bases were investigated using a three-dimensional domain with a full Dirichlet boundary. It was found that the optimal placement when using two initial bases was in the centre of each respective half and in the centre of each quadrant when using four bases. Comparing the sizes of the base structures, as the number of base structures was increased, the optimum width of the base structure decreased.

For all conductivity ratios and volume constraints, two bases performed better than one base and four bases performed better than two bases. A maximum decrease in  $\tau$  of 20.4 % was observed (comparing one base with four bases with  $V^* = 0.05$  and  $k^* = 500$ ). In the next chapter, the conclusions and recommendations for this study will be made.

## Chapter 8      **Conclusions and Recommendations**

The purpose of this study was to investigate topology optimisation in three dimensions for use in passive internal electronic cooling. The temperature distribution was solved using the finite volume method and a harmonic mean was used to calculate the conductivity ratio at the interface of elements. To calculate the sensitivities of the objective function, the adjoint method was used. To find the optimal material distribution, the method of moving asymptotes was used to minimise the objective function (average temperature).

A two-dimensional study was first investigated in order to build a solid base for the three-dimensional simulations and to verify the code with other papers, using a partial Dirichlet boundary condition. It was found that the asymptote parameters,  $s$  and  $s_0$ , had a significant effect on the converged maximum temperature. For the two-dimensional case, it was found that using  $s = 0.9$  and  $s_0 = 0.1$  performed significantly better than using the default parameters set out in the algorithm. A mesh-dependence study was also conducted and  $M = 100$  elements was found suitable. The converted architectures all followed the same pattern, i.e.: with two main branches extending to the upper corners of the domain and smaller secondary branches protruding from the main branches. The size and number of these secondary branches depended on the number of elements in the domain. The code was also compared against a popular paper by Gersborg-Hansen *et al.* [8]. The results were similar, but there were some minor differences in the secondary branches, which can be explained by the difference in methods used (FVM vs. FEM).

With the two-dimensional code investigated and validated, the three-dimensional domain could be investigated again using a partial Dirichlet boundary. It was found that the asymptote parameters again had a significant effect on the maximum temperature. For this specific condition,  $s = 0.85$  and  $s_0 = 0.15 - 0.25$  performed the best. A mesh- and iteration-dependence study was conducted for the three-dimensional domain and it was found that  $M = 50$  and  $I = 60$  were sufficient. The effect of constant and incremental increasing penalisation was also investigated. Constant penalisation proved to be detrimental to the thermal performance. Incremental increasing penalisation performed significantly better and  $p = 3$  proved to give the best relationship between a defined architecture and a low maximum temperature. The effect of the initial density distribution was also checked and it was found that the MMA is very sensitive to the initial distribution. Using a random initial density distribution was detrimental to the thermal performance and the final architecture. Using a uniform density distribution proved to be sufficient. With all the above parameters checked, optimisation runs for different conductivity ratios and volume constraints were done. It was found that increasing the volume and conductivity ratio decreased the maximum temperature. Increasing the volume constraint did show a decrease in the maximum temperature, although this effect was less at higher volume constraints. The architectures found were very similar to the architecture obtained in two dimensions. In this case, there are four main branches extending to the upper corners of the domain. Secondary branches protruded from the main branches, again varying in size and frequency depending on the iteration and element count.

The partial Dirichlet boundary condition which was considered first, may not be a realistic representation of a typical boundary. To represent the boundary more realistically, a full Dirichlet boundary condition was investigated. The challenge with this boundary was that if the MMA was allowed to modify the whole domain, no discernible architecture would be obtained since there would be no variation in the  $x$ - $y$ -plane of sensitivities. To counteract this problem, a part of the domain was fixed and supplied with an initial base. At first it was assumed that the optimal placement of this initial base was in the middle of the bottom boundary. The width and height of this base were investigated and it was found that  $c/L_D = 0.24$  and  $H_b/L_D = 0.04$  were optimal. The placement of the base was checked and it was found that the initial assumed position was correct. The effect of the asymptote parameters was checked again and  $s = 0.85$  and  $s_0 = 0.15 - 25$  were still sufficient. The mesh- and iteration-dependence study showed that  $M = 50$  and  $I = 60$  were still sufficient for the convergence of the maximum temperature. With the above parameters checked, the optimisation runs for different conductivity ratios and volume constraints could be completed. The full Dirichlet boundary condition followed the same pattern as the partial boundary. There was again a decrease in the maximum temperature with an increase in the conductivity ratio and volume constraint. The maximum temperature was lower compared with the partial boundary due to the larger isothermal boundary, allowing more flux and thus more heat to be extracted from the volume. The architecture obtained with the full Dirichlet boundary condition was very similar to that of the partial boundary. The four main branches again extended to the top four corners of the domain. There were less secondary branches compared with those of the partial boundary and the base of the structure was also smaller.

With the full Dirichlet boundary fully investigated, it was decided to investigate the effect of multiple base structures. Two base structures were first checked and it was assumed that the optimal placement of these structures was in the middle of each respective half. This was later confirmed. The size of the structure was investigated and it was found that  $c/L_D = 0.16$  and  $H_b/L_D = 0.04$  were optimal. This was unfortunately not true for all conductivity ratios and volume constraints. It was found that  $c/L_D = 0.24$  and  $H_b/L_D = 0.04$  performed better for most conductivity ratios and volume constraints. With the above known, optimisation runs for different conductivity ratios and volume constraints were completed. There was again a trend showing that an increase in the conductivity ratio and volume constraint decreased the maximum temperature.

With two initial bases completed, four initial bases were also investigated. It was assumed that the optimal placement of these bases was in the centre of each respective quadrant. This was later confirmed. It was found that the optimal sizes when using four bases were  $c/L_D = 0.12$  and  $H_b/L_D = 0.04$  for most conductivity ratios and volume constraints. The optimisation runs for different conductivity ratios and volume constraints were completed and it followed the same trend as with two bases.

The effect of multiple bases was compared and it was found that using two bases over one base always yielded a performance increase. Using four bases over two also always yielded a performance increase. A maximum of 20.4% decrease in the maximum temperature was observed when comparing one to four bases ( $k^* = 500$ ,  $V^* = 0.05$ ). It is expected that a performance increase will be observed if the number of initial bases is increased. On the other hand, this will increase the complexity of manufacturing.

It can, finally, be concluded that topology optimisation can be used to solve the heat conduction problem in three dimensions using either a partial or full Dirichlet boundary. Although this

investigation is purely theoretical, it shows that there is definitely a need to further investigate this method of cooling over a conventional cooling structure.

The author recommends that the following should be investigated in future studies:

1. Investigate different objective functions, for example, maximum temperature;
2. Investigate different boundary conditions, for example, heat flux or multiple isothermal cold spots, be it partial or full;
3. Investigate the increase in the number of initial bases to see if the increase in performance will continue with the increase in initial bases;
4. Investigate the comparison of topology optimisation with constructal theory and other topological methods for a full Dirichlet boundary;
5. Investigate  $s$  and  $s_0$  for different input parameters.

## References

- [1] “First Law of Thermodynamics”, *Georgia State University*. [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbase/thermo/firlaw.html>. [Accessed: 04-Dec-2012].
- [2] M. A. (MD) Verjee, “Recognizing Heat Stroke”, *Weill Cornell Medical College*. [Online]. Available: <http://qatar-weill.cornell.edu/media/intoHealth/InsightsHeatStroke.html>. [Accessed: 04-Feb-2012].
- [3] Y. A. Cengel, *Heat and Mass Transfer*, 3rd ed. New York: McGraw-Hill, 2006.
- [4] K. Azar, “Power Consumption and Generation in the Electronics Industry - A Perspective”, *Semiconductor Thermal Measurement and Management Symposium*, pp. 201–212, 2000.
- [5] J. Dirker, A. G. Malan, and J. P. Meyer, “Thermal Characterisation of Rectangular Cooling Shapes in Solids”, *International Journal of Numerical Methods for Heat Fluid and Flow*, vol. 17, no. 4, pp. 361–383, 2007.
- [6] J. Dirker and J. P. Meyer, “Heat Removal from Power Electronics in Two Direction Sets Using Embedded Solid State Cooling Layers – A Proposed Non-numerical Calculation Method”, *Journal for Heat Transfer Engineering*, vol. 30, no. 6, pp. 452–465, 2009.
- [7] J. Dirker and J. P. Meyer, “Thermal Characterisation of Embedded Cooling Layers in Rectangular Heat-Generating Electronic Modules”, *International Journal for Heat and Mass Transfer*, vol. 52, no. 5–6, pp. 1374–1384, 2009.
- [8] A. Gersborg-Hansen, M. P. Bendsøe, and O. Sigmund, “Topology Optimization of Heat Conduction Problems Using The Finite Volume Method”, *Structural and Multidisciplinary Optimization*, vol. 31, no. 4, pp. 251–259, Mar. 2006.
- [9] Y. Zhang and S. Liu, “Design of Conducting Paths Based on Topology Optimization”, *Heat and Mass Transfer*, vol. 44, no. 10, pp. 1217–1227, Dec. 2008.
- [10] J. Dirker and J. P. Meyer, “Topology Optimization for an Internal Heat-Conduction Cooling Scheme in a Square Domain for High Heat Flux Applications”, *Submitted to: Journal of Heat Transfer (Manuscript ID: HT-12-1239)*.
- [11] X. Cheng, Z. Li, and Z. Guo, “Constructs of Highly Effective Heat Transport Paths by Bionic Optimization”, *Science in China Series E*, vol. 46, no. 3, pp. 296–302, 2003.
- [12] A. Bejan, “Constructal-theory Network of Conducting Paths for Cooling a Heat Generating Volume”, *International Journal of Heat and Mass Transfer*, vol. 40, no. 4, pp. 799–811, 813–816, 1997.
- [13] G. A. Ladezma and A. Bejan, “Constructal Three-Dimensional Trees for Conduction Between a Volume and a Point”, *Journal of Heat Transfer*, vol. 120, no. 4, pp. 977–984, 1998.

- [14] M. Neagu and A. Bejan, “Three-Dimensional Tree Constructs of ‘Constant’ Thermal Resistance”, *Journal of Applied Physics*, vol. 86, no. 12, pp. 7107–7115, 1999.
- [15] H. Feng, L. Chen, and F. Sun, ““Volume-Point” Heat Conduction Constructal Optimization Based on Entropy Dissipation Rate Minimization With Three-Dimensional Cylindrical Element and Rectangular and Triangular Elements on Microscale and Nanoscale”, *Science China Technological Sciences*, vol. 55, no. 3, pp. 779–794, 2012.
- [16] M. P. Bendsøe and N. Kikuchi, “Generating optimal topologies in structural design using a homogenization method”, *Computer Methods in Applied Mechanics and Engineering*, pp. 197–224, 1998.
- [17] H. A. Eschenauer and N. Olhoff, “Topology optimization of continuum structures: A review”, *Applied Mechanics Reviews*, vol. 54, no. 4, pp. 331–390, 2001.
- [18] C. Zhuang, Z. Xiong, and H. Ding, “A Level Set Method for Topology Optimization of Heat Conduction Problem Under Multiple Load Cases”, *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4–6, pp. 1074–1084, 2007.
- [19] M. P. Bendsøe and O. Sigmund, *Topology Optimization - Theory, Methods and Applications*, 2nd ed. Berlin: Springer-Verlag, 2003.
- [20] O. Sigmund and J. Peterson, “Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing with Checkerboards, Mesh-Dependencies and Local Minima”, *Structural Optimization*, vol. 16, no. 1, pp. 68–75, 1998.
- [21] X. Huang and M. Xie, *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*, 1st ed. United Kingdom: Wiley, 2010.
- [22] A. Iga, S. Nishiwaki, K. Izui, and M. Yoshimura, “Topology Optimization for Thermal Conductors Considering Design-Dependent Effects, Including Heat Conduction and Convection”, *International Journal of Heat and Mass Transfer*, vol. 52, no. 11–12, pp. 2721–2732, May 2009.
- [23] J. Haslinger, A. Hillebrand, T. Kärkkäinen, and M. Miettinen, “Optimization of Conducting Structures by Using the Homogenization Method”, *Structural and Multidisciplinary Optimization*, vol. 24, no. 2, pp. 125–140, Sep. 2002.
- [24] Q. Li, G. P. Steven, O. M. Querin, and Y. M. Xie, “Shape and Topology Design for Heat Conduction by Evolutionary Structural Optimization”, *International Journal of Heat and Mass Transfer*, vol. 42, no. 1999, pp. 3361–3371, 1999.
- [25] Q. Li, G. Steven, Y. Xie, and O. Querin, “Evolutionary Topology Optimization for Temperature Reduction of Heat Conducting Fields”, *International Journal of Heat and Mass Transfer*, vol. 47, no. 23, pp. 5071–5083, Nov. 2004.
- [26] T. Gao, W. H. Zhang, J. H. Zhu, Y. J. Xu, and D. H. Bassir, “Topology Optimization of Heat Conduction Problem Involving Design-Dependent Heat Load Effect”, *Finite Elements in Analysis and Design*, vol. 44, no. 14, pp. 805–813, 2008.
- [27] R. Boichot, L. Luo, and Y. Fan, “Tree-Network Structure Generation for Heat Conduction by Cellular Automaton”, *Energy Conversion and Management*, vol. 50, no. 2, pp. 376–386, 2009.



- [28] S. H. Ha and S. Cho, “Topological Shape Optimization of Heat Conduction Problems using Level Set Approach”, *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, vol. 48, no. 1, pp. 67–88, Jul. 2005.
- [29] A. Bejan, *Advanced Engineering Thermodynamics*, 2nd ed. Wiley, 1997, p. 896.
- [30] M. Neagu and A. Bejan, “Constructal Placement of High-Conductivity Inserts in a Slab: Optimal Design of ‘Roughness’”, *Journal of Heat Transfer*, vol. 123, no. 6, pp. 1184–1189, 2001.
- [31] M. Almgöbel and A. Bejan, “Constructal Optimization of Nonuniformly Distributed Tree-Shaped Flow Structures for Conduction”, *International Journal of Heat and Mass Transfer*, vol. 44, no. 22, pp. 4185–4194, 2001.
- [32] A. Bejan and M. Almgöbel, “Constructal T-shaped Fins”, *International Journal of Heat and Mass Transfer*, vol. 43, no. 12, pp. 2101–2115, 2000.
- [33] A. Alebrahim and A. Bejan, “Constructal Trees of Circular Fins for Conductive and Convective Heat Transfer”, *International Journal of Heat and Mass Transfer*, vol. 42, no. 19, pp. 3585–3597, 1999.
- [34] L. A. O. Rocha, S. Lorente, and A. Bejan, “Conduction Tree Networks with Loops for Cooling a Heat Generating Volume”, *International Journal of Heat and Mass Transfer*, vol. 49, no. 15–16, pp. 2626–2635, 2006.
- [35] A. K. da Silva, C. Vasile, and A. Bejan, “Disc Cooled With High-Conductivity Inserts That Extend Inward From the Perimeter”, *International Journal of Heat and Mass Transfer*, vol. 47, no. 19–20, pp. 4257–4263, 2004.
- [36] L. Gosselin and A. Bejan, “Constructal Heat Trees at Micro and Nanoscales”, *Journal of Applied Physics*, vol. 96, no. 10, pp. 5852–5859, 2004.
- [37] S. Redner, “Fractal and Multifractal Scaling of Electrical Conduction in Random Resistor Networks Definition of the Subject”, *Center for Polymer Studies and Department of Physics*, pp. 1–26.
- [38] E. Green, “Introduction to Fractal Theory”, 1998. [Online]. Available: [http://pages.cs.wisc.edu/~ergreen/honors\\_thesis/fractal.html](http://pages.cs.wisc.edu/~ergreen/honors_thesis/fractal.html). [Accessed: 19-Jan-2012].
- [39] W. Wang, X. L. Huai, and Y. Tao, “Heat Conduction and Characteristic Size of Fractal Porous Media”, *Chinese Physics Letters*, vol. 23, no. 6, pp. 1511–1514, 2006.
- [40] X. Huai, W. Wang, and Z. Li, “Analysis of the Effective Thermal Conductivity of Fractal Porous Media”, *Applied Thermal Engineering*, vol. 27, no. 17–18, pp. 2815–2821, Dec. 2007.
- [41] O. de Weck and I. Y. Kim, “Finite Element Method”, 2004. [Online]. Available: [http://web.mit.edu/16.810/www/16.810\\_L4\\_CAE.pdf](http://web.mit.edu/16.810/www/16.810_L4_CAE.pdf). [Accessed: 17-Jan-2012].
- [42] H. M. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics*, 2nd ed. Essex: Pearson Education Limited, 2007.

- [43] W. H. Gray and N. M. Schnurr, “Comparison of the Finite Element and Finite Difference Methods for Analysis in the Steady Two Dimensional Heat Conduction Problems”, *Computer Methods in Applied Mechanics and Engineering*, vol. 6, no. 2, pp. 243–245, 1975.
- [44] T. N. Croft, “Unstructured Mesh - Finite Volume Algorithms for Swirling, Turbulent Reaction Flows”, University of Greenwich, London, 1998.
- [45] S. G. Johnson, “Notes on adjoint methods for 18.336”, *October*. pp. 1–7, 2007.
- [46] G. Strang, *Computational Science and Engineering*. Wellesley: Wellesley-Cambridge Press, 2007, p. 678.
- [47] T. E. Bruns, “Topology Optimization of Convection-Dominated, Steady-State Heat Transfer Problems”, *International Journal of Heat and Mass Transfer*, vol. 50, no. 15–16, pp. 2859–2873, Jul. 2007.
- [48] K.-T. Zuo, L.-P. Chen, Y.-Q. Zhang, and J. Yang, “Study of Key Algorithms in Topology Optimization”, *The International Journal of Advanced Manufacturing Technology*, vol. 32, no. 7–8, pp. 787–796, 2007.
- [49] K. Svanberg, “The Method for Moving Asymptotes - A New Method for Structural Optimization”, *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 359–373, 1987.
- [50] L. F. P. Etman, A. A. Groenwold, and J. E. Rooda, “First-Order Sequential Convex Programming Using Approximate Diagonal QP Subproblems”, *Structural and Multidisciplinary Optimization*, vol. 45, no. 4, pp. 479–488, 2012.
- [51] M. Bruyneel, P. Duysinx, and C. Fleury, “A Family of MMA Approximations for Structural Optimization”, *Structural and Multidisciplinary Optimization*, vol. 24, no. 4, pp. 263–276, 2002.
- [52] S. Zhou, L. Chen, and F. Sun, “Optimization of Constructal Volume-Point Conduction With Variable Cross Section Conducting Path”, *Energy Conversion and Management*, vol. 48, no. 1, pp. 106–111, 2007.

## A. Two-Dimensional Results

### A.1 Effect of $s$ and $s_0$ for Two Dimensions

This section contains the rest of the results for different values of  $s$  and  $s_0$ . As seen in Figure A.1 and Figure A.2, low values of  $s$  are very unstable in terms of  $\tau$ . In some cases, no optimum was found. Figure A.3 shows that  $s = 0.9$  gives stable results for  $\tau$  across all values of  $s_0$ . For  $s = 0.999$ , the results are stable in the lower region of  $s_0$  but diverges in the higher regions of  $s_0$  as shown in Figure A.4.

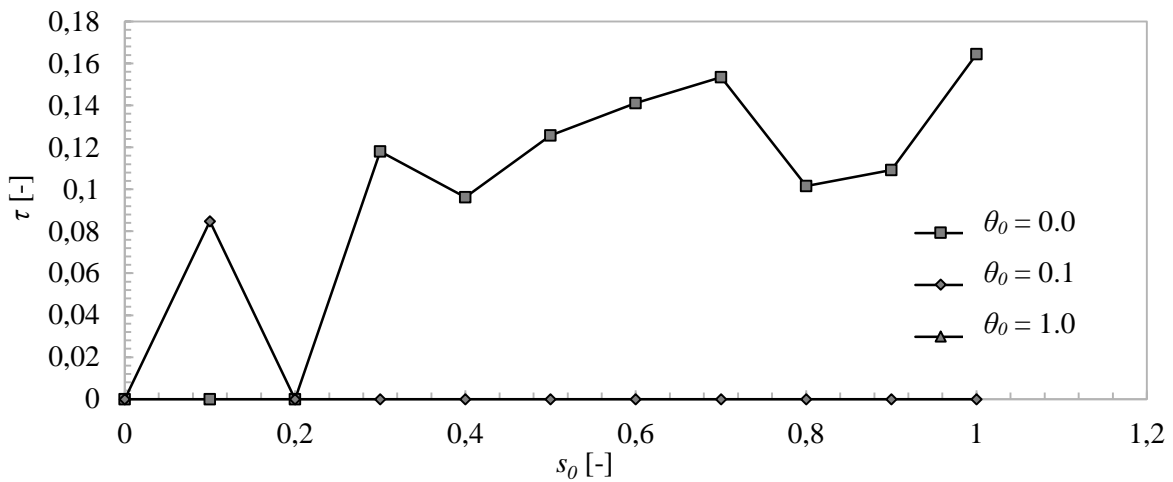


Figure A.1: The influence of  $s_0$  and  $\theta_0$  for  $s = 0.3$ .

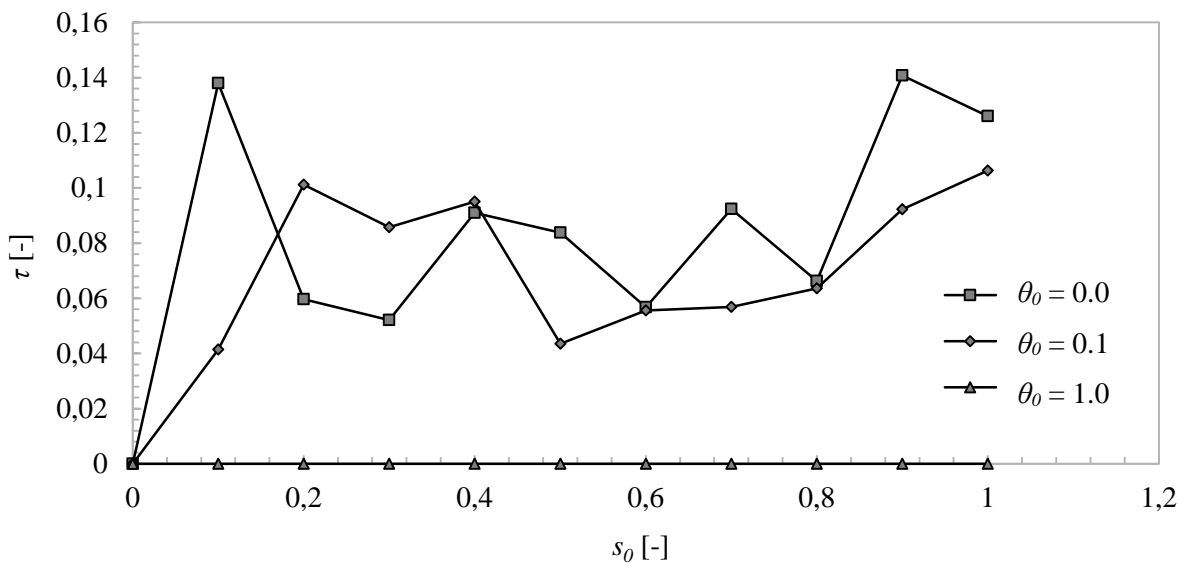
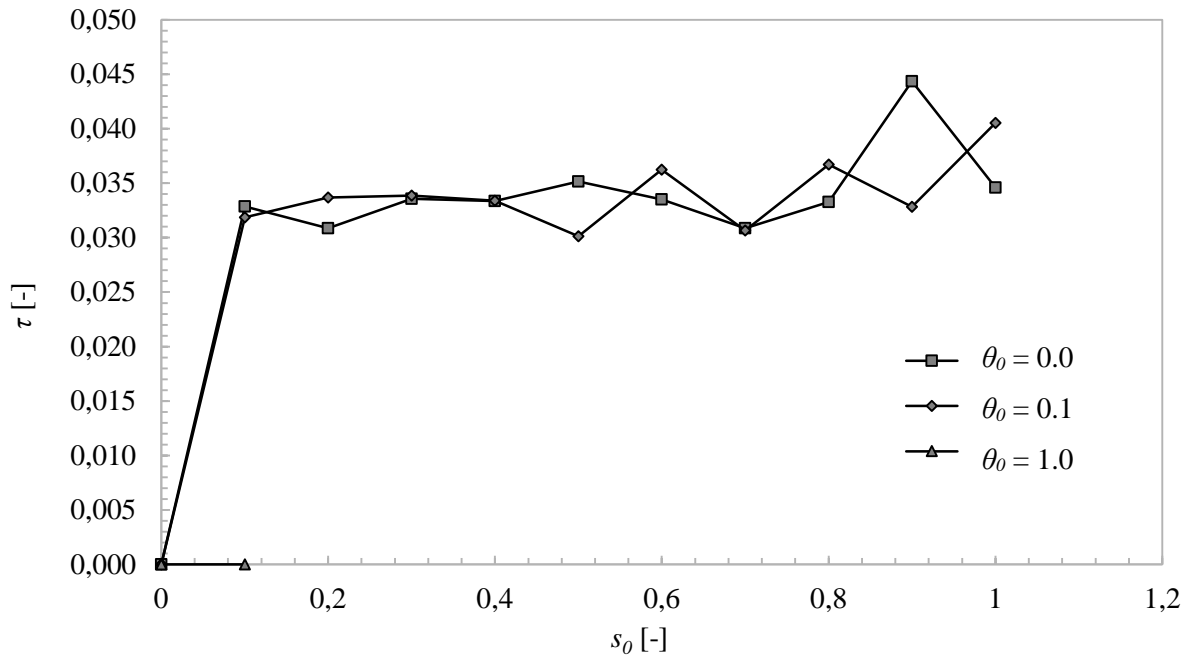
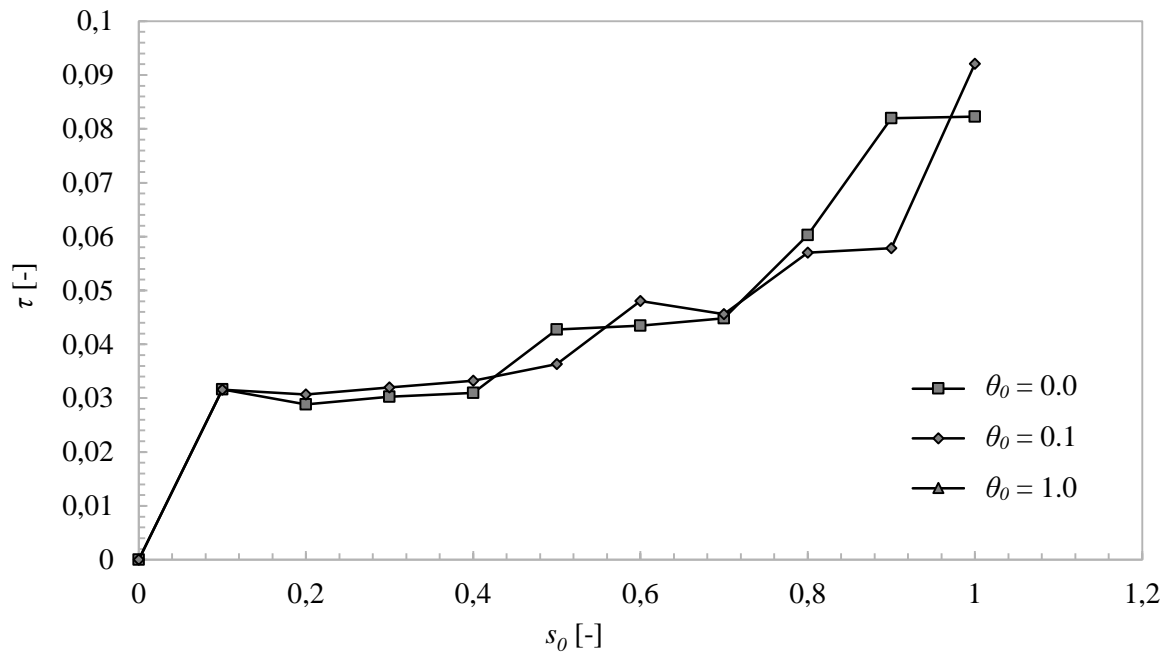


Figure A.2: The influence of  $s_0$  and  $\theta_0$  for  $s = 0.5$ .



**Figure A.3:** The influence of  $s_0$  and  $\theta_0$  for  $s = 0.9$ .



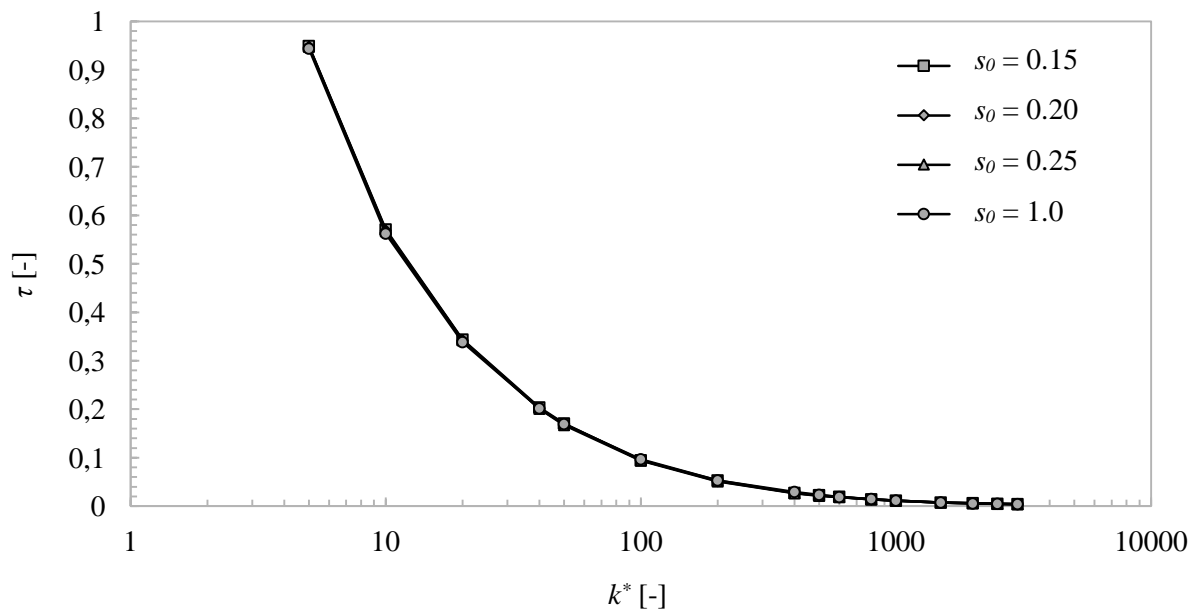
**Figure A.4:** The influence of  $s_0$  and  $\theta_0$  for  $s = 0.999$ .

## B. Three-Dimensional Results

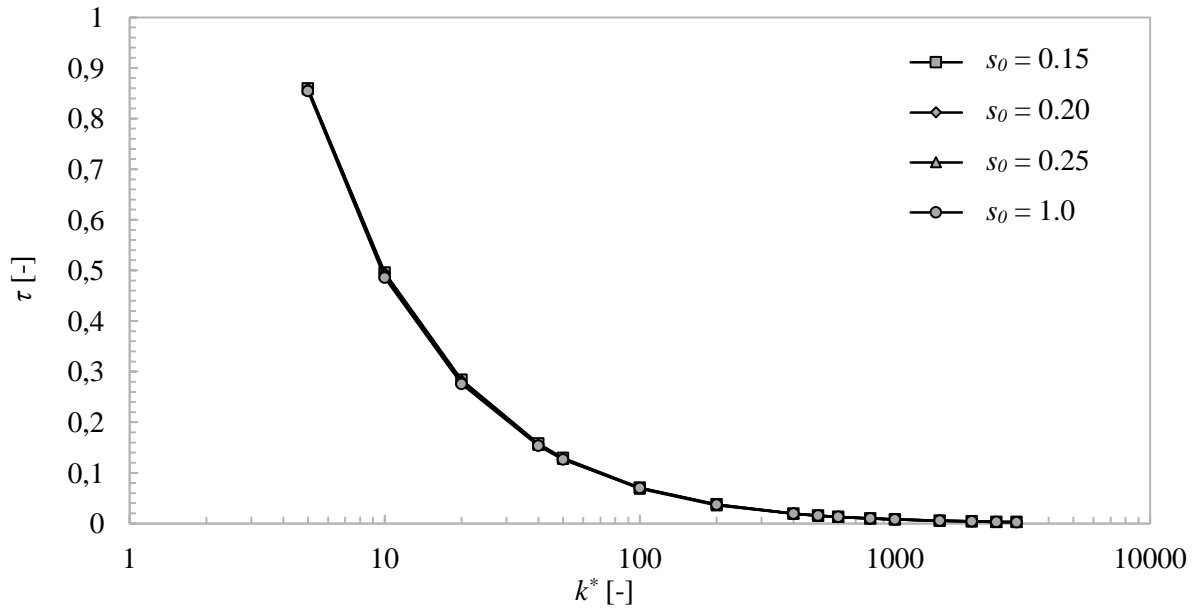
### B.1 Partial Dirichlet Boundary Results

#### B.1.1 Results for Different Conductivity Ratios and Volume Constraints

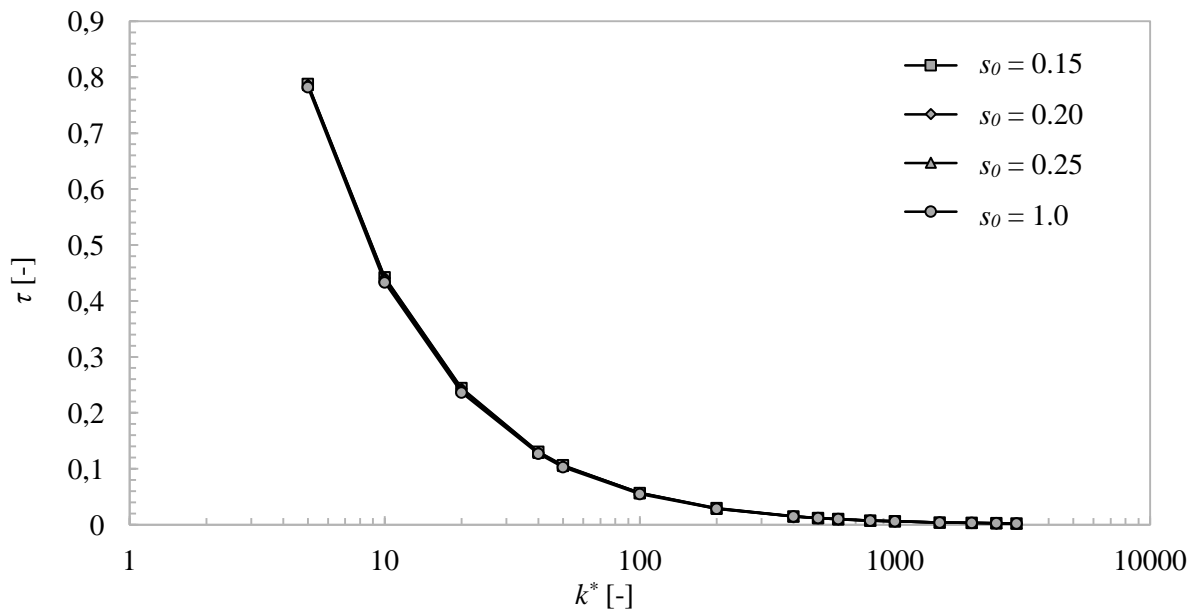
This subsection contains the rest of the results for a three-dimensional domain with a partial Dirichlet boundary, for different conductivity ratios and volume constraints. All results here follow the same pattern as discussed in Section 5.6. With an increase in the conductivity ratio, there is a decrease in the maximum temperature. This is also true for an increase in the volume constraint.



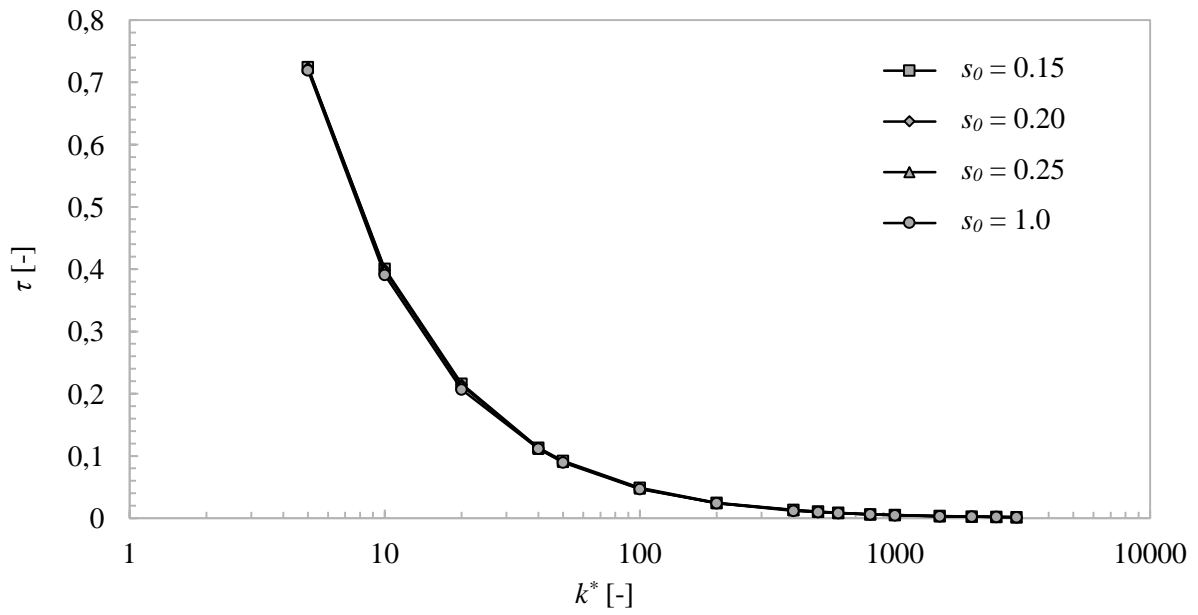
**Figure B.1:** Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.1$  for different values of  $k^*$ .



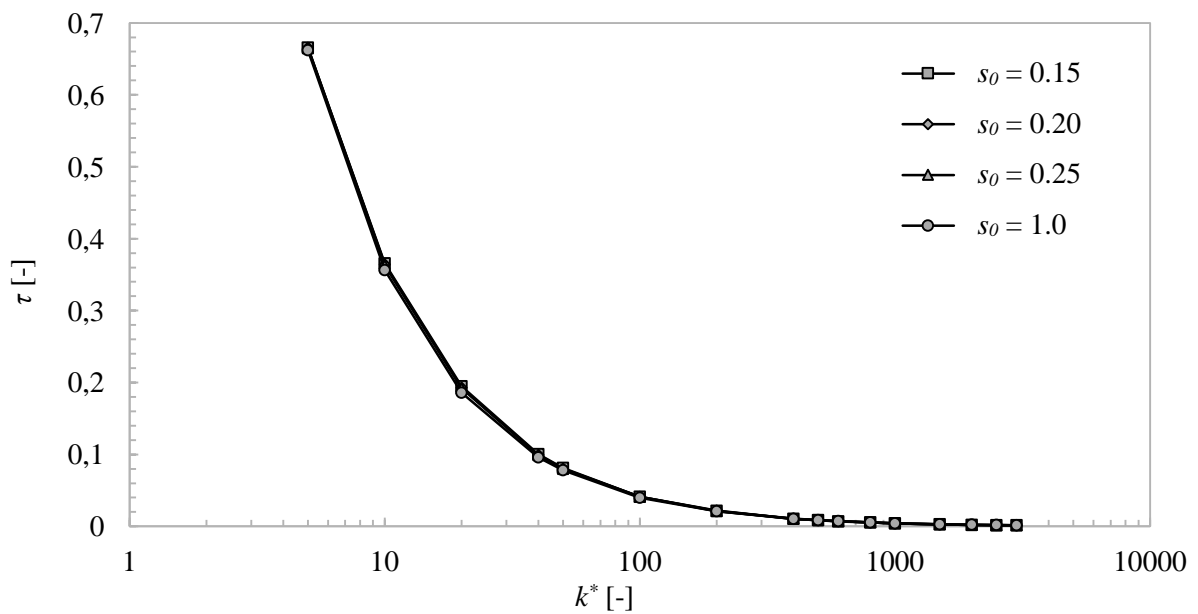
**Figure B.2:** Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.15$  for different values of  $k^*$ .



**Figure B.3:** Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.2$  for different values of  $k^*$ .



**Figure B.4:** Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.25$  for different values of  $k^*$ .

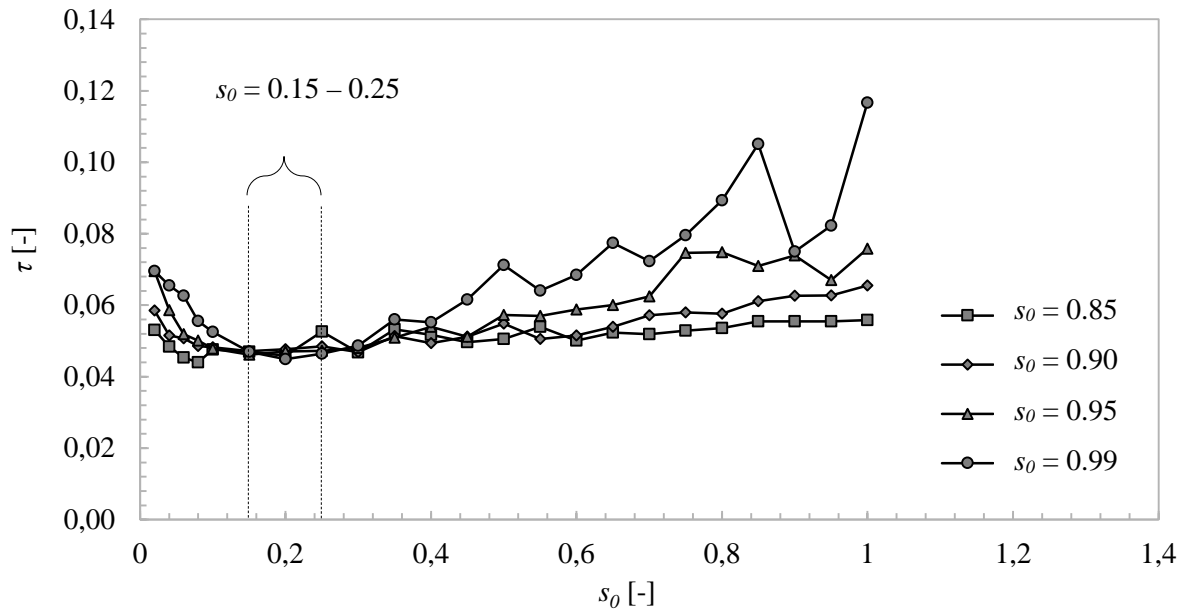


**Figure B.5:** Results for  $\tau$  for a partial Dirichlet boundary with  $V^* = 0.3$  for different values of  $k^*$ .

## B.2 Full Dirichlet Boundary Results

### B.2.1 Effect of the Asymptote Parameters, $s$ and $s_0$

















This subsection contains the results for the effect of the asymptote parameters,  $s$  and  $s_0$ . As seen in Figure B.6, there is an optimal range for  $s_0$  between 0.15 and 0.25. An  $s$  value of 0.85 is also the most stable for all values of  $s_0$ . Table B.1 shows some of the architectures obtained for varying values of  $s$  and  $s_0$ . There is again an increase in the amount of grey areas for low values of  $s_0$  and high values of  $s$ .



**Figure B.6:** The effect of the asymptote parameters on  $\tau$  for a three-dimensional domain with a full Dirichlet boundary.

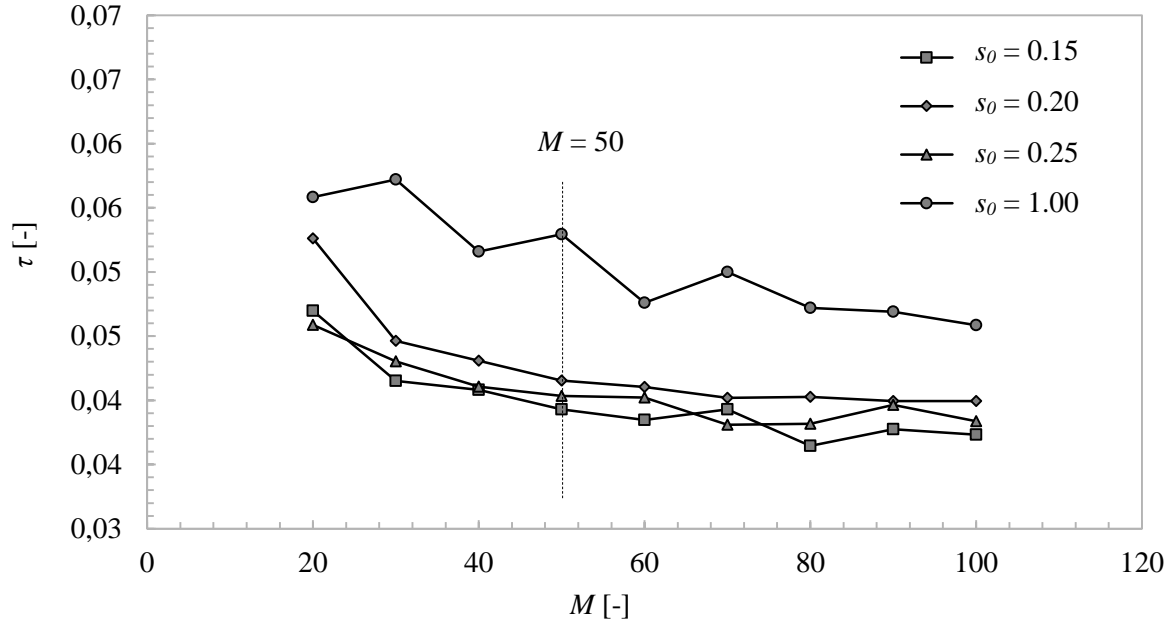


**Table B.1:** Architectures for different values of  $s$  and  $s_0$ .

$s_0$ [-]	0.85	0.85	0.85	0.85
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$5.302 \times 10^{-2}$	$4.841 \times 10^{-2}$	$4.533 \times 10^{-2}$	$4.401 \times 10^{-2}$
Density distribution				
$s$ [-]	0.9	0.9	0.9	0.9
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$5.854 \times 10^{-2}$	$5.157 \times 10^{-2}$	$5.071 \times 10^{-2}$	$4.861 \times 10^{-2}$
Density distribution				
$s$ [-]	0.95	0.95	0.95	0.95
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$6.950 \times 10^{-2}$	$5.858 \times 10^{-2}$	$5.190 \times 10^{-2}$	$5.014 \times 10^{-2}$
Density distribution				
$s$ [-]	0.99	0.99	0.99	0.99
$s_0$ [-]	0.02	0.15	0.5	1.0
$\tau$ [-]	$6.953 \times 10^{-2}$	$6.547 \times 10^{-2}$	$6.264 \times 10^{-2}$	$5.557 \times 10^{-2}$
Density distribution				

### B.2.2 Mesh-Dependence Study

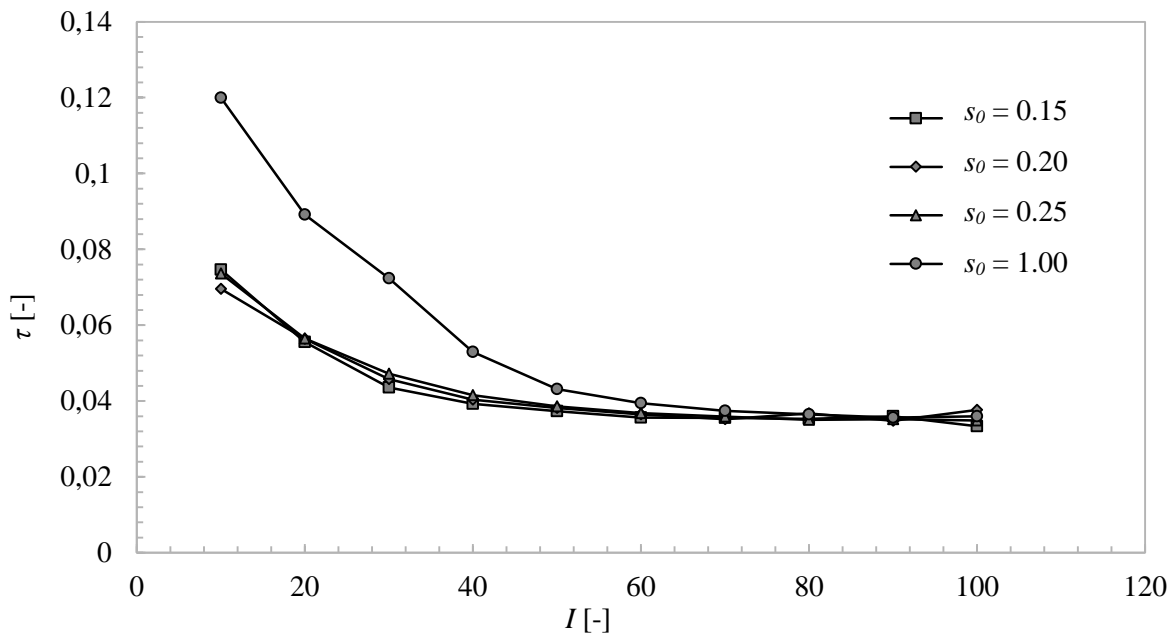
The mesh-dependence study for the full Dirichlet boundary condition is shown in Figure B.7.  $M = 50$  is again sufficient for the convergence of  $\tau$ .



**Figure B.7:** The mesh dependence for a three-dimensional domain using a full Dirichlet boundary with respect to  $\tau$ .

### B.2.3 Iteration-Dependence Study

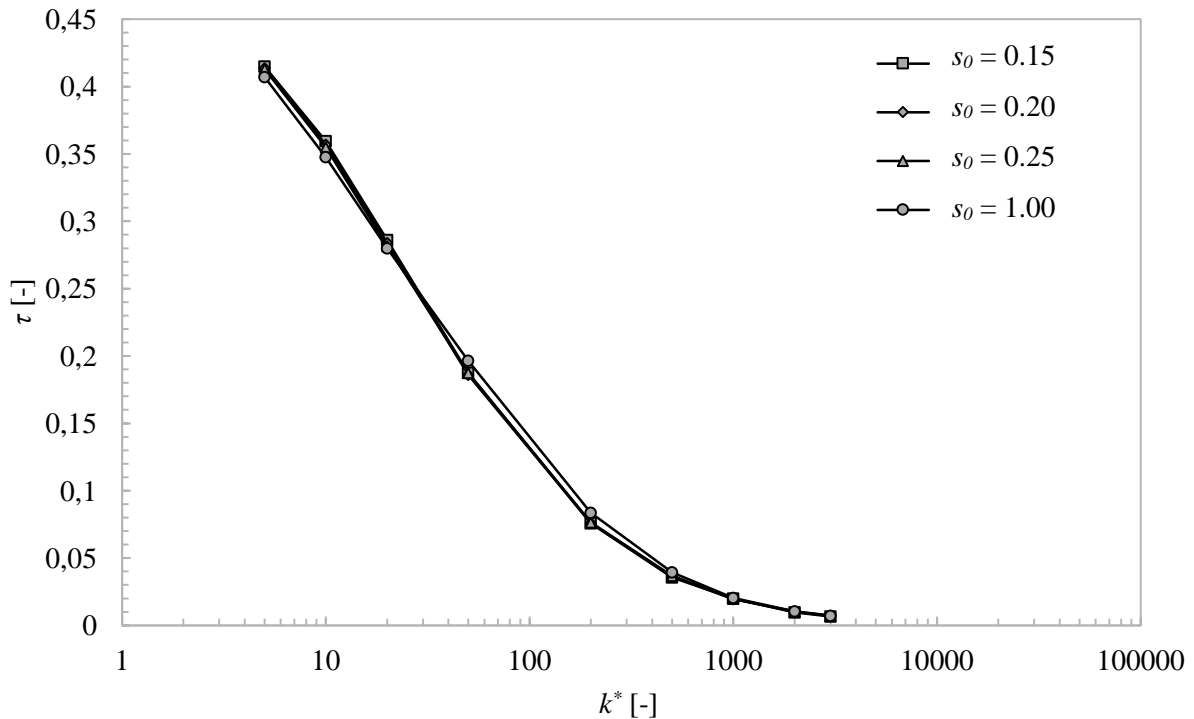
The dependence of  $\tau$  on the number of iterations is demonstrated in Figure B.8 .



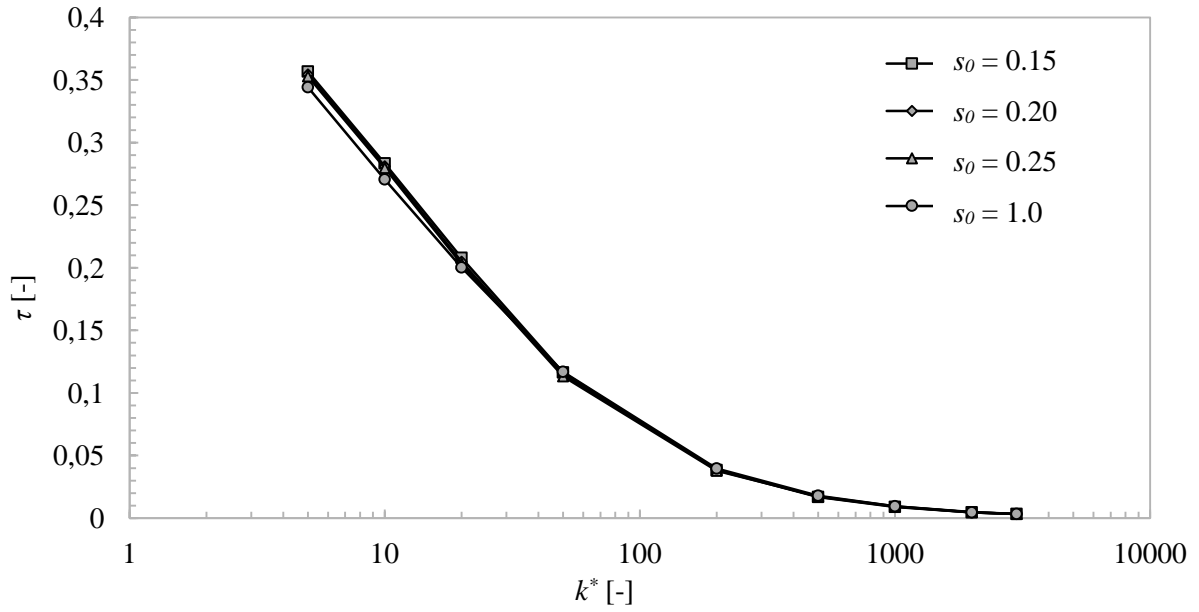
**Figure B.8:** The iteration study for a three-dimensional full Dirichlet boundary with respect to  $\tau$ .

### B.2.4 Results for Different Conductivity Ratios and Volume Constraints

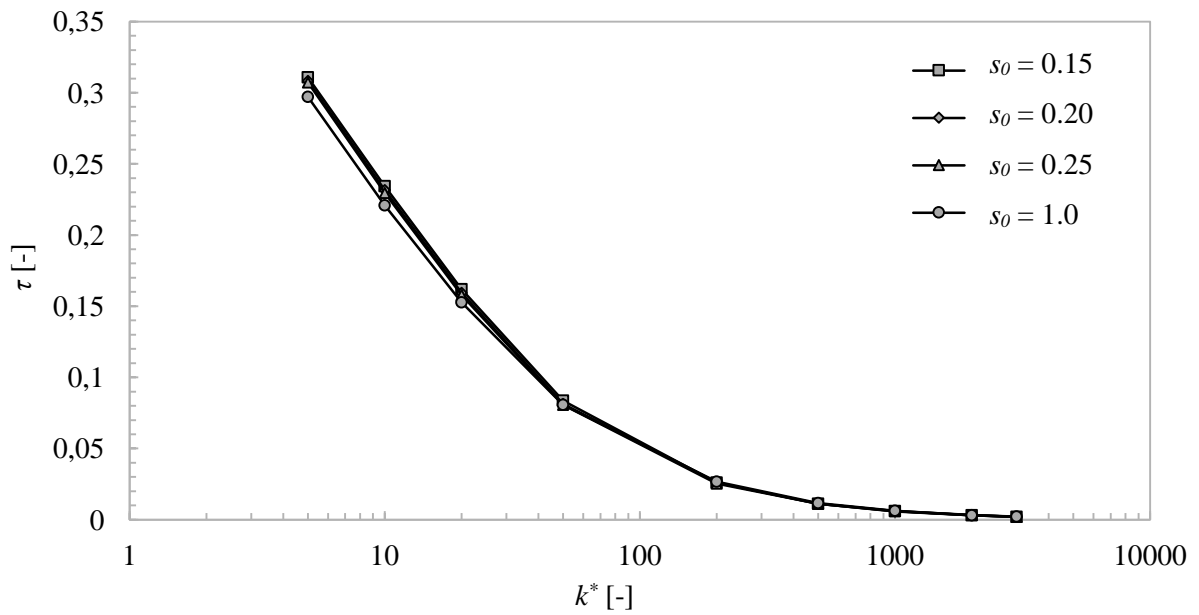
This subsection contains the rest of the results for a three-dimensional domain with a full Dirichlet boundary using one initial base, for different conductivity ratios and volume constraints. All results here follow the same pattern as discussed in Section 6.4. With an increase in the conductivity ratio, there is a decrease in the maximum temperature. This is also true for an increase in the volume constraint.



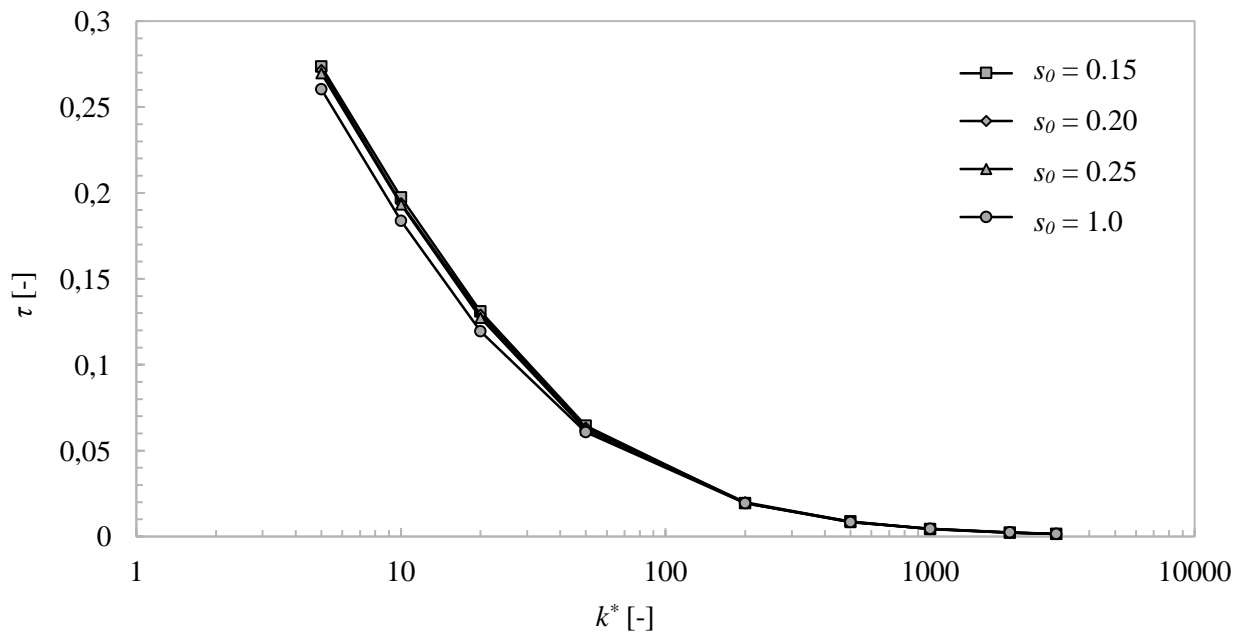
**Figure B.9:** The effect of  $k^*$  on the maximum temperature for a  $V^* = 0.05$  for a three-dimensional domain with a full Dirichlet boundary.



**Figure B.10:** Results for  $\tau$  for a full Dirichlet boundary with  $V^* = 0.1$  for different values of  $k^*$ .



**Figure B.11:** Results for  $\tau$  for a full Dirichlet boundary with  $V^* = 0.15$  for different values of  $k^*$ .

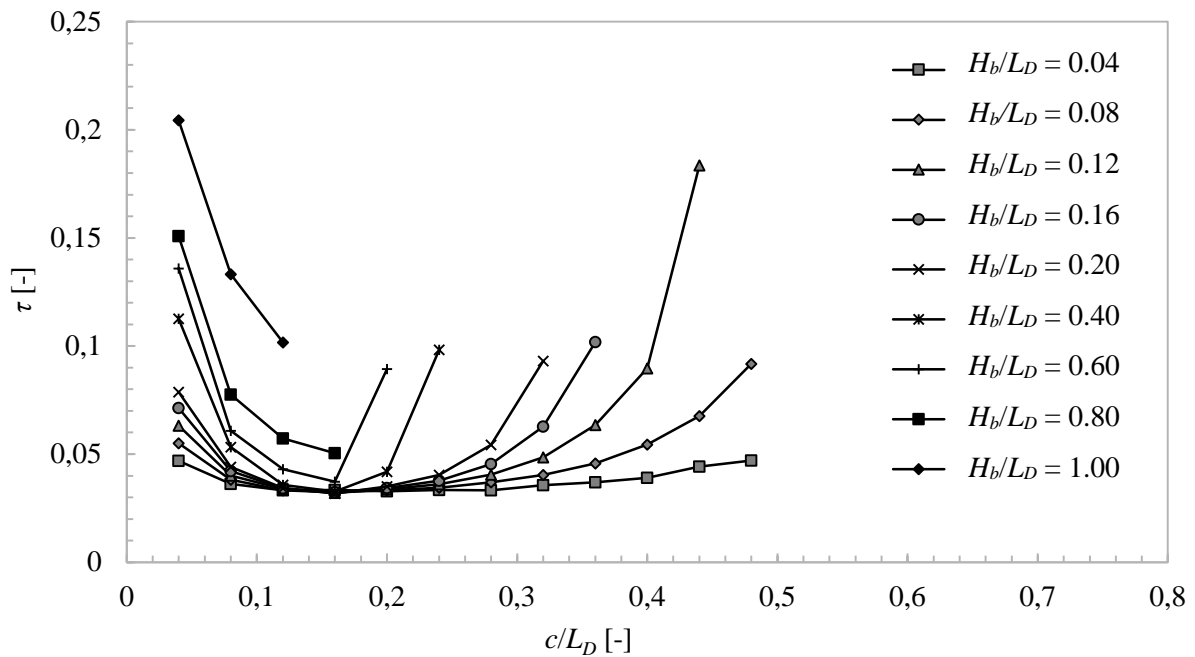


**Figure B.12:** Results for  $\tau$  for a full Dirichlet boundary with  $V^* = 0.2$  for different values of  $k^*$ .

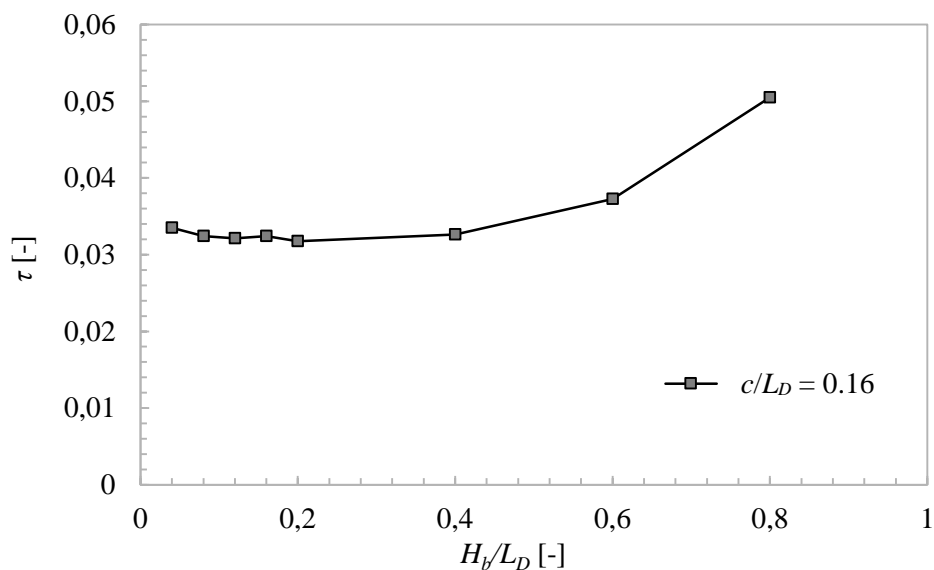
## B.3 Full Dirichlet Boundary Results for Two Bases

### B.3.1 Size of the Base Structure

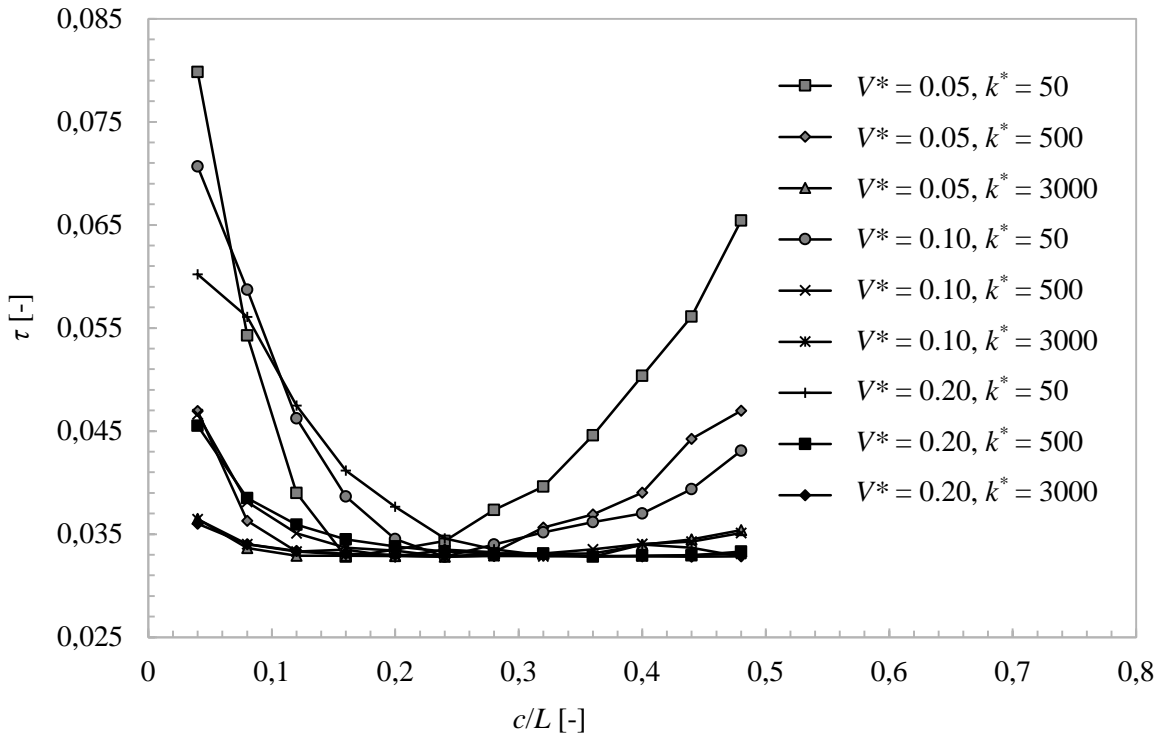
Figure B.13 shows the results for different values of  $H_b/L_D$  and  $c/L_D$ . As shown in the figure, there is an optimal point for  $c/L_D$  around 0.16. Figure B.14 shows that any value of  $H_b/L_D$  below 0.4 is optimal. After that point,  $\tau$  increase again. A  $H_b/L_D$  value of 0.04 was used to give the MMA as uch freedom as possible. Figure B.15 shows the results for different conductivity ratios and volume constraint for  $H_b/L_D = 0.04$  and different values of  $c/L_D$ . The optimal value of  $c/L_D$  for all conductivity ratios and volume constraints is 0.24.



**Figure B.13:** Results for  $\tau$  showing the effect of the size of the bases for a three-dimensional domain with a full Dirichlet boundary.



**Figure B.14:** The effect of  $H_b/L_D$  on  $\tau$  for  $c/L_D = 0.16$ .

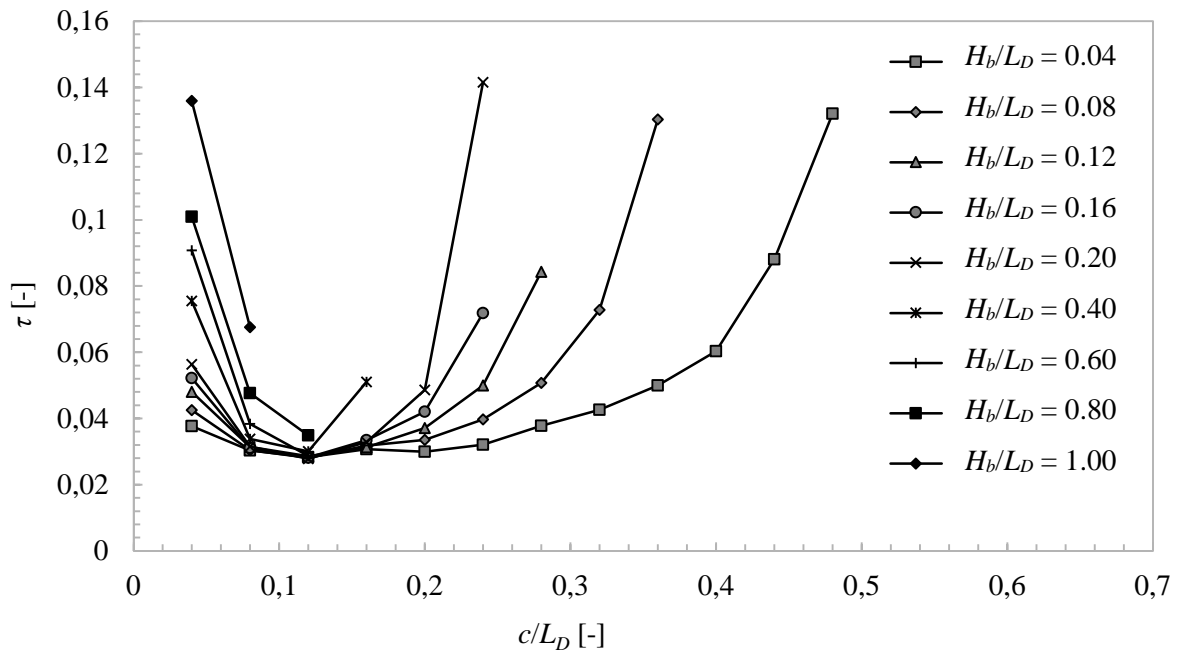


**Figure B.15:** Results for different conductivity ratios and volume constraints for different values of  $c/L_D$  and  $H_b/L_D = 0.04$  for a three-dimensional domain with a full Dirichlet boundary and two initial bases.

## B.4 Full Dirichlet Boundary Results for Four Bases

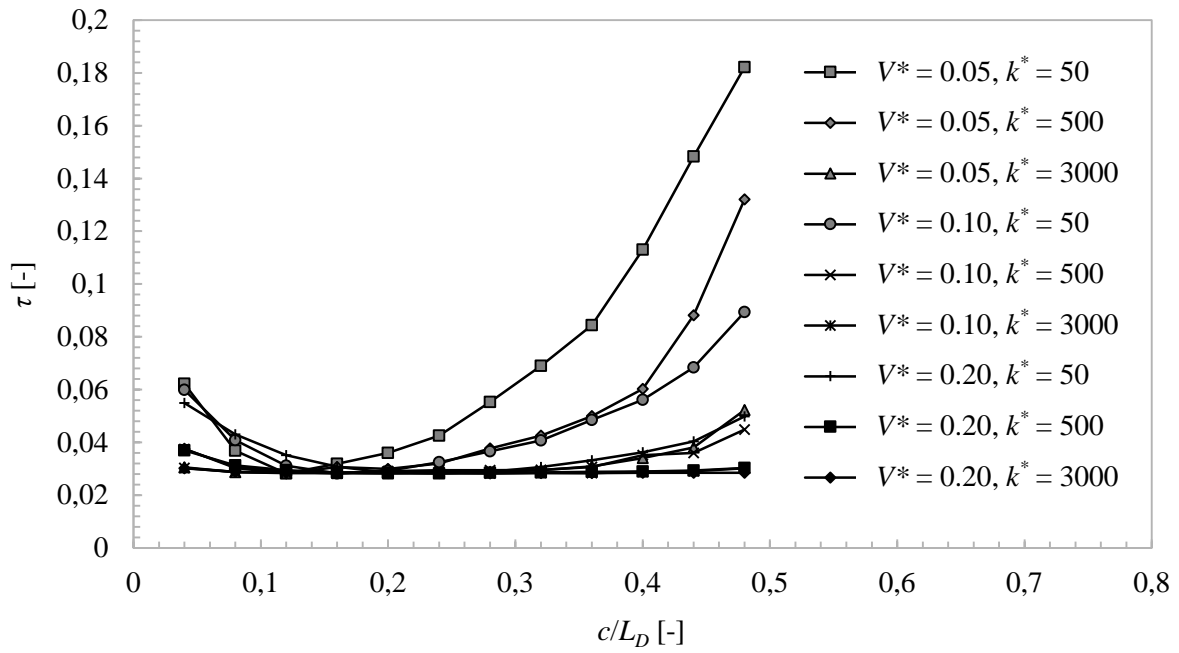
### B.4.1 Size of the Base Structures

Figure B.16 shows the results for different values of  $c/L_D$  and  $H_b/L_D$  using four initial bases. The optimal value of  $c/L_D$  is clearly seen at 0.12. All values of  $H_b/L_D$  below 0.6 were optimal, but it was decided to use  $H_b/L_D = 0.04$  to give the MMA as much freedom as possible. Figure B.17 shows the results for different conductivity ratios and volume constraint for  $H_b/L_D = 0.04$  and different values of  $c/L_D$ . The optimal value of  $c/L_D$  found in Figure B.16 was shown to be optimal for all conductivity ratios and volume constraints.



**Figure B.16:** The effect of the base size on  $\tau$  for a three-dimensional domain with a full Dirichlet boundary with four initial bases.





**Figure B.17:** Results for different conductivity ratios and volume constraints for different values of  $c/L_D$  and  $H_b/L_D = 0.04$  for a three-dimensional domain with a full Dirichlet boundary and four initial bases.