Title

Simulation and Optimization of Fused Deposition Modeling

Author

Wang, Jingyi

Publication Date

2020

Peer reviewed|Thesis/dissertation

Simulation and Optimization of Fused Deposition Modeling

by

Jingyi Wang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering- Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Panayiotis Papadopoulos, Chair
Professor Per-Olof Persson
Professor Robert L. Taylor

Spring 2020

Simulation and Optimization of Fused Deposition Modeling

Abstract

Simulation and Optimization of Fused Deposition Modeling

by

Jingyi Wang

Doctor of Philosophy in Engineering- Mechanical Engineering

University of California, Berkeley

Professor Panayiotis Papadopoulos, Chair

Since their invention decades ago, various kinds of additive manufacturing (AM) methods have emerged. In recent years additive manufacturing has been growing very rapidly as the technology matures and the cost of production declines. One of the first AM methods, fused deposition modeling (FDM) has evolved with more capable printers and novel materials making significant progress towards industrialization of the process. However, as it stands now, a lack of understanding of the fundamental physics during the additive manufacturing process leads to some severe drawbacks. For example, additively manufactured parts can suffer from low finish quality and prohibitively long production time.

Numerical simulation has been very helpful in understanding and designing traditional manufacturing processes, such as metal forming and welding. It has huge potential in AM as well. Gaining key knowledge pertaining to the printing process, such as the temperature field change, will help build better printers and enable designers to achieve design goals such as minimum residual stress or surface roughness. As a result, research interest in reliable, robust and efficient numerical models for additive manufacturing has surged in the last decade.

The objective of this dissertation is to establish a coupled thermomechanical finite element model that accurately simulates the FDM process and computes quantities of interest, such as the shape error. The temperature and deformation history is taken into consideration when depositing new material. The results of two-dimensional example problems are discussed and compared with those from an uncoupled finite element model. In addition, a parameterized PDE-constrained (partial differential equation) optimization study based on the simulation model is performed. Algorithms designed with knowledge gained from the simulation are proposed to find optimal solutions.

Contents

List of Figures

List of Tables

Acknowledgments

I want to thank my committee, Prof. Papadopoulos, Prof. Taylor and Prof. Persson, without whose support and help I wouldn't be able to finish this dissertation. I am very grateful to have Prof. Papadopoulos as my advisor who has made this journey a lot easier and more enjoyable. I also have had tremendous help from the Mathematics Group at the Berkeley Lab where I have learned so much and made friends. Last, I want to thank my family for being the best cheerleaders I can ask for.

Chapter 1

Introduction

Additive manufacturing (AM) builds objects by adding new material layer upon layer as opposed to the traditional way of removing materials from the original workpiece. This allows complicated topology to be printed in a much simpler way and shortens the design cycle time at low cost by removing the need for expensive tooling [1, 2, 3, 4, 5, 6]. A number of additive manufacturing methods have been developed in recent years, including fused deposition modeling (FDM), selective laser sintering, *etc.* Faster, more powerful, and integrated machines have been coming up online with more resources devoted to industry. AM has been adopted widely in prototyping and is steadily making its way into medical, aerospace, automotive and other industries [7, 8]. Due to its huge potential, AM has garnered interest both in industry and academia. On the national level, America Makes, established in 2012 and based in Youngstown, Ohio, which is the flagship Institute for Manufacturing USA (the National Network for Manufacturing Innovation), is coordinating efforts across the AM industry to promote the advancement and adoption of AM. National laboratories, including Lawrence Livermore National Laboratory, Sandia National Laboratories and Los Alamos National Laboratory, are conducting research on topics of additive manufacturing process for metals [9, 10].

On the simulation software side, efforts have been made to develop CAE tools for additive manufacturing. Software companies such as Autodesk and Ansys, provide continuum-scale process simulation and post-processing analysis to help design the objects and the process itself. However, such success is still limited and in its early stage as the AM process involves multi-scale physical phenomena, such as rapidly evolving microstructures and deformation caused by large and changing temperature gradients. Research is underway to connect models of different length scales by taking advantage of the latest supercomputers as computing resources [9, 10, 11, 12, 13]. To that end, more progress needs to be made before AM models become mainstream tools in industry.

In academia, a lot of research has focused on AM processes for metals, *e.g.*, powder bed fusion, where metal powders are melted and formed into shape with a high power laser as the heat source. Groups at Lawrence Livermore National Laboratory have studied multiscale models [10], one on the powder scale and the other on the continuous or medium

scale to optimize the additive manufacturing process. More specifically, the continuum scale model uses the finite element method to computationally build a complete part and predict properties, such as residual stress. Phase change is incorporated into the heat transfer calculations along with a sophisticated laser model. On the other hand, the powder scale model focuses on the melt pool and microstructure simulation of the metal to reduce defects in the printed object.

Most simulation methods for AM solve the transient heat transfer problem first and compute the mechanical part based on the obtained temperature history subsequently, thus leading to a weakly coupled model. Michaleris and coworkers [14, 15, 16] explored heat transfer simulations of 3D metal deposition problem using two ways for adding new elements, namely quiet and inactive element activation which are explained in Section 2.3.1. While the quiet element method is easier to implement and faster overall, inactive element activation offers better accuracy and stability. Conduction, convection and radiation are all considered in these works. The temperature history generated through these thermal analyses is used for mechanical calculations, such as residual stress considering thermal-induced strain. It is found that the inactive element method gives more accurate results. Kolossov *et al.* [17] modeled the heat transfer and thermal field of selective laser sintering using a nonlinear 3D finite element model, which took into consideration the temperature dependence of thermal conductivity and heat capacity. The results were compared with experimental data obtained through infrared camera and matched well. Roberts *et al.* [18] used element 'birth and death', equivalent to the inactive element activation mentioned above, to simulate the three-dimensional temperature field during the laser melting process. Again the model was implemented using the finite element method. The results showed temperature peaks occurring around newly-added elements. Zhang and Chou [19] developed a 3D finite element model to simulate FDM and computed the residual stresses in order to evaluate part distortion. The model used the inactive element method and calculated mechanical quantities at each time step after the temperature field was determined. Prototype parts were built and used to validate the simulation results.

Keller *et al.* [20] simulated the transient temperature field during a laser powder bed fusion manufacturing process. It was used to study the microstructure formulation and its thermodynamic driving forces. The temperature history obtained was then applied to solidification simulations and compared to experimental results observed from electron and X-ray diffraction analyses. Yang *et al.* [21] used the quiet element method to simulate directed energy deposition of Ti-6Al-4V with a new criterion of element activation. Both pointwise and global thermal and mechanical quantities were measured and compared. The simulation results compared well with experiments using laser scanning.

Costa *et al.* [22] developed a thermo-kinetic finite element model of multilayer laser powder deposition for the investigation of the microstructural transformations and hardness variations that occur during the deposition of steel parts. The model is capable of computing the final hardness distribution in the part. The temperature history was used for kinetic phase transformation calculations. Hu *et al.* [23] developed a 3D finite element model using a commercial code to study the molten pool thermal behavior in building a single-bead

wall with laser-based additive manufacturing which had a closed-loop control component to ensure a constant width of the molten pool. The results compared well with experiments conducted through an infrared imaging system. Moreover, the traverse velocity was found to have limited impact on the average temperature in the processing zone. Ji *et al.* [24] treated latent heat with enthalpy and calculated temperature distribution with a 3D commercial finite element model by solving the heat equation with convection and radiation. Zhang *et al.* [25] developed a finite element model in a commercial code using the quiet element method and thermomechanical calculations similar to the ones to be described in Section 3.5. Heat source was included and calculated as the enthalpy change of acrylonitrile butadiene styrene (ABS). Part distortion and residual stress were evident and found to be dependent on toolpath.

Another key concern in AM simulation is the sheer volume of computing power required for the simulation of complex geometries especially in three dimensions. A common technique is to use adaptive mesh refinement to reduce the number of elements needed far away from the heat source resulting in a coarser mesh for the already printed layers and elements. Other researchers are studying parallelization in time to reduce computing cost [10].

Current optimization research of additive manufacturing has mostly been focused on data-driven models where enough experiments with varying parameters can predict the outcome of a different parameter combination [1, 26]. Some of the techniques used include the Taguchi method, response surface methodology (RSM) and artificial neural network (ANN). Due to its flexibility and efficiency in studying large numbers of parameter combinations, finite element method is also widely adopted.

West *et al.* [27] proposed a process-planning method for aiding Stereolithography to achieve a balanced objective among geometric tolerances, surface finishes, and part build time. It relied on empirical data and included process variables such as part orientation and layer thickness. An adaptive slicing algorithm was proposed with the method. Vastola *et al.* [28] used a finite element model to investigate the effect of beam size, beam power density, beam scan speed and chamber bed temperature on residual stress. The simulation model was similar to the uncoupled thermomechanical ones to be introduced in Section 3.5 with a fixed Lagrangian mesh. Cho *et al.* [29] used a multilayer perception neural network to select the optimal parameters for Stereolithography, a type of additive manufacturing technology. Gockel *et al.* [30] developed process mapping based on finite element method of a solidification microstructure process for deposition of single beads of Ti-6Al-4V via electron beam wire-feed AM processes. The mapping was to optimize process variable combinations to generate constant beta grain size and morphology. Strano *et al.* [31] investigated the surface roughness and morphology for Steel 316L alloy parts made by Selective Laser Melting (SLM). The real surface roughness at different sloping angle was investigated and a mathematical model was proposed to efficiently predict its value. The model took into consideration the increasing presence of particles on the top surface in addition to the staircase effect. Nickel *et al.* [32] used uncoupled thermal and mechanical finite element analysis to examine the deposition pattern effect on the deflection of manufactured parts by shape deposition manufacturing. The deposition pattern included raster, spiral from inside to outside

and spiral from outside to inside. A strong correlation between the patterns and deflections was demonstrated. Krol *et al.* [33] investigated support structure for metal-based AM using finite element simulations. The heat transfer problem was solved and subsequently residual stress was computed. It was shown that a fractal adaptation of the support layout and an optimization of block supports could be done efficiently to achieve comprehensive objectives, *e.g.*, lowering residual stress to avoid failure during manufacturing. By choosing the finite element method, a large number of results can be calculated and compared easily. A process map was created by Vasinonta *et al.* [34] from numerical models of laser-based material deposition of thin-walled structures. The model used finite elements to solve the heat transfer problem with temperature-dependent conductivity and specific heat. The effects of wall height, laser power, deposition speed and part preheating on melt pool length were investigated. The map was applied to the Laser Engineered Net Shaping (LENS) process at Sandia National Laboratories.

Using the Taguchi technique Anitha *et al.* [35] conducted experiments and concluded that layer thickness had the most effect on surface roughness. More layers produce better surface roughness, while road width and printing speed also play a role. Slower printing speed in general also helped improve surface quality. Similar conclusions are drawn in other literature as well [36, 37, 38, 39, 40]. Part orientation, road width, layer thickness, air gap and raster angle were optimized for dimensional accuracy using the Taguchi method in [41, 36]. Here, shrinkage along the length, width and diameter of the hole of the fabricated part was observed compared to the designed value. It was concluded that decreasing layer thickness helps to reduce dimensional error.

Onwubolu *et al.* [42] developed a functional relationship between process parameters and tensile strength for FDM process. The tensile strength data were gathered through experiments. Then, using the group method of data handling (GMDH), five process parameters were considered layer thickness, orientation, raster angle, raster width, and air gap. It was found that smaller layer thickness would improve tensile strength while being the more costly in time. Sood *et al.* [43] similarly investigated the effect of these five processing parameters on the compressive strength of FDM-built parts. Quantum-behaved particle swarm optimization and artificial neural network were used to predict compressive strength for other parameters. Thrimurthulu *et al.* [39] studied the optimum part orientation to improve part surface finish while maintain a reasonable build time. Adaptive slicing is applied in the optimization process.

Chacón *et al.* [44] characterized the effect of build orientation, layer thickness and feed rate on the mechanical performance of FDM-produced polylactic acid (PLA) samples. Tensile and three-point bending tests were carried out to find the mechanical response. The mechanical properties strengthened with layer thickness and against the feed rate for the upright orientation. For on-edge and flat orientation however, they were found to be of small significance. Domingo-Espin *et al* [45] studied building orientation, as well as other parameters, to determine how they affect on the mechanical response by testing FDM-made specimen. Lee *et al.* [46] used the Taguchi method to investigate the effect of process parameters on the elastic performance of the part through experiments. Similar studies

with different parameters and methods focusing on mechanical properties can be found in [8, 47, 48, 49, 50, 51]. Dynamic mechanical properties are also dependent on the parameters, as studied in [52, 53].

Alafaghani *et al.* [54] examined the influence of FDM processing parameters on the final parts characteristics including mechanical properties and dimensional accuracy. Parameters studied were building directions, infill percentage, infill patterns, print speed, extrusion temperatures and layer height. Finite element model for the test specimen that took into account porosity of the FDM printed structure was also proposed. Zhang *et al.* used the finite element model described earlier in [25] to study parameters effect on part distortions. The results showed that part distortions were related to the stress accumulation during the deposition. Three parameters were considered, layer thickness, scan speed and road width. In general distortions increased with both road width and layer thickness and decreased with speed [55].

Other optimization efforts include topology optimization for FDM since much more complicated small-scale features can be manufactured using additive manufacturing technologies [56, 57] while the manufacturing process demands novel constraints on the optimization.

As already discussed, most of the research on the continuum scale modeling of additive manufacturing employs decoupled or weakly coupled thermomechanical calculations and predefined element addition paths. While these models lead to faster simulations, they might not work for applications that demand high accuracy in predicting quantities such as residual stress and shape error. Furthermore, to the best of our knowledge, the optimization efforts of the AM process rely on experimental data and statistical analysis thus limiting the input directly from simulation tools. In this work, a novel continuum-scale coupled thermomechanical finite element model is proposed for the fused deposition modeling manufacturing process in two dimensions, which can provide more comprehensive understanding of the coupling effect of temperature and deformation. The model also takes into consideration the deformation which occurs when adding new elements during the printing process to more accurately represent the overall process. Two optimization methods are explored based on the simulation including both discrete and continuous parameters. The first one is a gradient-less local search method called the method of local variants. It is straightforward and guaranteed to converge to a local minimum. The speed of convergence depends on the step size in the parameter space. The second method is based on the parameterization of the finite element model for continuous parameters. We use gradients computed throughout the simulation to find the optimization path and propose a gradient descent with line search algorithm. Unlike previous empirical optimization approaches, these simulation-based algorithms give insight into how the parameters through physical phenomenon affect the objectives of interest. In addition to rectangular geometries in two-dimension, we discuss ways to better simulate and optimize the approximation of curvatures as well. Detailed discussion on a parameter, the number of layers, is presented since it plays an important role in designing the optimization process for the curved geometry cases.

Chapter 2

Finite Element Model for Fused Deposition Modeling

2.1 Introduction

Finite element modeling has proven to be extremely powerful in simulating manufacturing and other engineering applications and is the dominant method in the technical literature for additive manufacturing simulation [58]. The fully-coupled continuum level thermomechanical problem solves simultaneously the linear momentum balance, angular momentum balance and the energy balance [59] for the displacement and temperature field. The coupling between thermal and mechanical effects can happen on several levels. For example, the stress constitutive law may depend both on strain and temperature. Thermal-induced strain is another important coupling factor. Also the heat source in FDM, unlike the laser melting process, is the hot material itself. The two-dimensional finite element model introduced in this section is based on the theory of continuum mechanics with coupled thermomechanical equations and incorporates dynamic element creation to simulate the fused deposition modeling process [60]. In order to build parts with curvature, slicing algorithms and partial elements are introduced. The simulation model developed in this chapter is the basis of the optimization study in Chapter 4.

2.2 Simulation Model

2.2.1 Balance Laws and Constitutive Assumptions

The kinematics relations used throughout the dissertation are introduced first. As shown in Figure 2.1, let $\boldsymbol{X}$ be the material point positions of the reference configuration, denoted by V. A motion $\bar{\boldsymbol{x}}$ assigned to each material point $\boldsymbol{X}$ and time t is such that

$$\boldsymbol{x} = \bar{\boldsymbol{x}}\left(\boldsymbol{X}, t\right), \tag{2.1}$$

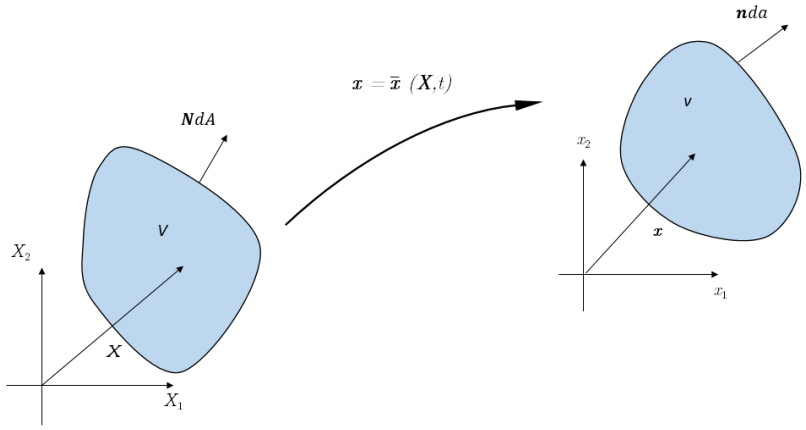


Figure 2.1: Kinematic mapping between reference and current configurations

where $\boldsymbol{x}$ is the position of the same material point at time t. The motion considered as a function of $\boldsymbol{X}$ at a given time t is the deformation and v is called the deformed configuration. A basic hypothesis of continuum mechanics is that the motion $\bar{\boldsymbol{x}}$ is one-to-one and onto for a fixed t. The deformation gradient, denoted by $\boldsymbol{F}$, is defined as

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \mathbf{1} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}}, \tag{2.2}$$

where $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ is the displacement and $\mathbf{1}$ the mixed identity tensor. A sufficient condition for the invertibility is

$$J = \det \boldsymbol{F} > 0, \tag{2.3}$$

where J denotes the determinant of the deformation gradient $\boldsymbol{F}$. To describe the deformation and strain, define by $\boldsymbol{C}$ the right Cauchy–Green deformation tensor and $\boldsymbol{E}$ the Lagrangian strain tensor as

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{F}^T \boldsymbol{F}, \\ \boldsymbol{E} &= \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}), \end{aligned} \tag{2.4}$$

where $\boldsymbol{I}$ is the identity tensor in the reference configuration. An important and unique decomposition of the deformation gradient is the right polar decomposition

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}, \tag{2.5}$$

where $\boldsymbol{R}$ is a rotation which can defined by $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$, $\det(\boldsymbol{R}) = 1$, and $\boldsymbol{U}$ is the right stretch tensor, which is positive-definite and symmetric. Details on how to obtain the decomposition can be found in [61].

The linear and angular momentum balance equations over the reference configuration are written as

$$\begin{aligned} \rho_0 \boldsymbol{a} &= \nabla \cdot \boldsymbol{P} + \rho_0 \boldsymbol{b}, \\ \boldsymbol{P}\boldsymbol{F}^T &= \boldsymbol{F}\boldsymbol{P}^T, \end{aligned} \tag{2.6}$$

where $\boldsymbol{P}$ is the first Piola–Kirchhoff stress tensor, ρ_0 the mass density per unit referential volume, $\boldsymbol{b}$ the body force per unit mass, $\boldsymbol{a}$ the acceleration, and $\nabla\cdot$ the divergence relative to the referential coordinates.

The energy balance equation takes the form

$$\rho_0 \dot{e} = \rho_0 r - \nabla\cdot\boldsymbol{q}_0 + \boldsymbol{P}\cdot\dot{\boldsymbol{F}}, \tag{2.7}$$

in terms of the internal energy per unit mass e, the heat supply per unit mass r, and the heat flux vector $\boldsymbol{q}_0$ over the geometry of the reference configuration.

The local referential form of the Clausius–Duhem inequality can be stated as

$$\dot{\psi} + \dot{\theta}\eta - \boldsymbol{P}\cdot\dot{\boldsymbol{F}} + \frac{\boldsymbol{q}_0\cdot\nabla\theta}{\theta} \leqslant 0. \tag{2.8}$$

Here η and

$$\psi = \rho_0 e - \theta\eta, \tag{2.9}$$

are the entropy and Helmholtz free energy per unit volume, respectively, ∇ denotes the gradient relative to the referential coordinates and θ is the temperature.

Assuming general thermalelasticity while guided by Equation (2.8) and the principle of equipresence [62], the free energy ψ, the first Piola–Kirchhoff stress $\boldsymbol{P}$, the entropy η and the heat flux $\boldsymbol{q}_0$ assume the constitutive forms

$$\begin{aligned}
\psi &= \bar{\psi}\left(\boldsymbol{F}, \theta, \nabla\theta\right), \\
\boldsymbol{P} &= \bar{\boldsymbol{P}}\left(\boldsymbol{F}, \theta, \nabla\theta\right), \\
\eta &= \bar{\eta}\left(\boldsymbol{F}, \theta, \nabla\theta\right), \\
\boldsymbol{q}_0 &= \bar{\boldsymbol{q}}_0\left(\boldsymbol{F}, \theta, \nabla\theta\right).
\end{aligned} \tag{2.10}$$

The frame-indifference principle is commonly used in continuum mechanics to guide the derivations of constitutive relations. It states that the physical laws should be independent of the new frame of reference. Consider a frame-rotation given by $\boldsymbol{Q}$, following [61, Section 57.1] the transformation law for $\boldsymbol{F}$ and $\boldsymbol{P}$ under such rotation is

$$\boldsymbol{F}^{+} = \boldsymbol{Q}\boldsymbol{F} \qquad \boldsymbol{P}^{+} = \boldsymbol{Q}\boldsymbol{P}, \tag{2.11}$$

while $\boldsymbol{q}_0$ and $\nabla\theta$ are invariant vector fields with

$$\boldsymbol{q}_0^{+} = \boldsymbol{q}_0 \qquad \nabla\theta^{+} = \nabla\theta. \tag{2.12}$$

Frame-indifference implies the response of Equation (2.10) should satisfy

$$\begin{aligned}
\bar{\psi}\left(\boldsymbol{F}, \theta, \nabla\theta\right) &= \bar{\psi}\left(\boldsymbol{Q}\boldsymbol{F}, \theta, \nabla\theta\right), \\
\bar{\boldsymbol{P}}\left(\boldsymbol{F}, \theta, \nabla\theta\right) &= \boldsymbol{Q}^{T}\bar{\boldsymbol{P}}\left(\boldsymbol{Q}\boldsymbol{F}, \theta, \nabla\theta\right), \\
\bar{\eta}\left(\boldsymbol{F}, \theta, \nabla\theta\right) &= \bar{\eta}\left(\boldsymbol{Q}\boldsymbol{F}, \theta, \nabla\theta\right), \\
\bar{\boldsymbol{q}}_0\left(\boldsymbol{F}, \theta, \nabla\theta\right) &= \bar{\boldsymbol{q}}_0\left(\boldsymbol{Q}\boldsymbol{F}, \theta, \nabla\theta\right),
\end{aligned} \tag{2.13}$$

for all rotations $\boldsymbol{Q}$ and all $(\boldsymbol{F}, \theta, \nabla\theta)$. Therefore taking $\boldsymbol{Q} = \boldsymbol{R}^T$ in (2.13) and recalling (2.4) and (2.5), the reduced constitutive functions are

$$\begin{aligned} \psi &= \bar{\psi}\left(\boldsymbol{C}, \theta, \nabla\theta\right), \\ \boldsymbol{P} &= \boldsymbol{F}\bar{\boldsymbol{P}}\left(\boldsymbol{C}, \theta, \nabla\theta\right), \\ \eta &= \bar{\eta}\left(\boldsymbol{C}, \theta, \nabla\theta\right), \\ \boldsymbol{q}_0 &= \bar{\boldsymbol{q}}_0\left(\boldsymbol{C}, \theta, \nabla\theta\right). \end{aligned} \tag{2.14}$$

Next, the Coleman–Noll procedure is applied to the frame-indifferent constitutive equations 2.14. The basic hypothesis of the procedure is that the body force $\boldsymbol{b}$ and heat supply r are arbitrarily assignable, which are then applied to Equation (2.8). The detailed derivations can be found in [61, Section 57.2]. With the thermodynamic restrictions imposed, the constitutive relations in Equation (2.14) reduce to

$$\begin{aligned} &\psi = \bar{\psi}\left(\boldsymbol{C}, \theta\right), \\ &\boldsymbol{P} = \bar{\boldsymbol{P}}\left(\boldsymbol{C}, \theta\right) = 2\boldsymbol{F}\frac{\partial\bar{\psi}\left(\boldsymbol{C}, \theta\right)}{\partial\boldsymbol{C}}, \\ &\eta = \bar{\eta}\left(\boldsymbol{C}, \theta\right) = -\frac{\partial\bar{\psi}\left(\boldsymbol{C}, \theta\right)}{\partial\theta}, \\ &\bar{\boldsymbol{q}}_0\left(\boldsymbol{C}, \theta, \nabla\theta\right)\cdot\nabla\theta \leqslant 0, \quad \textit{for all } \left(\boldsymbol{C}, \theta, \nabla\theta\right), \end{aligned} \tag{2.15}$$

The second Piola-Kirchoff stress $\boldsymbol{S}$ is defined as

$$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}. \tag{2.16}$$

From Equation $(2.15)_2$, it is concluded that

$$\boldsymbol{S} = \bar{\boldsymbol{S}}\left(\boldsymbol{C}, \theta\right) = 2\frac{\partial\bar{\psi}\left(\boldsymbol{C}, \theta\right)}{\partial\boldsymbol{C}}. \tag{2.17}$$

The results in Equation (2.15) also lead to the introduction of the elasticity tensor, denoted by $\mathbb{C}$, that describes the dependence of the stress tensor $\boldsymbol{S}$ on $\boldsymbol{C}$. Using Equation (2.17), $\mathbb{C}$ is defined as

$$\mathbb{C}\left(\boldsymbol{C}, \theta\right) = 2\frac{\partial\bar{\boldsymbol{S}}\left(\boldsymbol{C}, \theta\right)}{\partial\boldsymbol{C}} = 4\frac{\partial^2\bar{\psi}\left(\boldsymbol{C}, \theta\right)}{\partial\boldsymbol{C}^2}. \tag{2.18}$$

By (2.15) and (2.17), the Gibbs relation is obtained by differentiating ψ with respect to time

$$\dot{\psi} = \frac{1}{2}\boldsymbol{S}\cdot\dot{\boldsymbol{C}} - \eta\dot{\theta}. \tag{2.19}$$

Further, by Equation (2.9)

$$\dot{\psi} + \eta\dot{\theta} = \rho_0\dot{e} - \theta\dot{\eta}, \tag{2.20}$$

and therefore a second Gibbs relations is obtained from (2.19) and (2.9) as

$$\rho_0\dot{e} = \frac{1}{2}\boldsymbol{S}\cdot\dot{\boldsymbol{C}} + \theta\dot{\eta}. \tag{2.21}$$

It is obvious based on the definitions (2.4) and (2.16) that

$$\frac{1}{2}\boldsymbol{S}\cdot\dot{\boldsymbol{C}} = \boldsymbol{S}\cdot\dot{\boldsymbol{E}} = \boldsymbol{P}\cdot\dot{\boldsymbol{F}}. \tag{2.22}$$

Substituting Equation (2.21) into the energy balance (2.7) leads to

$$\theta\dot{\eta} = -\nabla\boldsymbol{q}_0 + \rho_0 r. \tag{2.23}$$

Next, differentiate ψ twice and define the following quantities

$$\boldsymbol{M}\left(\boldsymbol{C},\theta\right) = 2\frac{\partial^2\bar{\psi}\left(\boldsymbol{C},\theta\right)}{\partial\boldsymbol{C}\partial\theta} \qquad c\left(\boldsymbol{C},\theta\right) = -\theta\frac{\partial^2\bar{\psi}\left(\boldsymbol{C},\theta\right)}{\partial\theta^2}. \tag{2.24}$$

Here, $\boldsymbol{M}$ is the stress-temperature modulus and c is the heat capacity. Further, differentiating η with respect to time and multiplying by θ based on the constitutive relations (2.15) and (2.24) leads to

$$\begin{aligned}\theta\dot{\eta} &= \theta\frac{\partial\bar{\eta}\left(\boldsymbol{C},\theta\right)}{\partial\boldsymbol{C}}\cdot\dot{\boldsymbol{C}} + \theta\frac{\partial\bar{\eta}\left(\boldsymbol{C},\theta\right)}{\partial\theta}\dot{\theta} \\ &= -\theta\boldsymbol{M}\left(\boldsymbol{C},\theta\right)\cdot\dot{\boldsymbol{E}} + c\left(\boldsymbol{C},\theta\right)\dot{\theta}.\end{aligned} \tag{2.25}$$

Here Equation (2.24) and $\frac{1}{2}\dot{\boldsymbol{C}} = \dot{\boldsymbol{E}}$ based on (2.4) are applied. By Equations (2.25) and (2.23) the energy balance (dropping explicit dependency notations for brevity) can take the following form

$$c\dot{\theta} = -\nabla\cdot\boldsymbol{q}_0 + \rho_0 r + \theta\boldsymbol{M}\cdot\dot{\boldsymbol{E}}. \tag{2.26}$$

Moreover, assume that the heat flux observes Fourier's law, given as

$$\boldsymbol{q}_0 = -\boldsymbol{K}\nabla\theta, \tag{2.27}$$

where $\boldsymbol{K}$ is the conductivity tensor. Convection heat transfer, usually the dominant form of heat transfer in liquids and gases, is also used in the model at the boundary. The heat flux $\bar{q}$ is calculated as

$$\bar{q} = h\left(\theta_\infty - \theta\right), \tag{2.28}$$

where h is the convection coefficient and θ_∞ is the ambient temperature. Some literature [14] considers radiation heat transfer given by the Stefan-Boltzmann law as well, in the form

$$q_{rad} = \epsilon\sigma\left(\theta^4 - \theta_\infty^4\right), \tag{2.29}$$

where ϵ and σ are the surface emissivity and the Stefan-Boltzmann constant. The effect of radiation here is less prominent as FDM involves a much lower temperature of $500K$ compared to the metal melting methods of $2000K$. Therefore it is not included in the calculations and analysis.

In order to further specify the constitutive relations, the reference state is chosen as

$$\left(\boldsymbol{C},\theta\right)|_0 = \left(\boldsymbol{I},\theta_0\right), \tag{2.30}$$

where the subscript 0 denotes the reference state and θ_0 is the reference temperature. For a configuration to be natural at θ_0, meaning it can withstand small perturbations in strain and temperature, several conditions need to be satisfied. The derivations are deferred to [61, Section 58.2] and the conclusions are described briefly here. First, barring external forces, the residual stress should vanish. Second, the elasticity tensor $\mathbb{C}$ should be symmetric and positive-definite. Last, the heat capacity c needs to be nonnegative. It is easy to verify they all hold with the specified constitutive law for Helmholtz energy later in Equation (2.41).

For the scale and condition of the problems considered in the dissertation, the model is simplified by assuming infinitesimal deformations with respect to the reference state. This will prove to be consistent with the results in Chapter 3. To this end, define the relative displacement gradient as

$$\boldsymbol{H} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}}. \tag{2.31}$$

The infinitesimal deformation assumption gives

$$\begin{aligned} \boldsymbol{F} =& \mathbf{1} + \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}} = \mathbf{1} + \boldsymbol{H}, \\ \boldsymbol{E} =& \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right) + o(|\boldsymbol{H}|^2), \end{aligned} \tag{2.32}$$

as $|\boldsymbol{H}| \to 0$, where

$$\lim_{|\boldsymbol{H}|^2 \to 0} \frac{|o(|\boldsymbol{H}|)|}{|\boldsymbol{H}|} = 0, \tag{2.33}$$

and $|\boldsymbol{H}|$ denotes any norm of $\boldsymbol{H}$.The infinitesimal strain $\boldsymbol{\epsilon}$ hence is defined as

$$\boldsymbol{\epsilon} = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right), \tag{2.34}$$

where $\boldsymbol{\epsilon}$ is equivalent to $\boldsymbol{E}$ under the infinitesimal deformation assumption. The determinant of the deformation gradient $\boldsymbol{F}$ under infinitesimal strain J simplifies to

$$J = 1 + tr\boldsymbol{\epsilon}, \tag{2.35}$$

where tr denotes trace.Expanding $\boldsymbol{S}(\boldsymbol{C}, \theta)$ about the reference state $(\boldsymbol{I}, \theta_0)$ in deformation while fixing temperature, and using Equation (2.18) gives rise to

$$\begin{aligned} \bar{\boldsymbol{S}}(\boldsymbol{C}, \theta)|_{\theta=\theta_0} =& \boldsymbol{S}|_0 + \frac{\partial \bar{\boldsymbol{S}}(\boldsymbol{C}, \theta)}{\partial \boldsymbol{C}}|_0 (\boldsymbol{C} - \boldsymbol{I}) + o(|\boldsymbol{C} - \boldsymbol{I}|) \qquad \boldsymbol{C} \to \boldsymbol{I}, \\ =& \mathbb{C}|_0 \boldsymbol{E} + o(|\boldsymbol{E}|) \qquad \boldsymbol{E} \to \mathbf{0}. \end{aligned} \tag{2.36}$$

Here the zero residual stress condition for the reference state is applied. Further by Equation (2.16) and (2.32),

$$\bar{\boldsymbol{P}}(\boldsymbol{C}, \theta)|_{\theta=\theta_0} = \bar{\boldsymbol{S}}(\boldsymbol{C}, \theta)|_{\theta=\theta_0} + o(|\boldsymbol{H}|) \qquad |\boldsymbol{H}| \to 0. \tag{2.37}$$

Therefore the stress measures coincide under infinitesimal deformation. From here on it is denoted as σ, and

$$\boldsymbol{\sigma} = \boldsymbol{P} = \boldsymbol{S}. \tag{2.38}$$

Since the stress measures differ only by the magnitude of the displacement gradient, Equation (2.38) holds true for temperature away from the reference temperature as well. Using Equation (2.18) and (2.38) in Equation (2.36), and ignoring the higher order terms, for an isothermal expansion about the reference temperature of the stress, its relation with the strain is

$$\boldsymbol{\sigma}\left(\boldsymbol{\epsilon}, \theta\right)|_{\theta=\theta_0} = \mathbb{C}_0\boldsymbol{\epsilon}. \tag{2.39}$$

The notation $\mathbb{C}_0$ is used to indicate the elasticity tensor at the reference temperature. In addition, assuming isotropic material properties and through a standard derivation given in [61, Section 59.3], further simplification is made to Equation (2.39) and the conductivity tensor in Equation (2.27)

$$\begin{aligned} \mathbb{C}_0\boldsymbol{\epsilon} =& 2\mu\boldsymbol{\epsilon} + \lambda(tr\boldsymbol{\epsilon})\boldsymbol{I}, \\ \boldsymbol{K} =& k\boldsymbol{I}, \end{aligned} \tag{2.40}$$

where k is the conductivity and assumed to be constant and uniform, μ and λ are the material elastic moduli obtained from experiments at the given temperature. Though the conductivity depends on temperature, the change in value within the temperature range considered is negligible based on experimental data from literature [63]. As a result it is assumed to be constant.

Notably no assumption is made on temperature changes due to the fact that FDM process incurs large variation in temperature. Usually the material used in FDM is a type of strong, durable thermoplastics such as acrylonitrile butadiene styrene (ABS), polycarbonate (PC) or ULTEM$^{\text{TM}}$ 9085 Resin [64]. Enhancements are made to these materials through various techniques such as blending them with metals to form composites [65]. Hence, the constitutive model for a specific material can be very complicated and hard to obtain without sufficient experimental data. Following [66], the Helmholtz free energy is chosen as the constitutive law for a typical material used in FDM within the deformation and temperature range targeted, according to

$$\psi(\boldsymbol{\epsilon}, \theta) = \tilde{\psi}(\boldsymbol{\epsilon}, \theta) = f\left(\theta\right)\frac{1}{2}\boldsymbol{\epsilon}\cdot\mathbb{C}_0\boldsymbol{\epsilon} + \kappa\alpha\ln(1 + tr\boldsymbol{\epsilon})(\theta_0 - \theta) + \bar{c}(\theta - \theta_0 - \theta\ln\frac{\theta}{\theta_0}), \tag{2.41}$$

where $\kappa = \lambda + \frac{2}{3}\mu$ is the isothermal bulk modulus, α is the coefficient of thermal expansion, and $\bar{c}$ is the specific heat at reference temperature. The function $f\left(\theta\right)$ in Equation (2.41) represents the dependency of elasticity tensor on temperature. Since infinitesimal deformation is assumed in this dissertation, this function reflects the impact of temperature field on the mechanical response of the system. The function f will be specified when the test problems are described in Chapter 3. Following Equation (2.26) and Equation (2.34), the balance laws can be written as

$$\begin{aligned} \rho_0\boldsymbol{a} &= \nabla\cdot\boldsymbol{\sigma} + \rho_0\boldsymbol{b}, \\ c\dot{\theta} &= -\nabla\cdot\boldsymbol{q}_0 + \rho_0 r + \theta\boldsymbol{M}\cdot\dot{\boldsymbol{\epsilon}}, \end{aligned} \tag{2.42}$$

while the angular momentum balance ensures the stress $\boldsymbol{\sigma}$ is symmetric [61]. Combining Equations (2.15), (2.41) and (2.38), the stress-strain relation becomes

$$\boldsymbol{\sigma} = f(\theta)\,\mathbb{C}_0\boldsymbol{\epsilon} + \kappa_0\alpha_0(\theta_0 - \theta)\,\frac{1}{1+tr\boldsymbol{\epsilon}}\boldsymbol{I}. \tag{2.43}$$

It is worth pointing out that Equation (2.43) derived from the specific Helmholtz free energy in Equation (2.41) satisfies the conditions from isothermal linear elasticity given in Equation (2.39) provided $f(\theta_0) = 1$.

Substituting Equation (2.41) into (2.24), the stress-temperature modulus $\boldsymbol{M}$ is given by

$$\boldsymbol{M} = \frac{\partial f}{\partial\theta}\mathbb{C}_0\boldsymbol{\epsilon} - \kappa_0\alpha_0\frac{1}{1+tr\boldsymbol{\epsilon}}\boldsymbol{I}. \tag{2.44}$$

Finally, the specific heat c can also be derived from Equations (2.24) and (2.41) as

$$c = -\frac{\partial^2 f}{\partial\theta^2}\theta\frac{1}{2}\boldsymbol{\epsilon}\cdot\mathbb{C}_0\boldsymbol{\epsilon} + \bar{c}. \tag{2.45}$$

2.2.2 Finite Element Method

Given the simplified balance laws in the referential configuration in Equation (2.42), the continuous Lagrangian finite element model is developed accordingly. The integral form of the balance laws can be written as

$$\begin{aligned} \int_{\Omega_0}\nabla\cdot\boldsymbol{\sigma}\;dV &= \int_{\Omega_0}\rho_0(\boldsymbol{a}-\boldsymbol{b})\;dV, \\ \int_{\Omega_0}c\dot{\theta}\;dV + \int_{\Omega_0}\nabla\cdot\boldsymbol{q}_0\;dV &= \int_{\Omega_0}\rho_0 r\;dV + \int_{\Omega_0}\theta\boldsymbol{M}\cdot\dot{\boldsymbol{\epsilon}}\;dV, \end{aligned} \tag{2.46}$$

where Ω_0 is the referential domain of interest. Consider the solution space for both displacement and temperature,

$$\begin{aligned} L_2(\Omega) &= \left\{\boldsymbol{v}:\ \text{defined on } \Omega \text{ and } \int_\Omega |\boldsymbol{v}|^2\,dx < \infty\right\}, \\ H^1(\Omega) &= \left\{\boldsymbol{v}\in L_2(\Omega) : \frac{\partial\boldsymbol{v}}{\partial x_i}\in L_2(\Omega)\ \text{ for } i=1,...d\right\}, \\ H_0^1(\Omega) &= \left\{\boldsymbol{v}\in H^1(\Omega) : \boldsymbol{v} = \bar{\boldsymbol{v}} \text{ on } \Gamma_N\right\}, \end{aligned} \tag{2.47}$$

where $d = 2$ is the number of space dimensions, H^1 the Sobolev space of order 2, and Γ_N the Dirichlet boundary. Weight functions $\boldsymbol{\xi}$ for the linear momentum balance and ζ for energy balance belong to the same space satisfying corresponding Dirichlet boundary conditions in $H_0^1(\Omega)$. Multiplying with weight functions and using the divergence theorem, the weighted

residual for the balance laws are

$$\begin{aligned}\int_{\Omega_0} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \boldsymbol{\sigma}\ dV &= \int_{\Gamma^u_{N,0}} \boldsymbol{\xi} \cdot \bar{\boldsymbol{p}}\ dA + \int_{\Omega_0} \boldsymbol{\xi} \cdot \rho_0(\boldsymbol{b} - \boldsymbol{a})\ dV, \\ \int_{\Omega_0} \zeta c\dot{\theta}\ dV - \int_{\Omega_0} \frac{\partial \zeta}{\partial \boldsymbol{X}} \cdot \boldsymbol{q}_0\ dV &= \int_{\Omega_0} \zeta \rho_0 r\ dV + \int_{\Omega_0} \zeta\theta \boldsymbol{M} \cdot \dot{\boldsymbol{\epsilon}}\ dV - \int_{\Gamma^t_{N,0}} \zeta \bar{q}\ dA,\end{aligned} \tag{2.48}$$

where $\Gamma^t_{N,0}$ represents the part of the boundary with Neumann boundary conditions on temperature, and $\Gamma^u_{N,0}$ represents the part with Neumann boundary conditions on displacement. The solution is approximated using piecewise polynomials on isoparametric elements. A quantity $\boldsymbol{v}(\boldsymbol{X}, t)$ can be approximated as $\boldsymbol{v}_h(\boldsymbol{X}, t)$ by basis functions $\boldsymbol{\phi}_i(\boldsymbol{X})$ of the solution space according to

$$\boldsymbol{v}_h(\boldsymbol{X}, t) = \sum_{i=1}^{n} \boldsymbol{\phi}_i(\boldsymbol{X}) v_i(t), \tag{2.49}$$

where the subscript h denotes the approximation and v_i are the corresponding coefficients for the basis functions. To form the discretized matrix form of the finite element method, for a typical Lagrangian element e with domain Ω^e_0 in the reference configuration, at time $t = t_{n+1}$, the nodal values at the vertices of an element of the quantity, denoted by $\left[\hat{\boldsymbol{v}}^e_{n+1}\right]$, can be expressed in vector form as

$$\begin{bmatrix} \boldsymbol{v}^e_1 & \boldsymbol{v}^e_2 & ... & \boldsymbol{v}^e_{nen} \end{bmatrix}^T_{n+1}, \tag{2.50}$$

where nen is the number of nodes in each element. For a variable with d degrees of freedom, $\left[\hat{\boldsymbol{v}}^e_{n+1}\right]$ will have $d \times nen$ number of entries. To simplify the notation, for the approximate solution of $\boldsymbol{v}$ we write $\boldsymbol{v}_h\left(\boldsymbol{X}, t_{n+1}\right)|_{\Omega^e_0} = \boldsymbol{v}^e_{n+1}$. Apply this now to both displacement-and-temperature-like variables in two-dimensions and the interpolation of variables are given as

$$\begin{aligned}\boldsymbol{u}_h\left(\boldsymbol{X}, t_{n+1}\right)|_{\Omega^e_0} &= \boldsymbol{u}^e_{n+1} = \left[\boldsymbol{N}^e_u\right]\left[\hat{\boldsymbol{u}}^e_{n+1}\right], \\ \boldsymbol{a}_h\left(\boldsymbol{X}, t_{n+1}\right)|_{\Omega^e_0} &= \boldsymbol{a}^e_{n+1} = \left[\boldsymbol{N}^e_u\right]\left[\hat{\boldsymbol{a}}^e_{n+1}\right], \\ \boldsymbol{\xi}_h\left(\boldsymbol{X}, t_{n+1}\right)|_{\Omega^e_0} &= \boldsymbol{\xi}^e_{n+1} = \left[\boldsymbol{N}^e_u\right]\left[\hat{\boldsymbol{\xi}}^e_{n+1}\right], \\ \theta_h\left(\boldsymbol{X}, t_{n+1}\right)|_{\Omega^e_0} &= \theta^e_{n+1} = \left[\boldsymbol{N}^e_t\right]\left[\hat{\boldsymbol{\theta}}^e_{n+1}\right], \\ \zeta_h\left(\boldsymbol{X}, t_{n+1}\right)|_{\Omega^e_0} &= \zeta^e_{n+1} = \left[\boldsymbol{N}^e_t\right]\left[\hat{\boldsymbol{\zeta}}^e_{n+1}\right],\end{aligned} \tag{2.51}$$

where $\left[\hat{\boldsymbol{u}}^e_{n+1}\right], \left[\hat{\boldsymbol{a}}^e_{n+1}\right], \left[\hat{\boldsymbol{\xi}}^e_{n+1}\right], \left[\hat{\boldsymbol{\theta}}^e_{n+1}\right], \left[\hat{\boldsymbol{\zeta}}^e_{n+1}\right]$ are the nodal values at the vertices of the element ordered in vector form as in (2.50). The interpolation matrices are given as

$$\begin{aligned}\left[\boldsymbol{N}^e_u\right] &= \begin{bmatrix} N^e_1 & 0 & N^e_2 & 0 & ... & N^e_{nen} & 0 \\ 0 & N^e_1 & 0 & N^e_2 & ... & 0 & N^e_{nen} \end{bmatrix}, \\ \left[\boldsymbol{N}^e_t\right] &= \begin{bmatrix} N^e_1 & N^e_2 & ... & N^e_{nen} \end{bmatrix},\end{aligned} \tag{2.52}$$

with the subscript u denoting displacement and θ for temperature. First-order differential terms are written as vectors and the dot product can be computed with vector multiplication. Angular momentum balance ensures symmetric $\boldsymbol{\xi}$ as mentioned in Section 2.2.1. The vector form of these terms in two-dimensions, omitting dependencies on $\boldsymbol{X}$ and t_{n+1}, are then given as

$$\begin{aligned}\left[\frac{\partial \boldsymbol{\xi}^e_{n+1}}{\partial \boldsymbol{X}}\right] &= \begin{bmatrix}\xi^e_{1,1} & \xi^e_{2,2} & \xi^e_{1,2} & \xi^e_{2,1}\end{bmatrix}^T_{n+1}, \\ \left[\frac{\partial \zeta^e_{n+1}}{\partial \boldsymbol{X}}\right] &= \begin{bmatrix}\zeta^e_{,1} & \zeta^e_{,2}\end{bmatrix}^T_{n+1}.\end{aligned} \tag{2.53}$$

This transfer from tensor to matrix form is usually called a "Voigt notation". Similarly we write the components of the stress vector $\boldsymbol{\sigma}^e_{n+1}$ as

$$\left[\boldsymbol{\sigma}^e_{n+1}\right] = \begin{bmatrix}\sigma^e_{11} & \sigma^e_{22} & \sigma^e_{12} & \sigma^e_{21}\end{bmatrix}^T_{n+1}. \tag{2.54}$$

The subscripts of the entries in (2.53) are defined by

$$\xi^e_{i,j} = \frac{\partial \xi^e_i}{\partial X_j} \qquad \eta^e_{,i} = \frac{\partial \eta^e}{\partial X_i}. \tag{2.55}$$

For (2.54) the subscript of an entry represents its row and column number in the matrix form. The matrix forms of $\boldsymbol{M}^e_{n+1}$ and $\dot{\boldsymbol{\epsilon}}^e_{n+1}$ are

$$\left[\boldsymbol{M}^e_{n+1}\right] = \begin{bmatrix}M^e_{11} & M^e_{22} & M^e_{12} & M^e_{21}\end{bmatrix}^T_{n+1}, \ \left[\dot{\boldsymbol{\epsilon}}^e_{n+1}\right] = \begin{bmatrix}\dot{\epsilon}^e_{11} & \dot{\epsilon}^e_{22} & \dot{\epsilon}^e_{12} & \dot{\epsilon}^e_{21}\end{bmatrix}^T_{n+1}. \tag{2.56}$$

The dot product between them can then be expressed as $\left[\boldsymbol{M}^e_{n+1}\right]^T \left[\dot{\boldsymbol{\epsilon}}^e_{n+1}\right]$. The matrix interpolation forms of the gradient are given as

$$\left[\frac{\partial \boldsymbol{\xi}^e_{n+1}}{\partial \boldsymbol{X}}\right] = [\boldsymbol{B}^e_u]\left[\hat{\boldsymbol{\xi}}^e_{n+1}\right] \qquad \left[\frac{\partial \zeta^e_{n+1}}{\partial \boldsymbol{X}}\right] = [\boldsymbol{B}^e_t]\left[\hat{\boldsymbol{\zeta}}^e_{n+1}\right], \tag{2.57}$$

where $\left[\hat{\boldsymbol{\xi}}^e_{n+1}\right]$ and $\left[\hat{\boldsymbol{\zeta}}^e_{n+1}\right]$ are defined in Equation (2.51). The gradient operator matrices $[\boldsymbol{B}^e_u]$ and $[\boldsymbol{B}^e_t]$ can be written as

$$\begin{aligned}[\boldsymbol{B}^e_u] &= \begin{bmatrix}N^e_{1,1} & 0 & \dots & N^e_{nen,1} & 0 \\ 0 & N^e_{1,2} & \dots & 0 & N^e_{nen,2} \\ N^e_{1,2} & 0 & \dots & N^e_{nen,2} & 0 \\ 0 & N^e_{1,1} & \dots & 0 & N^e_{nen,1}\end{bmatrix}_{n+1}, \\ [\boldsymbol{B}^e_t] &= \begin{bmatrix}N^e_{1,1} & N^e_{2,1} & \dots & N^e_{nen,1} \\ N^e_{1,2} & N^e_{2,2} & \dots & N^e_{nen,2}\end{bmatrix}_{n+1},\end{aligned} \tag{2.58}$$

where $N^e_{i,j} = \frac{\partial N^e_i}{\partial X_j}$.

Substituting all interpolations into the weak form (2.48), one finds that

$$\begin{aligned}
&\int_{\Omega_0^e} \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \left[\boldsymbol{B}_u^e\right]^T \left[\boldsymbol{\sigma}_{n+1}^e\right] \, dV = \int_{\Gamma_{N,0}^u \cap \partial\Omega_0^e} \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \left[\boldsymbol{N}_u^e\right]^T \left[\bar{\boldsymbol{p}}_{n+1}^e\right] \, dA \\
&\qquad \int_{\partial\Omega_0^e \setminus \Gamma_{N,0}^u} \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \left[\boldsymbol{N}_u^e\right]^T \left[\boldsymbol{p}_{n+1}^e\right] \, dA + \int_{\Omega_0^e} \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \left[\boldsymbol{N}_u^e\right]^T \rho_0 ([\boldsymbol{b}_{n+1}] - \left[\boldsymbol{N}_u^e\right] \left[\hat{\boldsymbol{a}}_{n+1}^e\right]) \, dV, \\
&\int_{\Omega_0^e} \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left[\boldsymbol{N}_t^e\right]^T c_{n+1}^e \left[\hat{\dot{\boldsymbol{\theta}}}_{n+1}^e\right] \, dV = \int_{\Omega_0^e} \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left[\boldsymbol{B}_t^e\right]^T \left[\boldsymbol{q}_{0,n+1}^e\right] \, dV \\
&\qquad + \int_{\Omega_0^e} \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left[\boldsymbol{N}_t^e\right]^T \rho_0 r \, dV + \int_{\Omega_0^e} \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left[\boldsymbol{N}_t^e\right]^T \theta_{n+1}^e \left[\boldsymbol{M}_{n+1}^e\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^e\right] \, dV \\
&\qquad - \int_{\Gamma_{N,0}^t \cap \partial\Omega_0^e} \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left[\boldsymbol{N}_t^e\right]^T \left[\bar{\boldsymbol{q}}_{n+1}^e\right] \, dA - \int_{\partial\Omega_0^e \setminus \Gamma_{N,0}^t} \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left[\boldsymbol{N}_t^e\right]^T \left[\boldsymbol{q}_{n+1}^e\right] \, dA,
\end{aligned} \tag{2.59}$$

where $\Gamma_{N,0}^u \cap \partial\Omega_0^e$ is the part of the element boundary which (potentially) intersects with the Neumann boundary $\Gamma_{N,0}^u$ for displacement and $\partial\Omega_0^e \setminus \Gamma_{N,0}^u$ is the rest of the element boundary. The notation for the temperature is similar.

This simplifies to

$$\begin{aligned}
&\left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \int_{\Omega_0^e} \left[\boldsymbol{B}_u^e\right]^T \left[\boldsymbol{\sigma}_{n+1}^e\right] \, dV = \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \int_{\Gamma_{N,0}^u \cap \partial\Omega_0^e} \left[\boldsymbol{N}_u^e\right]^T \left[\bar{\boldsymbol{p}}_{n+1}^e\right] \, dA \\
&\qquad + \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \int_{\partial\Omega_0^e \setminus \Gamma_{N,0}^u} \left[\boldsymbol{N}_u^e\right]^T \left[\boldsymbol{p}_{n+1}^e\right] \, dA + \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \int_{\Omega_0^e} \left[\boldsymbol{N}_u^e\right]^T \rho_0 \left[\boldsymbol{b}_{n+1}\right] \, dV \\
&\qquad\qquad - \left[\hat{\boldsymbol{\xi}}_{n+1}^e\right]^T \left(\int_{\Omega_0^e} \left[\boldsymbol{N}_u^e\right]^T \rho_0 \left[\boldsymbol{N}_u^e\right] \, dV \right) \left[\hat{\boldsymbol{a}}_{n+1}^e\right], \\
&\left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \left(\int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T c_{n+1}^e \left[\boldsymbol{N}_t^e\right] \, dV \right) \left[\hat{\dot{\boldsymbol{\theta}}}_{n+1}^e\right] = \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \int_{\Omega_0^e} \left[\boldsymbol{B}_t^e\right]^T \left[\boldsymbol{q}_{0,n+1}^e\right] \, dV \\
&\qquad + \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \rho_0 r \, dV + \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \theta_{n+1}^e \left[\boldsymbol{M}_{n+1}^e\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^e\right] \, dV \\
&\qquad\qquad - \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \int_{\Gamma_{N,0}^t \cap \partial\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \left[\bar{\boldsymbol{q}}_{n+1}^e\right] \, dA - \left[\hat{\boldsymbol{\zeta}}_{n+1}^e\right]^T \int_{\partial\Omega_0^e \setminus \Gamma_{N,0}^t} \left[\boldsymbol{N}_t^e\right]^T \left[\boldsymbol{q}_{n+1}^e\right] \, dA.
\end{aligned} \tag{2.60}$$

Since $\left[\hat{\boldsymbol{\xi}}^e_{n+1}\right]$ and $\left[\hat{\boldsymbol{\zeta}}^e_{n+1}\right]$ are arbitrary vectors, it follows that

$$\begin{aligned}
\int_{\Omega^e_0} \left[\boldsymbol{B}^e_u\right]^T \left[\boldsymbol{\sigma}^e_{n+1}\right] \, dV &= \int_{\Gamma^u_{N,0}\cap\partial\Omega^e_0} \left[\boldsymbol{N}^e_u\right]^T \left[\bar{\boldsymbol{p}}^e_{n+1}\right] \, dA + \int_{\Omega^e_0} \left[\boldsymbol{N}^e_u\right]^T \rho_0 \left[\boldsymbol{b}_{n+1}\right] \, dV \\
&\quad - \int_{\Omega^e_0} \left[\boldsymbol{N}^e_u\right]^T \rho_0 \left[\boldsymbol{N}^e_u\right] \, dV \left[\hat{\boldsymbol{a}}^e_{n+1}\right] + \int_{\partial\Omega^e_0\setminus\Gamma^u_{N,0}} \left[\boldsymbol{N}^e_u\right]^T \left[\boldsymbol{p}^e_{n+1}\right] \, dA, \\
\int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T c^e_{n+1} \left[\boldsymbol{N}^e_t\right] \, dV \left[\hat{\boldsymbol{\theta}}^e_{n+1}\right] &= \int_{\Omega^e_0} \left[\boldsymbol{B}^e_t\right]^T \left[\boldsymbol{q}^e_{0,n+1}\right] \, dV + \int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \rho_0 r \, dV \\
&\quad + \int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \theta^e_{n+1} \left[\boldsymbol{M}^e_{n+1}\right]^T \left[\dot{\boldsymbol{\epsilon}}^e_{n+1}\right] \, dV - \int_{\Gamma^t_{N,0}\cap\partial\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \left[\bar{\boldsymbol{q}}^e_{n+1}\right] \, dA \\
&\quad - \int_{\partial\Omega^e_0\setminus\Gamma^t_{N,0}} \left[\boldsymbol{N}^e_t\right]^T \left[\boldsymbol{q}^e_{n+1}\right] \, dA.
\end{aligned} \tag{2.61}$$

Fourier's law is applied to the heat flux, as in Equation (2.27) and the Neumann boundary condition for heat flux is set to convection heat transfer, as given in Equation (2.28)

$$\left[\bar{\boldsymbol{q}}^e_{n+1}\right] = h\left(\left[\boldsymbol{\theta}_\infty\right] - \left[\hat{\boldsymbol{\theta}}^e_{n+1}\right]\right). \tag{2.62}$$

Starting from Equation (2.61), the element equations in matrix form can be written as

$$\begin{aligned}
&\left[\boldsymbol{M}^e_u\right]\left[\hat{\boldsymbol{a}}^e_{n+1}\right] + \left[\boldsymbol{R}^e_{u,n+1}\left(\left[\hat{\boldsymbol{u}}^e_{n+1}\right], \left[\hat{\boldsymbol{\theta}}^e_{n+1}\right]\right)\right] = \left[\boldsymbol{F}^e_{n+1}\right] + \int_{\partial\Omega^e_0\setminus\Gamma^u_{N,0}} \left[\boldsymbol{N}^e_u\right]\left[\boldsymbol{p}^e_{n+1}\right] \, dA, \\
&\left[\boldsymbol{T}^e\right]\left[\hat{\dot{\boldsymbol{\theta}}}^e_{n+1}\right] + \left[\boldsymbol{M}^e_t\right]\left[\hat{\boldsymbol{\theta}}^e_{n+1}\right] + \left[\boldsymbol{R}^e_{t,n+1}\left(\left[\hat{\boldsymbol{u}}^e_{n+1}\right], \left[\hat{\boldsymbol{\theta}}^e_{n+1}\right]\right)\right] = \\
&\qquad \left[\boldsymbol{Q}^e_{n+1}\right] - \int_{\partial\Omega^e_0\setminus\Gamma^t_{N,0}} \left[\boldsymbol{N}^e_t\right]^T \left[\boldsymbol{q}^e_{n+1}\right] \, dA,
\end{aligned} \tag{2.63}$$

where

$$
\begin{aligned}
[\boldsymbol{M}_u^e] &= \int_{\Omega_0^e} [\boldsymbol{N}_u^e]^T \, \rho_0 \, [\boldsymbol{N}_u^e] \; dV, \\
\left[\boldsymbol{R}_{u,n+1}^e\right] &= \int_{\Omega_0^e} [\boldsymbol{B}_u^e]^T \left[\boldsymbol{\sigma}_{n+1}^e\right] \; dV, \\
\left[\boldsymbol{F}_{n+1}^e\right] &= \int_{\Omega_0^e} [\boldsymbol{N}_u^e]^T \, \rho_0 \, [\boldsymbol{b}_{n+1}] \; dV + \int_{\Gamma_{N,0}^u \cap \partial\Omega_0^e} [\boldsymbol{N}_u^e]^T \left[\bar{\boldsymbol{p}}_{n+1}^e\right] \; dA, \\
\left[\boldsymbol{T}_{n+1}^e\right] &= \int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, c_{n+1}^e \, [\boldsymbol{N}_t^e] \; dV, \\
[\boldsymbol{M}_t^e] &= \int_{\Omega_0^e} k \, [\boldsymbol{B}_t^e]^T \, [\boldsymbol{B}_t^e] \; dV - \int_{\Gamma_{N,0}^t \cap \partial\Omega_0^e} [\boldsymbol{N}_t^e]^T \, h \, [\boldsymbol{N}_t^e] \; dA, \\
\left[\boldsymbol{R}_{t,n+1}^e\right] &= \int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, \theta_{n+1}^e \left[\boldsymbol{M}_{n+1}^e\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^e\right] \; dV, \\
\left[\boldsymbol{Q}_{n+1}^e\right] &= \int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, \rho_0 r_{n+1} \; dV - \int_{\Gamma_{N,0}^t \cap \partial\Omega_0^e} [\boldsymbol{N}_t^e]^T \, h \, [\boldsymbol{\theta}_\infty] \; dA.
\end{aligned}
\tag{2.64}
$$

Assembling over the entire domain with the assembly operator $\underset{e}{\mathbf{A}}$ gives

$$
\begin{aligned}
[\boldsymbol{M}_u] \, [\hat{\boldsymbol{a}}_{n+1}] + \left[\boldsymbol{R}_{u,n+1}\left([\hat{\boldsymbol{u}}_{n+1}], \left[\hat{\boldsymbol{\theta}}_{n+1}\right]\right)\right] &= [\boldsymbol{F}_{n+1}], \\
[\boldsymbol{T}] \left[\dot{\hat{\boldsymbol{\theta}}}_{n+1}\right] + [\boldsymbol{M}_t] \left[\hat{\boldsymbol{\theta}}_{n+1}\right] + \left[\boldsymbol{R}_{t,n+1}\left([\hat{\boldsymbol{u}}_{n+1}], \left[\hat{\boldsymbol{\theta}}_{n+1}\right]\right)\right] &= [\boldsymbol{Q}_{n+1}].
\end{aligned}
\tag{2.65}
$$

The global matrices and vectors in Equation (2.65) are

$$
\begin{aligned}
[\boldsymbol{M}_u] &= \underset{e}{\mathbf{A}} \, [\boldsymbol{M}_u^e], \\
[\boldsymbol{R}_{u,n+1}] &= \underset{e}{\mathbf{A}} \left[\boldsymbol{R}_{u,n+1}^e\right], \\
[\boldsymbol{F}_{n+1}] &= \underset{e}{\mathbf{A}} \left[\boldsymbol{F}_{n+1}^e\right], \\
[\boldsymbol{T}_{n+1}] &= \underset{e}{\mathbf{A}} \left[\boldsymbol{T}_{n+1}^e\right], \\
[\boldsymbol{M}_t] &= \underset{e}{\mathbf{A}} \, [\boldsymbol{M}_t^e], \\
[\boldsymbol{R}_{t,n+1}] &= \underset{e}{\mathbf{A}} \left[\boldsymbol{R}_{t,n+1}^e\right], \\
[\boldsymbol{Q}_{n+1}] &= \underset{e}{\mathbf{A}} \left[\boldsymbol{Q}_{n+1}^e\right].
\end{aligned}
\tag{2.66}
$$

The terms, $\underset{e}{\mathbf{A}} \int_{\partial\Omega_0^e \setminus \Gamma_{N,0}^u} [\boldsymbol{N}_u^e] \left[\boldsymbol{p}_{n+1}^e\right] \; dA$ and $\underset{e}{\mathbf{A}} \int_{\partial\Omega_0^e \setminus \Gamma_{N,0}^t} [\boldsymbol{N}_t^e]^T \left[\boldsymbol{q}_{n+1}^e\right] \; dA$, which describe the cumulative inter-element jump in tractions and heat flux are neglected, as is customary in continuous Galerkin finite element approximations. In future element-level derivations, these terms are ignored. The global unknown variables $\boldsymbol{u}_{n+1}$ and $\boldsymbol{\theta}_{n+1}$ are ordered in vector form,

denoted by $[\hat{\boldsymbol{u}}_{n+1}]$ and $[\hat{\boldsymbol{u}}_{n+1}]$, as

$$[\hat{\boldsymbol{u}}_{n+1}] = \begin{bmatrix}\boldsymbol{u}_1 & \boldsymbol{u}_2 & ... & \boldsymbol{u}_{nnod}\end{bmatrix}^T_{n+1}, \quad \left[\hat{\boldsymbol{\theta}}_{n+1}\right] = \begin{bmatrix}\theta_1 & \theta_2 & ... & \theta_{nnod}\end{bmatrix}^T_{n+1}, \tag{2.67}$$

where *nnod* is the total number of nodes of the entire mesh. For the time-stepping scheme, the Newmark integrator [67] based on truncated Taylor series expansion is used for displacement and backward Euler scheme is used for temperature. At time step t_{n+1}, the scheme is given in Equation (2.68) in the global vector form

$$\begin{aligned}
[\hat{\boldsymbol{u}}_{n+1}] &= [\hat{\boldsymbol{u}}_n] + [\hat{\boldsymbol{v}}_n]\,\Delta t_n + \frac{1}{2}\left\{(1-2\beta)\,[\hat{\boldsymbol{a}}_n] + 2\beta\,[\hat{\boldsymbol{a}}_{n+1}]\right\}\Delta t_n^2, \\
[\hat{\boldsymbol{v}}_{n+1}] &= [\hat{\boldsymbol{v}}_n] + \left\{(1-\gamma)\,[\hat{\boldsymbol{a}}_n] + \gamma\,[\hat{\boldsymbol{a}}_{n+1}]\right\}\Delta t_n, \\
\left[\hat{\boldsymbol{\theta}}_{n+1}\right] &= \left[\hat{\boldsymbol{\theta}}_n\right] + \left[\hat{\dot{\boldsymbol{\theta}}}_{n+1}\right]\Delta t_n.
\end{aligned} \tag{2.68}$$

Apply integration rules (2.68) to Equation (2.65) and again write them in vector forms as

$$\begin{aligned}
&\frac{1}{\beta\Delta t_n^2}\,[\boldsymbol{M}_u]\,[\hat{\boldsymbol{u}}_{n+1}] + \left[\boldsymbol{R}_{u,n+1}\left([\hat{\boldsymbol{u}}_{n+1}], \left[\hat{\boldsymbol{\theta}}_{n+1}\right]\right)\right] = \\
&\qquad [\boldsymbol{F}_{n+1}] + [\boldsymbol{M}_u]\left\{([\hat{\boldsymbol{u}}_n] + [\hat{\boldsymbol{v}}_n]\,\Delta t_n)\,\frac{1}{\beta\Delta t_n^2} + \frac{1-2\beta}{2\beta}\,[\hat{\boldsymbol{a}}_n]\right\}, \\
&\frac{1}{\Delta t_n}\,[\boldsymbol{T}_{n+1}]\left[\hat{\boldsymbol{\theta}}_{n+1}\right] + [\boldsymbol{M}_t]\left[\hat{\dot{\boldsymbol{\theta}}}_{n+1}\right] + \left[\boldsymbol{R}_{t,n+1}\left([\hat{\boldsymbol{u}}_{n+1}], \left[\hat{\boldsymbol{\theta}}_{n+1}\right]\right)\right] = [\boldsymbol{Q}_{n+1}] + \frac{1}{\Delta t_n}\,[\boldsymbol{T}_{n+1}]\left[\hat{\boldsymbol{\theta}}_n\right].
\end{aligned} \tag{2.69}$$

While there is no subscript $n+1$ for matrices such as $[\boldsymbol{M}_u]$ and $[\boldsymbol{M}_t]$, they do change over time in the sense that the integration domain changes from step to step. Coefficients $\beta = \frac{1}{4}$, $\gamma = \frac{1}{2}$ are chosen which makes the Newmark scheme equivalent to the Trapezoidal rule. Since $\gamma > \frac{1}{2}$ introduces artificial damping, it is avoided here. With the time-stepping methods, the discretized form of $\dot{\boldsymbol{\epsilon}}^e_{n+1}$ in Equation $(2.64)_6$ can be specified with Newmark integrator. It is noted that

$$\dot{\boldsymbol{\epsilon}} = \frac{\partial \boldsymbol{\epsilon}}{\partial t} = \frac{\partial}{\partial t}\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}}\right) = \frac{\partial}{\partial \boldsymbol{X}}\left(\frac{\partial \boldsymbol{u}}{\partial t}\right). \tag{2.70}$$

Therefore the discrete form is interpolated as

$$\left[\dot{\boldsymbol{\epsilon}}^e_{n+1}\right] = [\boldsymbol{B}^e_{u2}]\left[\hat{\boldsymbol{v}}^e_{n+1}\right] = [\boldsymbol{B}^e_{u2}]\left(\frac{2}{\Delta t_n}\left(\left[\hat{\boldsymbol{u}}^e_{n+1}\right] - [\hat{\boldsymbol{u}}^e_n]\right) - [\hat{\boldsymbol{v}}^e_n]\right), \tag{2.71}$$

which makes it consistent in order of accuracy with the element-level $\boldsymbol{u}^e_{n+1}$ itself and

$$[\boldsymbol{B}^e_{u2}] = \begin{bmatrix} N^e_{i,1} & 0 & ... & N^e_{nen,1} & 0 \\ 0 & N^e_{i,2} & ... & 0 & N^e_{nen,2} \\ \frac{1}{2}N^e_{i,2} & \frac{1}{2}N^e_{i,1} & ... & \frac{1}{2}N^e_{nen,2} & \frac{1}{2}N^e_{nen,1} \\ \frac{1}{2}N^e_{i,2} & \frac{1}{2}N^e_{i,1} & ... & \frac{1}{2}N^e_{nen,2} & \frac{1}{2}N^e_{nen,1} \end{bmatrix}. \tag{2.72}$$

As seen in Equation (2.69), the discrete PDE system is nonlinear and implicit and thus requires an iterative solution method. Let $\boldsymbol{U}_{n+1} = \begin{bmatrix}\boldsymbol{u}_{n+1} & \boldsymbol{\theta}_{n+1}\end{bmatrix}^T$, write the discretized system (2.69) at time t_{n+1} as

$$\boldsymbol{f}_{n+1}\left(\boldsymbol{U}_{n+1}\right) = \boldsymbol{0}. \tag{2.73}$$

Then the Newton-Raphson method states that the $(k+1)$st iterative solution is of the form

$$\begin{aligned} \Delta\boldsymbol{U}_{n+1}^{(k)} &= -\left[\boldsymbol{J}_{n+1}\left(\boldsymbol{U}_{n+1}^{(k)}\right)\right]^{-1}\boldsymbol{f}_{n+1}\left(\boldsymbol{U}_{n+1}^{(k)}\right), \\ \boldsymbol{U}_{n+1}^{(k+1)} &= \boldsymbol{U}_{n+1}^{(k)} + \Delta\boldsymbol{U}_{n+1}^{(k)}, \end{aligned} \tag{2.74}$$

where $\boldsymbol{J}_{n+1}$ is the Frechet derivative of $\boldsymbol{f}_{n+1}$. The iteration stops when $\left|\boldsymbol{U}_{n+1}^{(k+1)} - \boldsymbol{U}_{n+1}^{(k)}\right|^2 < tol$, with tol a given small positive scalar value, *e.g.* 10^{-7}. To apply the Newton-Raphson iteration , the fully discretized derivatives of Equation (2.65) are derived. At time step t_{n+1} and for the kth iteration for an element e we have

$$\begin{aligned} &\int_{\Omega_0^e}\left[\boldsymbol{N}_u^e\right]^T\rho_0\left[\boldsymbol{N}_u^e\right]\left[\hat{\boldsymbol{a}}_{n+1}^{e,(k)}\right]\,dV + \int_{\Omega_0^e}\left[\boldsymbol{B}_u^e\right]^T\left[\boldsymbol{\sigma}_{n+1}^{e,(k)}\right]\,dV - \int_{\Omega_0^e}\left[\boldsymbol{N}_u^e\right]^T\rho_0\left[\boldsymbol{b}_{n+1}\right]\,dV \\ &\quad - \int_{\partial\Omega_0^e\cap\Gamma_{N,0}^u}\left[\boldsymbol{N}_u^e\right]^T\left[\bar{\boldsymbol{p}}_{n+1}^e\right]\,dA + \int_{\Omega_0^e}\left[\boldsymbol{N}_u^e\right]^T\rho_0\frac{4}{\Delta t_n^2}\left[\boldsymbol{N}_u^e\right]\,dV\left[\Delta\hat{\boldsymbol{u}}_{n+1}^{e,(k)}\right] \\ &\qquad + \int_{\Omega_0^e}\left[\boldsymbol{B}_u^e\right]^T\left[\left(\frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\epsilon}}\right)_{n+1}^{e,(k)}\right]\left[\boldsymbol{B}_{u2}^e\right]\,dV\left[\Delta\hat{\boldsymbol{u}}_{n+1}^{e,(k)}\right] \\ &\qquad\quad + \int_{\Omega_0^e}\left[\boldsymbol{B}_u^e\right]^T\left[\left(\frac{\partial\boldsymbol{\sigma}}{\partial\theta}\right)_{n+1}^{e,(k)}\right]\left[\boldsymbol{N}_t^e\right]\,dV\left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] = \boldsymbol{0}. \end{aligned} \tag{2.75}$$

In addition,

$$\begin{aligned}
&\int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, c_{n+1}^{e,(k)} \, [\boldsymbol{N}_t^e] \; dV \left[\hat{\dot{\boldsymbol{\theta}}}_{n+1}^{e,(k)}\right] + \int_{\Omega_0^e} k \, [\boldsymbol{B}_t^e]^T \, [\boldsymbol{B}_t^e] \; dV \left[\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] - \int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, \rho_0 r \; dV - \\
&\int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, \theta_{n+1}^{e,(k)} \left[\boldsymbol{M}_{n+1}^{e,(k)}\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] \; dV + \int_{\partial\Omega_0^e \cap \Gamma_{N,0}^t} [\boldsymbol{N}_t^e]^T \left[\bar{\boldsymbol{q}}_{n+1}^e\right] \; dA + \\
&\int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \left[\left(\frac{\partial c}{\partial \theta}\right)_{n+1}^{e,(k)}\right] \dot{\theta}_{n+1}^{e,(k)} \boldsymbol{N}_t^e \; dV \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] + \int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, \frac{c_{n+1}^{e,(k)}}{\Delta t_n} \, [\boldsymbol{N}_t^e] \; dV \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] + \\
&\int_{\Omega_0^e} \left\{ k \, [\boldsymbol{B}_t^e]^T \, [\boldsymbol{B}_t^e] - [\boldsymbol{N}_t^e]^T \, \theta_{n+1}^{e,(k)} \left[\left(\frac{\partial \boldsymbol{M}}{\partial \theta}\right)_{n+1}^{e,(k)}\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] [\boldsymbol{N}_t^e] \right\} \; dV \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] - \\
&\int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \left[\boldsymbol{M}_{n+1}^{e,(k)}\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] [\boldsymbol{N}_t^e] \; dV \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] + \\
&\int_{\partial\Omega_0^e \cap \Gamma_{N,0}^t} [\boldsymbol{N}_t^e]^T \left[\left(\frac{\partial \bar{\boldsymbol{q}}}{\partial \theta}\right)_{n+1}^{e,(k)}\right] [\boldsymbol{N}_t^e] \; dA \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] + \\
&\int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \left[\left(\frac{\partial c}{\partial \boldsymbol{\epsilon}}\right)_{n+1}^{e,(k)}\right] \dot{\theta}_{n+1}^{e,(k)} \, [\boldsymbol{B}_{u2}^e] - [\boldsymbol{N}_t^e]^T \, \theta_{n+1}^{e,(k)} \left[\boldsymbol{M}_{n+1}^{e,(k)}\right]^T \frac{1}{\Delta t_n} \, [\boldsymbol{B}_{u2}^e] \; dV \left[\Delta\hat{\boldsymbol{u}}_{n+1}^{e,(k)}\right] - \\
&\int_{\Omega_0^e} [\boldsymbol{N}_t^e]^T \, \theta_{n+1}^{e,(k)} \left[\left(\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{\epsilon}}\right)_{n+1}^{e,(k)}\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] [\boldsymbol{B}_{u2}^e] \; dV \left[\Delta\hat{\boldsymbol{u}}_{n+1}^{e,(k)}\right] = \boldsymbol{0}.
\end{aligned} \tag{2.76}$$

When solving the equations, displacement and temperature are combined as

$$\left[\Delta\hat{\boldsymbol{U}}_{n+1}^{(k)}\right] = \left[\Delta\hat{\boldsymbol{u}}_{n+1}^{(k)} \quad \Delta\hat{\boldsymbol{\theta}}_{n+1}^{(k)}\right]^T. \tag{2.77}$$

In matrix form we have

$$\begin{aligned}
&[\boldsymbol{M}_u^e] \left[\hat{\boldsymbol{a}}_{n+1}^{e,(k)}\right] + \left[\boldsymbol{R}_{u,n+1}^{e,(k)}\right] - \left[\boldsymbol{F}_{n+1}^e\right] \\
&\qquad + \left[\boldsymbol{K}_{u,n+1}^{e,(k)}\right] \left[\Delta\hat{\boldsymbol{u}}_{n+1}^{e,(k)}\right] + \left[\boldsymbol{K}_{t,n+1}^{e,(k)}\right] \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] = \boldsymbol{0}, \\
&[\boldsymbol{T}^e] \left[\hat{\dot{\boldsymbol{\theta}}}_{n+1}^{e,(k)}\right] + [\boldsymbol{M}_t^e] \left[\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] + \left[\boldsymbol{R}_{t,n+1}^{e,(k)}\right] - \left[\boldsymbol{Q}_{n+1}^e\right] \\
&\qquad + \left[\boldsymbol{A}_{u,n+1}^{e,(k)}\right] \left[\Delta\hat{\boldsymbol{u}}_{n+1}^{e,(k)}\right] + \left[\boldsymbol{A}_{t,n+1}^{e,(k)}\right] \left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{e,(k)}\right] = \boldsymbol{0},
\end{aligned} \tag{2.78}$$

where

$$
\begin{aligned}
\left[\boldsymbol{K}_{u,n+1}^{e,(k)}\right] &= \int_{\Omega_0^e} \left[\boldsymbol{N}_u^e\right]^T \rho_0 \frac{4}{\Delta t_n^2} \left[\boldsymbol{N}_u^e\right] \, dV + \int_{\Omega_0^e} \left[\boldsymbol{B}_u^e\right]^T \left[\left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}}\right)_{n+1}^{e,(k)}\right] \left[\boldsymbol{B}_{u2}^e\right] \, dV, \\
\left[\boldsymbol{K}_{t,n+1}^{e,(k)}\right] &= \int_{\Omega_0^e} \left[\boldsymbol{B}_u^e\right]^T \left[\left(\frac{\partial \boldsymbol{\sigma}}{\partial \theta}\right)_{n+1}^{e,(k)}\right] \left[\boldsymbol{N}_t^e\right] \, dV, \\
\left[\boldsymbol{A}_{u,n+1}^{e,(k)}\right] &= \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \left[\left(\frac{\partial c}{\partial \boldsymbol{\epsilon}}\right)_{n+1}^{e,(k)}\right] \dot{\theta}_{n+1}^{e,(k)} \left[\boldsymbol{B}_{u2}^e\right] \, dV \\
&\quad - \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \theta_{n+1}^{e,(k)} \left[\boldsymbol{M}_{n+1}^{e,(k)}\right]^T \frac{1}{\Delta t_n} \left[\boldsymbol{B}_{u2}^e\right] \, dV \\
&\quad - \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \theta_{n+1}^{e,(k)} \left[\left(\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{\epsilon}}\right)_{n+1}^{e,(k)}\right] \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] \left[\boldsymbol{B}_{u2}^e\right] \, dV, \\
\left[\boldsymbol{A}_{t,n+1}^{e,(k)}\right] &= \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \left[\left(\frac{\partial c}{\partial \theta}\right)_{n+1}^{e,(k)}\right] \dot{\theta}_{n+1}^{e,(k)} \left[\boldsymbol{N}_t^e\right] \, dV + \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \frac{c_{n+1}^{e,(k)}}{\Delta t_n} \left[\boldsymbol{N}_t^e\right] \, dV \\
&\quad + \int_{\Omega_0^e} k \left[\boldsymbol{B}_t^e\right]^T \left[\boldsymbol{B}_t^e\right] \, dV - \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \left[\boldsymbol{M}_{n+1}^{e,(k)}\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] \left[\boldsymbol{N}_t^e\right] \, dV \\
&\quad - \int_{\Omega_0^e} \left[\boldsymbol{N}_t^e\right]^T \theta_{n+1}^{e,(k)} \left[\left(\frac{\partial \boldsymbol{M}}{\partial \theta}\right)_{n+1}^{e,(k)}\right]^T \left[\dot{\boldsymbol{\epsilon}}_{n+1}^{e,(k)}\right] \left[\boldsymbol{N}_t^e\right] \, dV \\
&\quad + \int_{\partial\Omega_0^e \cap \Gamma_{N,0}^t} \left[\boldsymbol{N}_t^e\right]^T h \left[\boldsymbol{N}_t^e\right] \, dA.
\end{aligned}
\tag{2.79}
$$

Assemble again the elemental matrices, similar to Equation (2.65), to obtain the fully discretized form of the kth iterative PDEs at time step t_{n+1}

$$
\begin{aligned}
&\left[\boldsymbol{M}_u\right] \left[\hat{\boldsymbol{a}}_{n+1}\right] + \left[\boldsymbol{R}_{u,n+1}^{(k)}\right] - \left[\boldsymbol{F}_{n+1}\right] \\
&\qquad + \left[\boldsymbol{K}_{u,n+1}^{(k)}\right] \left[\Delta \hat{\boldsymbol{u}}_{n+1}^{(k)}\right] + \left[\boldsymbol{K}_{t,n+1}^{(k)}\right] \left[\Delta \hat{\boldsymbol{\theta}}_{n+1}^{(k)}\right] = \boldsymbol{0}, \\
&\left[\boldsymbol{T}\right] \left[\dot{\hat{\boldsymbol{\theta}}}_{n+1}\right] + \left[\boldsymbol{M}_t\right] \left[\hat{\boldsymbol{\theta}}_{n+1}\right] + \left[\boldsymbol{R}_{t,n+1}^{(k)}\right] - \left[\boldsymbol{Q}_{n+1}\right] \\
&\qquad + \left[\boldsymbol{A}_{u,n+1}^{(k)}\right] \left[\Delta \hat{\boldsymbol{u}}_{n+1}^{(k)}\right] + \left[\boldsymbol{A}_{t,n+1}^{(k)}\right] \left[\Delta \hat{\boldsymbol{\theta}}_{n+1}^{(k)}\right] = \boldsymbol{0}.
\end{aligned}
\tag{2.80}
$$

Combined with the time integration schemes, the solution for this iteration and time step

are computed via

$$\begin{aligned}
&\frac{4}{\Delta t_n^2}\left[\boldsymbol{M}_u\right]\left[\hat{\boldsymbol{u}}_{n+1}\right]-\left[\boldsymbol{M}_u\right]\left\{\frac{4}{\Delta t_n^2}\left(\left[\hat{\boldsymbol{u}}_n\right]+\left[\hat{\boldsymbol{v}}_n\right]\Delta t_n\right)+\left[\hat{\boldsymbol{a}}_n\right]\right\}+\left[\boldsymbol{R}_{u,n+1}^{(k)}\right]-\left[\boldsymbol{F}_{n+1}\right]\\
&\qquad\qquad+\left[\boldsymbol{K}_{u,n+1}^{(k)}\right]\left[\Delta\hat{\boldsymbol{u}}_{n+1}^{(k)}\right]+\left[\boldsymbol{K}_{t,n+1}^{(k)}\right]\left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{(k)}\right]=\mathbf{0},\\
&\frac{1}{\Delta t_n}\left[\boldsymbol{T}\right]\left[\hat{\boldsymbol{\theta}}_{n+1}\right]-\frac{1}{\Delta t_n}\left[\boldsymbol{T}\right]\left[\hat{\boldsymbol{\theta}}_n\right]+\left[\boldsymbol{M}_t\right]\left[\hat{\boldsymbol{\theta}}_{n+1}\right]+\left[\boldsymbol{R}_{t,n+1}^{(k)}\right]-\left[\boldsymbol{Q}_{n+1}\right]\\
&\qquad\qquad+\left[\boldsymbol{A}_{u,n+1}^{(k)}\right]\left[\Delta\hat{\boldsymbol{u}}_{n+1}^{(k)}\right]+\left[\boldsymbol{A}_{t,n+1}^{(k)}\right]\left[\Delta\hat{\boldsymbol{\theta}}_{n+1}^{(k)}\right]=\mathbf{0}.
\end{aligned}\tag{2.81}$$

2.3 Mesh Generation and Slicing Algorithms

2.3.1 Fixed Lagrangian Mesh

The nature of additive manufacturing demands effective algorithms for adding elements over time that reflect the changing domain and boundaries of the printed object. It is crucial to build and assemble matrices efficiently to avoid large increase in computation cost. Most studies to date involve simulations based on a predetermined Lagrangian mesh without tracking the deformation during the printing process. Among them two distinctive methods have been applied to simulate the addition of new materials/elements.

The first one uses the so-called "quiet" elements where a referential mesh of the entire final printed structure is constructed at the beginning and adopted throughout the process. The elements that are not yet printed at a given time, *i.e.*, the "quiet" elements, are assigned artificial material properties to minimize their impact on the thermal and mechanical computations of the already printed ones. For example, thermal conductivity and specific heat for the "quiet" elements can be multiplied by a small scaling factor to reduce their contributions to heat transfer and temperature field. This method requires minimum change to the data structure and element connectivity as no new elements are being added, and thus is easy to implement. However, it inevitably introduces errors into the results. Indeed prior work shows that the errors are noticeable even in simple heat transfer cases [14]. The scaling factors can also cause ill-conditioning in the algebraic system and produce inaccurate results, hence may be challenging to choose and justify.

The second method employs "active/inactive" elements, where the entire final printed structure is meshed but only the already printed elements are included in the current time step calculation. This leads to changing size of the system and connectivity between nodes and elements. Most of the research reviewed in Chapter 1 uses such an "active/inactive" element method. While the number of "active" elements changes with time, the locations of newly added elements are computed before the simulation begins based on the path of material deposition. In other words, no deformation of previous layers and elements is dynamically considered when introducing new elements. This works well for uncoupled problems since only the heat transfer equations are solved with the added elements so the elements have no deformation.

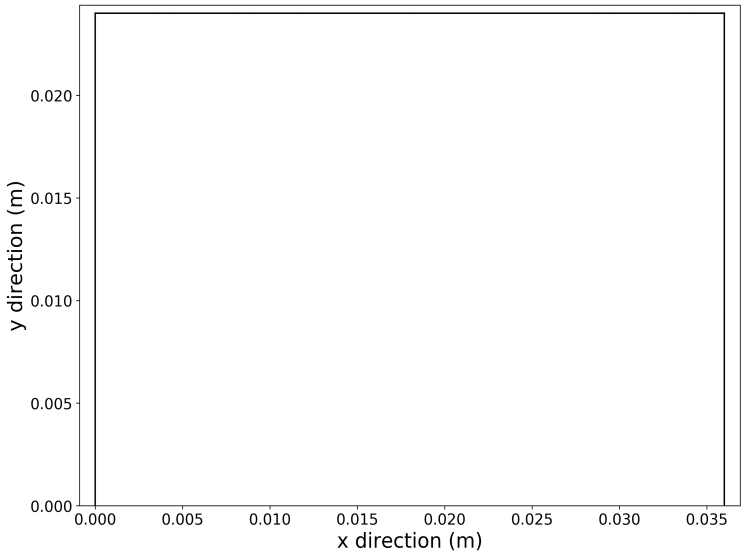


Figure 2.2: Boundary edges of two-dimensional wall

In this dissertation, the "active/inactive" element method is used to build the fixed reference configuration upon which the coupled Lagrangian finite element problem is solved. The new elements in the deformed configuration are added on top of the deformed layers and elements to account for existing deformation. This leads to initial displacement for the iterative solvers when new elements are first added. The element addition in the deformed configuration is left to Section 2.3.2, and the mesh generation in the reference configuration is explained first.

A preliminary example of the evolution of the fixed reference configuration when building a simple two-dimensional wall object is illustrated first. Taking the y-direction as the building direction and the x-direction as the layer direction, the boundary of the wall is shown in Figure 2.2. The wall has a total height of $0.024m$ and a width of $0.036m$. The first step of the printing is to slice the domain into a given number of layers, in this example 60 layers, as shown in Figure 2.3. The layer thickness is set to be uniform in this dissertation and thus is directly determined by the height of the object and the number of layers.

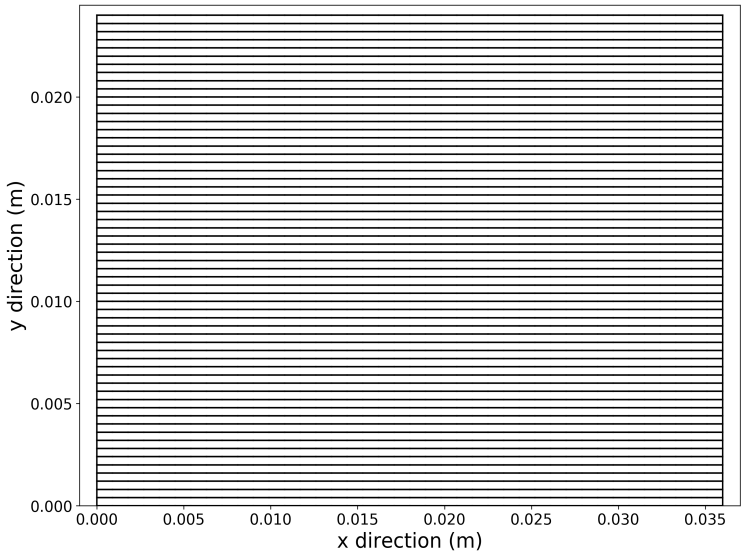
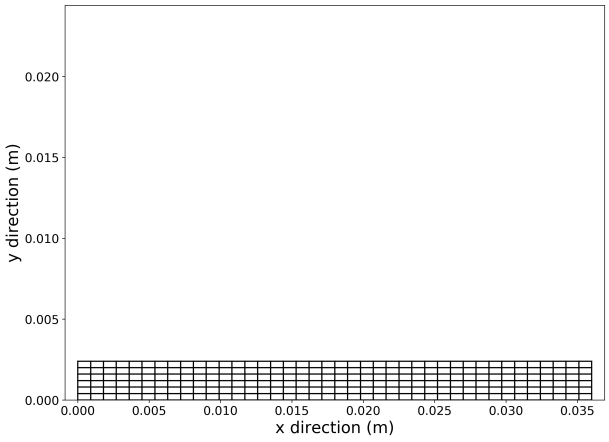


Figure 2.3: Two-dimensional wall sliced

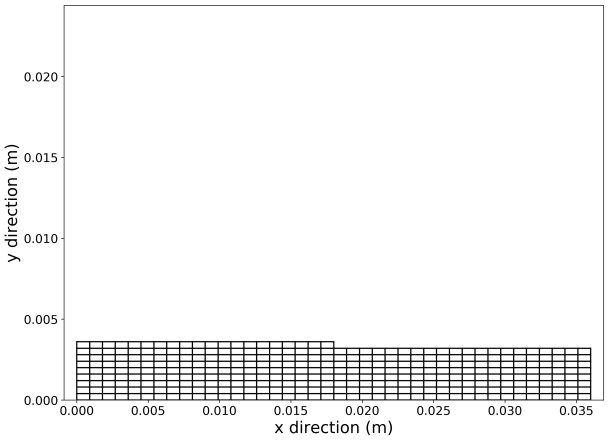
Each layer is then discretized into two-dimensional quadrilateral-shaped elements. The height of each element is determined by the layer thickness while the width relies on the number of elements chosen in the x-direction (here set to 40) and total width of the wall. The rate of the deposition is prescribed, and its value depends on the particular FDM process being simulated. Given the width of the element, the time-step size can be computed so that it takes exactly one time step to lay down one element. The deposition is assumed to always occur from the left end to the right end for each layer. Starting with a base bottom layer, as time proceeds, material, represented by elements, is added to the existing layers. Figures 2.4a to 2.4f show the referential mesh at different time steps from 200 to 1200 during the process of building the two-dimensional wall. The deposition speed is set to $75mm/s$ and corresponds to $0.012s$ time step for each element.

Due to the simple geometry, at each time step a new element is always added. New elements are assumed to become a continuous part of the already printed object and share the existing nodes and edges from previous layers and neighboring elements. For the new nodes, the x-direction coordinates are given by the position of the extrusion head at the time step while the y-direction coordinates are computed based on the area of the new element. Details of the algorithm for adding elements is described in the appendix (Algorithm 5). Since this is the fixed referential mesh, while it evolves with the time steps, no deformation occurs. In the deformed configuration, however, new element addition needs to take into account the displacement of existing layers and elements. This is discussed in Section 2.3.2.

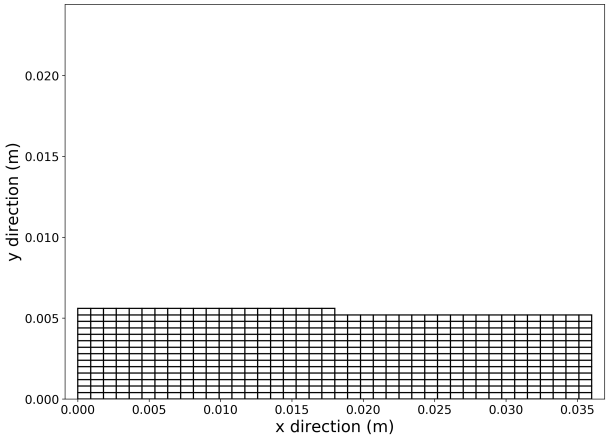
Often times, real applications involve curved surfaces and edges and thus require complex slicing algorithms to approximate the curvature. For this reason, a second preliminary example with more complicated geometry is described next. As shown in Figure 2.5, a two-dimensional wall with a quarter-circular hole at the top-right end is the object. The width and height of the wall are again $0.036m$ and $0.024m$ while the radius of the hole is $0.008m$.



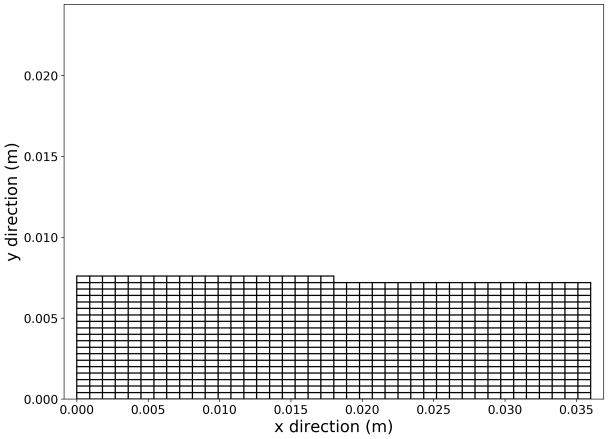
(a) Two-dimensional wall elements at time step 200



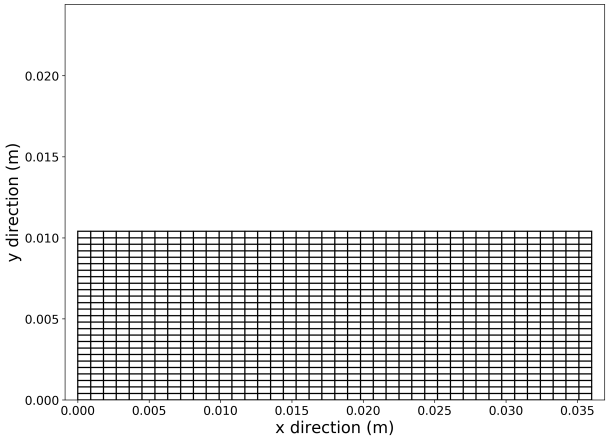
(b) Two-dimensional wall elements at time step 300



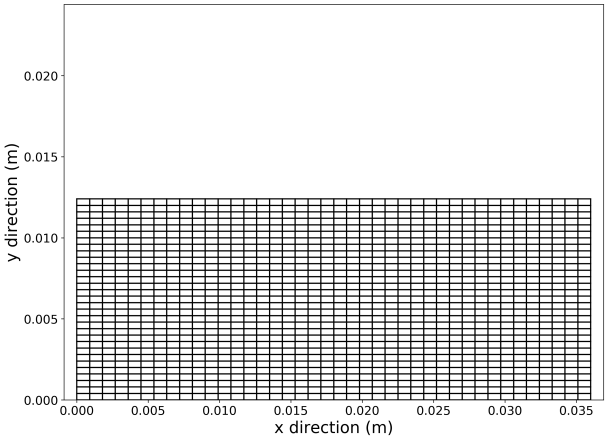
(c) Two-dimensional wall elements at time step 500



(d) Two-dimensional wall elements at time step 700



(e) Two-dimensional wall elements at time step 1000



(f) Two-dimensional wall elements at time step 1200

Figure 2.4: Two-dimensional wall building with elements added at different time steps

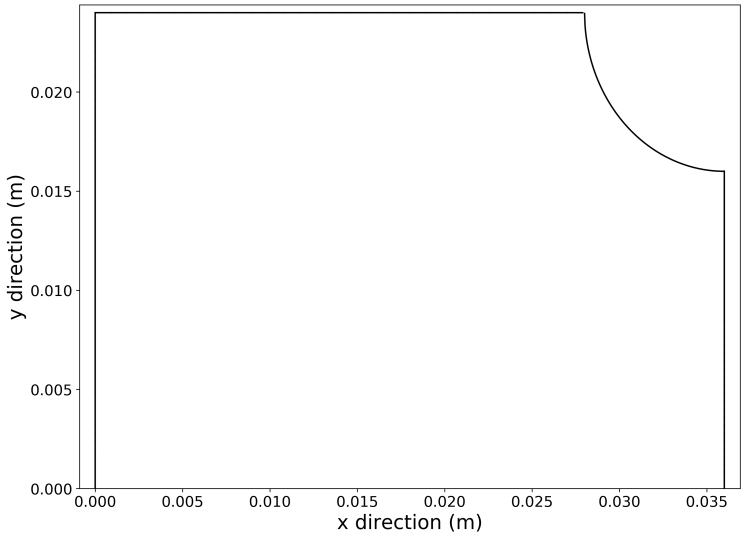


Figure 2.5: Example of two-dimensional wall with curved edges

It is clear that the printed geometry will not be exact around the hole due to the nature of layered manufacturing.

First the wall is sliced into a given number of layers, as in the earlier example. The x-direction ending positions for each layer now are no longer the same at the top part of the wall and are governed by the slicing algorithm, which determines how the curved edges are approximated. The curve approximation leads to geometric error and the slicing algorithm is crucial as to how large such an error would be. Here two choices of slicing algorithms deemed most suitable for the simulation and optimization problems in this dissertation are discussed. Other algorithms can be adapted and fit into the framework as well. For the purpose of effective simulation and optimization, the slicing algorithm adopted should be numerically stable and efficient.

For the description of the slicing algorithm, it is assumed that the curve function is known (here a quarter-circular curve). A naive slicing algorithm approaches the curve from one side with boundary nodes right on the curve, as shown in Figure 2.6.

Here, the x-direction coordinates of the nodes at the boundary can be fully determined by their y-direction coordinates using the curve function. The printed edges remain on the material side of the curve and never cross it. Hence the name 'one-sided' is adopted for this method.

Another potential method is to minimize the area difference enclosed by the curve and the layered step edges which requires more complicated calculations at each element on the boundary. This leads to frequent intersections between the ideal curve and printed edges, as shown in Figure 2.7. In addition, this method uses the coordinate information in both dimensions of the relevant nodes to determine the size and width of the elements on the curved boundary. This method is thus called 'minimum-area'. Once the boundary of the printed structure is determined, each layer is discretized with uniform bilinear quadrilateral

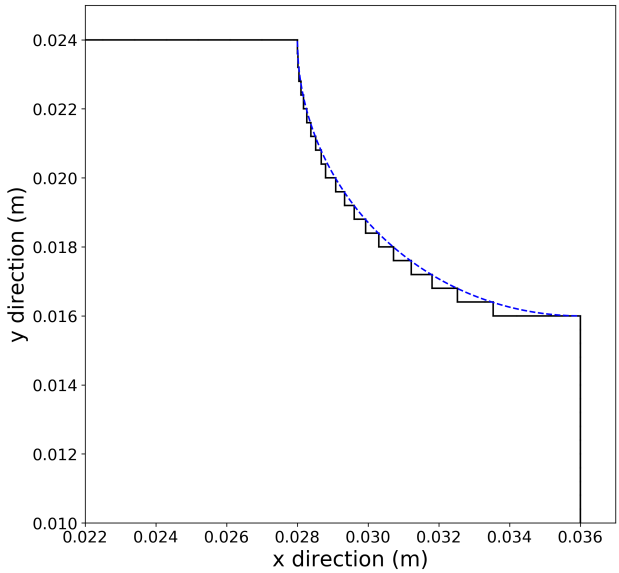


Figure 2.6: One-sided approximation of two-dimensional wall with curved edges

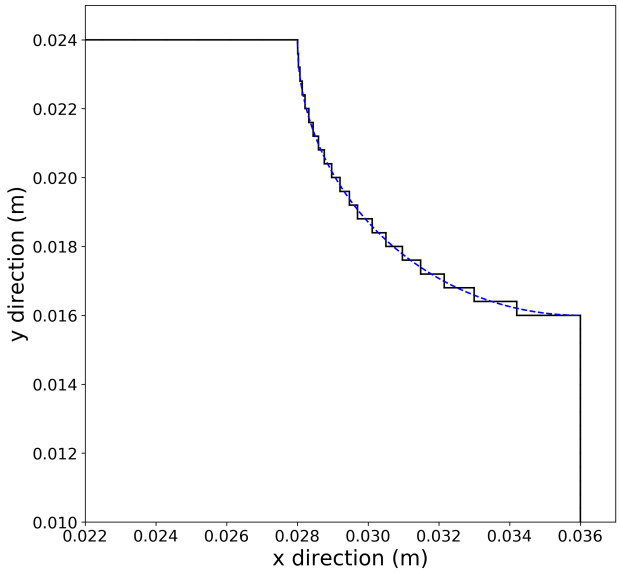


Figure 2.7: Minimum-area approximation of two-dimensional wall with curved edges

elements internally, similar to the solid wall case. At the boundary, elements with different widths are used in order to accurately match it, as can be seen in Figure 2.8. Such a boundary (*vs* interior) element would have the same height but a smaller area compared to a regular one. Hence, it is called a partial element.

To design and implement the two slicing algorithms and corresponding element addition methods, nodal level-set functions are introduced to implicitly represent the geometry and

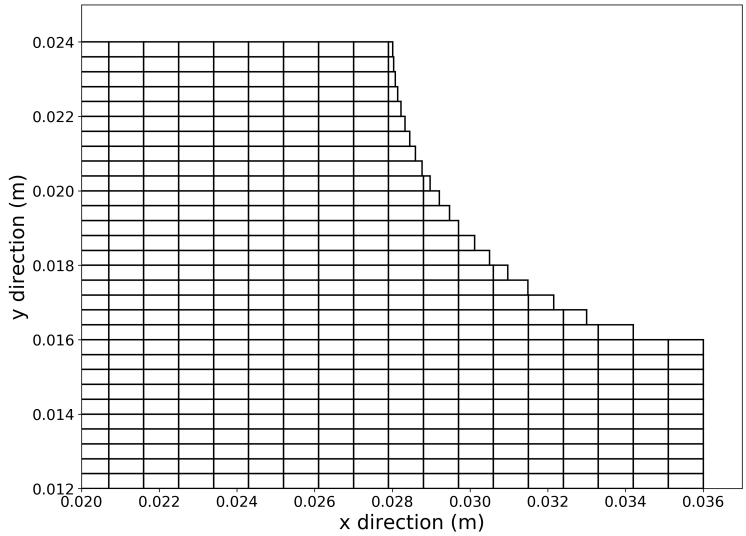


Figure 2.8: Two-dimensional wall elements with curved edges

determine whether an element is being cut through by the curve and hence needs to be adjusted to better fit it. Using level-set functions [68] is an implicit way of describing the boundary of a geometry and has been popular in tackling computational geometry and evolving boundary problems. Suppose the curve being approximated in two-dimension is given by $f(x, y) = 0$ where x and y are coordinates in both directions. A level-set function $\phi(x, y)$ at position (x, y) in space can be defined to determine whether the node is 'inside' f, on f or 'outside' f. The most apparent choice for ϕ is f itself and, in this case, $\phi = 0$ represents the curve. In this dissertation, ϕ is fully given by the specific problem and chosen to be f.

Contrary to the solid wall case where at each time step a new element is added, there now exist time steps with no new element deposition or partial element creation. This is based on the assumption that the extrusion head still moves at constant speed from left to right for each layer with the same time discretization when no element is added. When deciding whether an element should be added, the level-set function values for each of its vertex is calculated. If $\phi > 0$ for all four vertices, the entire element is material and should be added. On the other hand, $\phi < 0$ for all four vertices leads to no element addition. An element is considered to be on the boundary if the values of ϕ at the vertices change sign. If so, slicing algorithms are called to decide whether to add a partial element or a full element or no element. Upon deciding to add a partial element, its width is determined by the bilinear interpolation of level-set function values at the vertices and again the specific slicing algorithm employed to approximate the curved boundary. The procedure is summarized in the appendix (Algorithm 4). As for the slicing algorithms, the two methods discussed earlier are presented in detail in the appendix as well. The one-sided slicing algorithm is given in Algorithm 6 while the minimum-area slicing algorithm is given in Algorithm 7. All twelve viable cases for the level-set function signs at the four vertices of an element referenced in

the algorithms are showed in Figure 2.9 and 2.10.

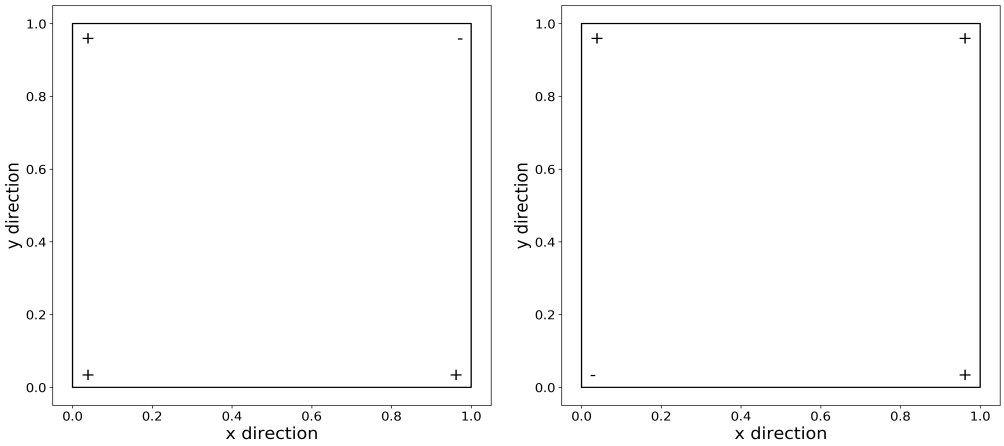
If not constrained, the partial elements can be very narrow relative to the full element (Figure 2.8). These poor aspect-ratio elements when deployed in the simulation problems in Chapter 3 would cause inaccurate deformation and temperature results. Since the model takes into account previous deformation, this would also affect the element addition process and give rise to even larger errors. Therefore an empirical threshold on the minimum area of the partial element is imposed to ensure accurate solutions and convergence of the iterative solver. It is found through testing that avoiding elements with areas smaller than 40% of a full element's provides the desired properties. Elements with area not meeting this threshold are rejected and instead during those time steps no new element are added. Obviously this artificial restriction causes inexact approximation at some parts of the curve. But it is a necessary compromise for the current model. Once the areas of the elements and where they should be deposited for the final printed structure are decided, the referential mesh at different time steps can be constructed. Based on the deposition speed, temporal discretization is designed such that each element width, whether full or partial, takes one corresponding time step to travel.

For the curved geometry, changing the number of layers leads to different approximations of the curves. This is due to the intrinsic macro-scale discreteness in the layered manufacturing process. To be able to later establish a reasonable measure of shape error, how these slicing algorithms would affect the printed geometries is examined in Chapter 4. Here it is mentioned that to better understand how the layer number alters the geometry, the one-sided slicing algorithm is chosen for the simulations.

2.3.2 Deformed mesh

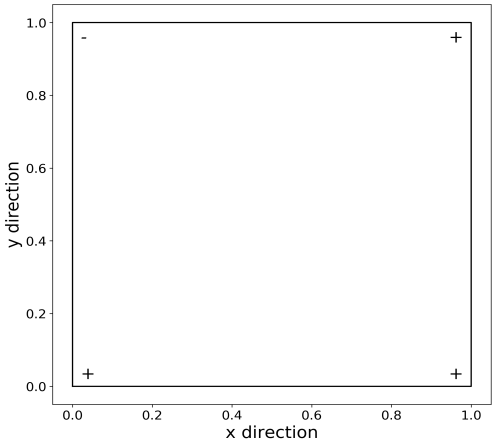
Since a coupled thermomechanical problem is solved at each time step in the model, deformation from previous layers must be taken into consideration when depositing new materials in the current or deformed configuration. The newly added elements will initially inherit displacement due to continuity of the material. The element addition algorithm handles this by conserving area added in two dimensions within a time step and connecting the elements with existing neighboring nodes. The infinitesimal deformation assumption ensures such continuity treatment of the nodes is reasonable. Using the same wall building example shown in Figure 2.4, it is illustrated next how such element addition in the deformed configuration works.

A new element is about to be added at the end of time step t_n on top of the deformed layers as in Figure 2.11. Its rightmost node is a new node. The position of the new node at the beginning of the iteration at time t_{n+1} can be computed by the amount of material printed during that time step and the position of the extrusion head. As shown in Figure 2.12, the star sign marks the x-direction position of the new node of the new element, which is given by the position of the extrusion head. Its y-direction coordinate in the deformed configuration is calculated based on the known area of the new element and the deformed coordinates of

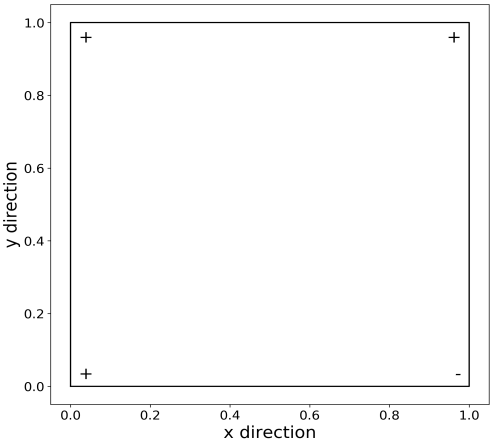


(a) case 1

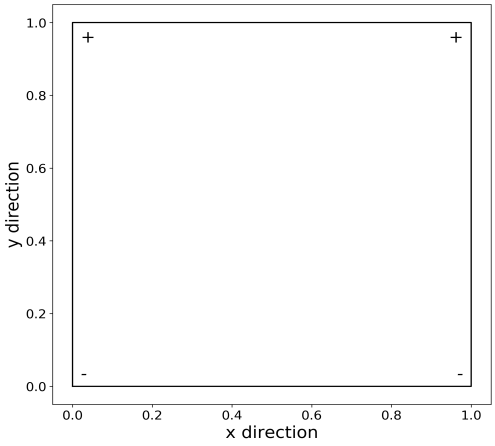
(b) case 2



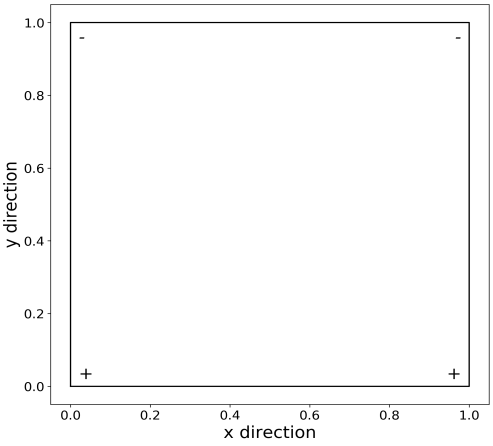
(c) case 3



(d) case 4

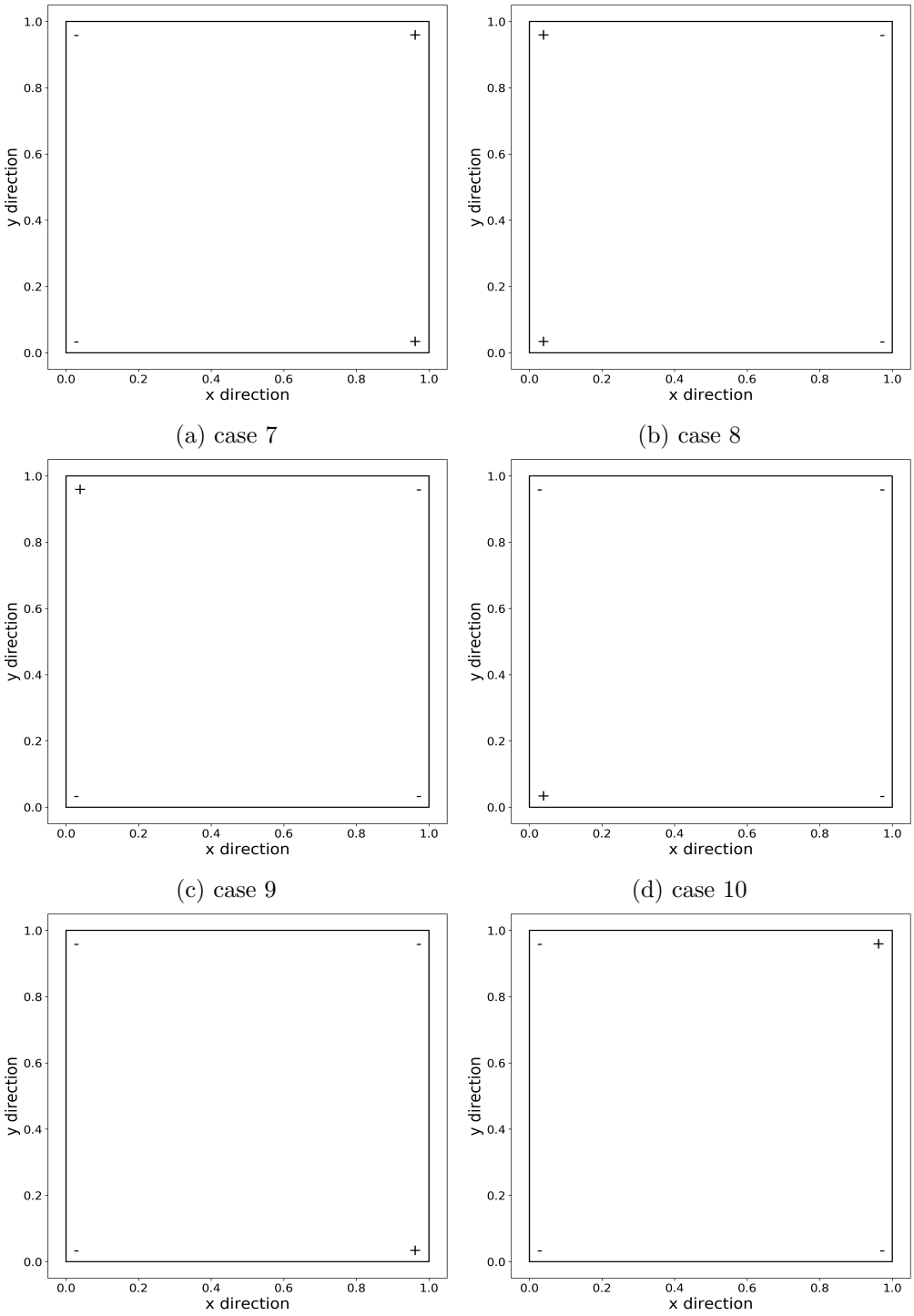


(e) case 5



(f) case 6

Figure 2.9: Different cases for level-set function values at vertices of the element 1



(a) case 7

(b) case 8

(c) case 9

(d) case 10

(e) case 11

(f) case 12

Figure 2.10: Different cases for level-set function values at vertices of the element 2

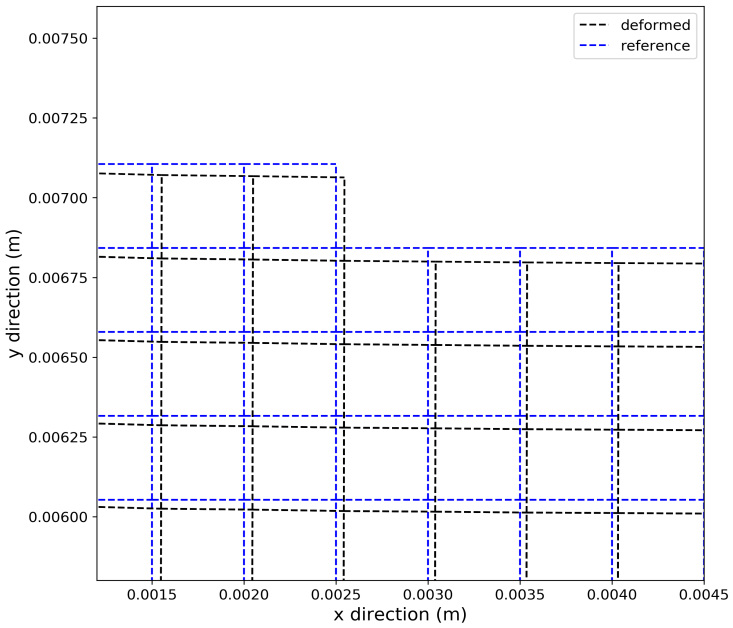


Figure 2.11: Deformed and reference elements at t_n

existing nodes (Figure 2.13). The initial displacement of the new node with respect to the reference configuration is evident.

However, it is reasonable to assume zero initial stress and strain when a new element is first added which can then serve as the reference state for the element satisfying Equation (2.30). To address this, the initial displacement and temperature values are stored as 'history variables'. Figure 2.14 shows the zoomed-in view of the newly added element where the arrows mark the initial displacement at the vertices. The displacement history variables are subtracted from the nodal displacement values when computing strain on element levels and temperature history values are used as reference temperatures for each element as in Equation (2.82).

$$\begin{aligned} \boldsymbol{u}^e &= \boldsymbol{u} - \boldsymbol{u}^{his} \\ \boldsymbol{\theta}_0^e &= \boldsymbol{\theta}^{his}. \end{aligned} \tag{2.82}$$

Here, $\boldsymbol{u}^{his}$ and $\boldsymbol{\theta}^{his}$ denote the displacement and temperature history variables. The element-level computations use $\boldsymbol{u}^e$ for strain as mentioned. This treatment also allows the model to capture large accumulated displacement but maintain small strain, in line with the infinitesimal strain assumption. The history variable vectors for each element in matrix form are

$$\left[\boldsymbol{u}^{his,e}\right] = \left[\boldsymbol{u}^e_{1,t_{0,e}} \quad \boldsymbol{u}^e_{2,t_{0,e}} \quad \dots \quad \boldsymbol{u}^e_{nen,t_{0,e}}\right], \qquad \left[\boldsymbol{\theta}^{his,e}\right] = \left[\theta^e_{1,t_{0,e}} \quad \theta^e_{2,t_{0,e}} \quad \dots \quad \theta^e_{nen,t_{0,e}}\right], \tag{2.83}$$

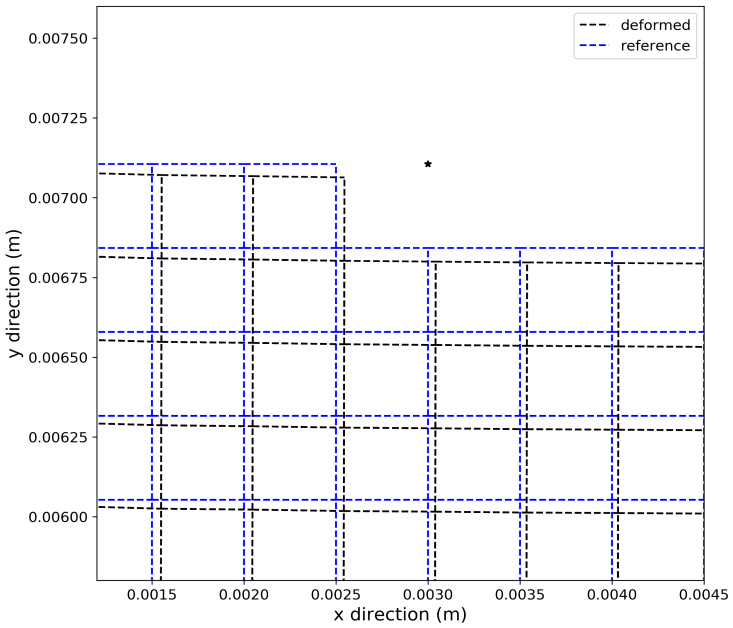


Figure 2.12: Printer head position marked at t_{n+1}

where $\boldsymbol{u}^e_{i,t_{0,e}}$ is the displacement of the ith node of the element, nen is the number of nodes per element and $t_{0,e}$ denotes the time when the element is first added. Likewise $\theta^e_{i,t_{0,e}}$ is the average temperature of the element when it is first created.

Although the history variables remain the same for every element throughout the simulation, their sizes change due to the evolving domain. As a result, at time step t_n, the element-level history variables are denoted as $\boldsymbol{u}^{his,e}_n$ and $\boldsymbol{\theta}^{his,e}_n$ for displacement and temperature, respectively. The global history variable matrices $\left[\boldsymbol{u}^{his}_n\right]$ and $\left[\boldsymbol{\theta}^{his}_n\right]$ are stored as two-dimensional arrays with the dimension of $\{nel, ndim \times nen\}$ and $\{nel, nen\}$, where nel is the number of elements, $ndim$ is the dimension of the displacement variable.

To apply this framework to specific simulations, typical material properties used in FDM and an appropriate constitutive law need to be chosen. Specifically the coefficients in the constitutive laws for the Helmholtz free energy need to be consistent with experimental data to present meaningful simulation results. Moreover, numerical issues such as computation cost for a large scale problem requires careful design of the implementation. These are discussed in Chapter 3.

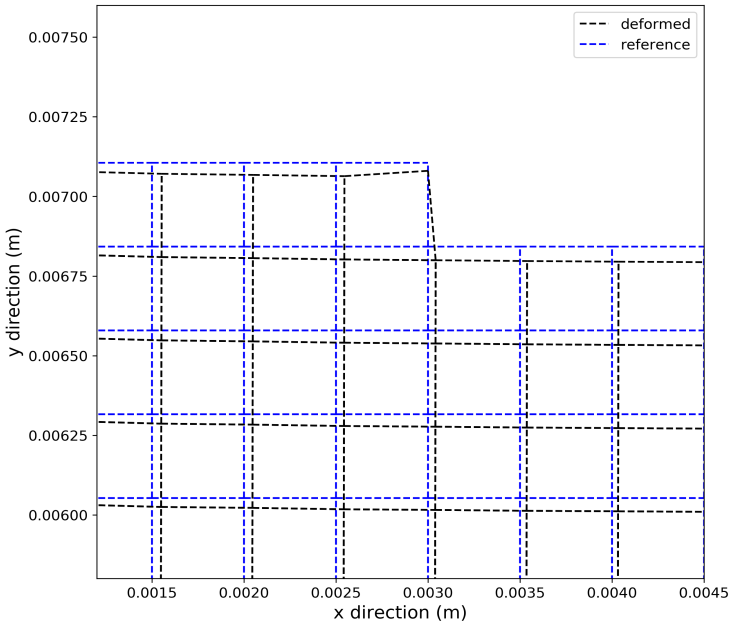


Figure 2.13: Initial positions of new element at t_{n+1}

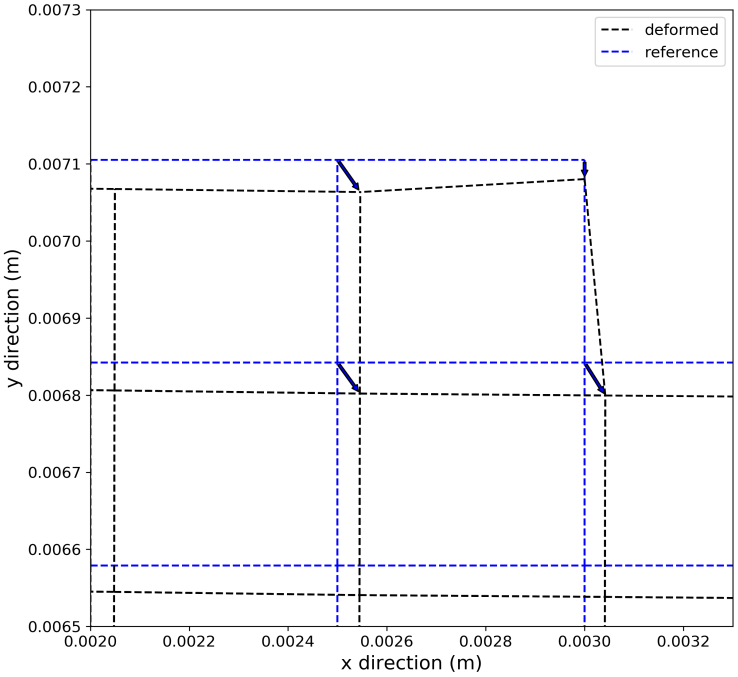


Figure 2.14: Initial deformation $\boldsymbol{u}_{n+1}^{his}$ marked by arrows at t_{n+1}

Chapter 3

Simulation Setup and Results

3.1 Introduction

The materials used in FDM are typically thermoplastics which upon printing are transitioned into molten state in a liquefier head at a high temperature (around 500K) and then deposited through a nozzle layer upon layer at given positions determined by the printing machine software. Upon deposition, the material cools down rapidly and solidifies to form the desired geometries [69, 70, 71]. Printing pattern can be designed for many machines so that the material gets deposited in a particular order. For example, in two dimensions printing can go from the left end to the right end of the object or *vice versa*. Support materials are often necessary when building parts such as a ball that do not have large bottom support. These support materials would be removed when printing is done.

Some studies have developed mathematical and computational models for the extrusion process, which share similarities with injection molding. Bellini *et al.* [72] developed a mathematical model to analyze the liquefier dynamics in order to establish control strategies for flows during the extrusion process of the material. Crockett *et al.* [73] investigated an analytical model for the liquid-solid transition in stereodeposition. Here a flow rule for viscosity was used to establish differential equations for bead width and contact angle. Solidification clearly impacts the material properties as well as the geometry but much remains to be understood about the transition dynamics. Venkataraman *et al.* [74] developed a methodology for characterizing the compressive mechanical properties of fused deposition of ceramics filament feedstocks to avoid potential buckling during the extrusion process of fused deposition of ceramics. Ramanath *et al.* [75] researched the melt flow behavior of Poly-ϵ-caprolactone (PCL) in the melt flow channel of the FDM, in particular the thermal behavior, pressure drop and velocity gradient. Both mathematical and simulation models have been developed and results concur that pressure drop depends on the nozzle diameter and angle variation along the melt flow channel. These parameters can be tuned to satisfy design objectives.

The simulation model developed in Chapter 2 in its current capacity does not account for

the extrusion process and how it might change material properties. Nevertheless, it is worth exploring in the future for more comprehensive models. In this chapter, the constitutive law for the Helmholtz free energy is specified and additional numerical issues such as the condition number of the algebraic system are addressed. The model is then deployed on simulations to understand the displacement and temperature change during the FDM process as well as the coupling effect between them.

3.2 Constitutive Law

In Section 2.2 the general temperature-dependent linear elastic constitutive law for the material used is introduced. A lot of research has been done on modeling thermoplastics, leading to the models able to capture the entire stress-strain curve until failure. It is stressed that the best choice for a specific constitutive model should be based on the problem and specific material used and ideally verified with experimental data [63, 76, 77]. The model chosen here is used as an example to demonstrate the coupled effect of temperature and displacement on the quantities of interest. Due to the assumption of infinitesimal deformation, its use is limited. However, results with this model are able to illustrate the qualitative features of the FDM-manufactured objects and provide insights into the optimization procedure within a reasonable range of displacement gradient and temperature. To that end, following the derivations from [66] with the addition of infinitesimal deformation assumption, a model that encompasses more complex dependence of elasticity tensor on temperature is proposed. The function $f(\theta)\mathbb{C}_0$ in Equation (2.41) takes the form

$$f(\theta)\,\mathbb{C}_0 = \tilde{f}(\theta)\,\mathbb{C}_{0,m} = \left[b\left(\frac{\theta-\theta_g}{\theta_{0,m}-\theta_g}\right)^a + \left(\frac{1}{\theta_{0,m}} - \frac{ab}{\theta_{0,m}-\theta_g}\right)\theta - b + \frac{ab\theta_0}{\theta_{0,m}-\theta_g} + \frac{c}{(\theta_{0,m}-\theta_m)^2}(\theta-\theta_0)^2 \right] \mathbb{C}_{0,m} \tag{3.1}$$

Here, $a, b, c, \theta_g, \theta_{0,m}, \theta_m$ are coefficients of the constitutive law that can be estimated through experimention. In physical terms, θ_g can be interpreted as the glass transition temperature. In order to expand the flexibility of the model and better correlate experimental results, the reference temperature $\theta_{0,m}$ used to compute the elasticity tensor, is allowed to differ from the temperature for the reference state θ_0, and its corresponding elasticity tensor is denoted as $\mathbb{C}_{0,m}$. Notice that, mandated by consistency to isothermal elasticity, $\tilde{f}(\theta_{0,m}) = 1$. The elasticity tensor is assumed to be isotropic and computed based on Equation (2.40) with μ and λ the elastic moduli at $\theta_{0,m}$. The constitutive model is suitable for the case of infinitesimal deformation and large temperature gradient. It is essential for this constitutive law to have a good correlation with experiments in terms of elastic modulus as this is central to how the mechanical response changes with temperature. The elastic modulus $E(\theta)$ here is defined as the first row and column component of the elasticity tensor $\mathbb{C}(\theta)_{1,1}$. Thus, the aforementioned coefficients are chosen and compared with published experimental data

of a polycarbonate material used in FDM printing by Stratasys [78]. Figure 3.1 shows the experimental data points from [78] and the curve with manually chosen coefficients based on trial and error within the range of temperature used. At a relatively low value of $325K$ the elastic modulus decreases slightly with temperature. It then experiences a sudden drop around the glass transition temperature followed by a plateau as the temperature continues to increase. The constitutive model is not accurate when the temperature is relatively low, but can track the behavior closely once the temperature moves closer to the glass transition temperature. The temperature range for the simulation problems is set to be $370 - 500K$. Therefore the model generates realistic elastic modulus and thus mechanical response. The specific heat which also varies with temperature based on Equation (2.43) gives negligible changes against temperature.

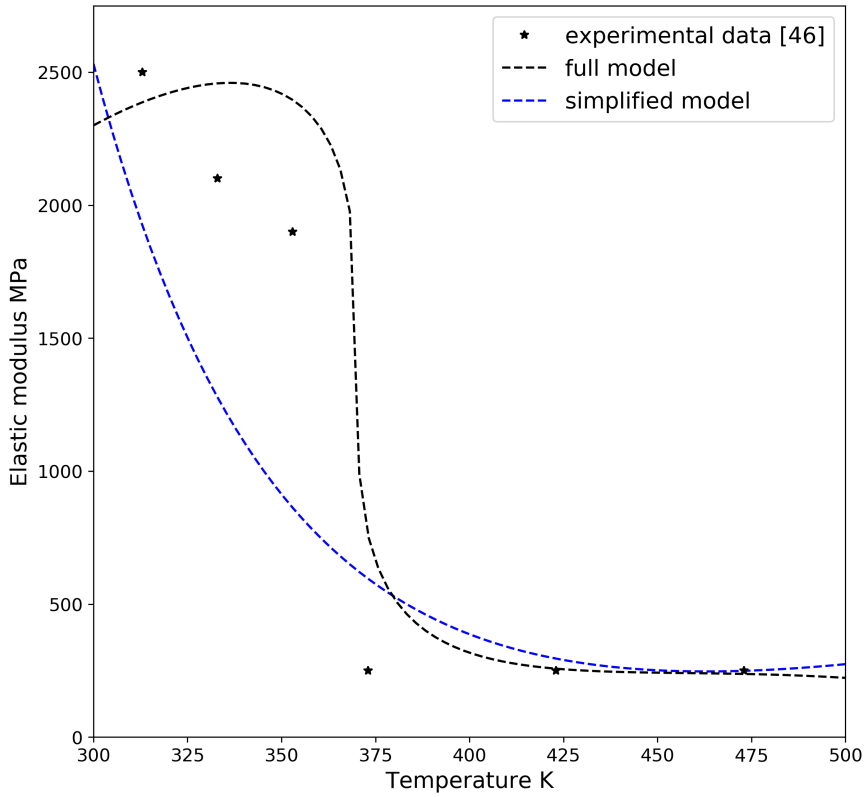


Figure 3.1: Elastic modulus *vs* temperature

The material coefficients are listed in Table 3.1. The elastic moduli μ and λ can be computed through the elastic modulus $E_{0,m}$ and the choice of Poisson's ratio 0.35.

Material coefficients	Numerical values	Unit
a	0.32	N/A
b	0.82	N/A
c	-0.5	N/A
$\theta_{0,m}$	300	K
θ_m	500	K
θ_g	375	K
$E_{0,m}$	2300	MPa

Table 3.1: Material coefficients for the constitutive law

A simplified constitutive model that captures the essence of the correlation between elastic modulus and temperature while produces realistic elastic modulus is proposed next. Its form is given in Equation (3.2)

$$\begin{aligned}\psi(\boldsymbol{\epsilon},\theta) = \tilde{\psi}(\boldsymbol{\epsilon},\theta) = &\left[b\left(\frac{\theta}{\theta_{0,m}}\right)^a + b\,(a-1) + (1-ab)\,\frac{\theta}{\theta_{0,m}}\right]\frac{1}{2}\boldsymbol{\epsilon}\cdot\mathbb{C}_{0,m}\boldsymbol{\epsilon} \\ &+ \kappa_0\alpha_0\ln(1+tr\boldsymbol{\epsilon})(\theta_0-\theta) + \bar{c}\left(\theta-\theta_0-\theta\ln\frac{\theta}{\theta_0}\right).\end{aligned} \tag{3.2}$$

The result with the chosen coefficients, listed in Table 3.2, is plotted in Figure 3.1 as well. Due to its easy implementation and ability to capture the general modulus change, the simplified constitutive law is employed in the simulation problems.

Material coefficients	Numerical values	Unit
a	-5.5	N/A
b	1.	N/A
$\theta_{0,m}$	475	K
$E_{0,m}$	250	MPA

Table 3.2: Material coefficients for the simplified constitutive law

In addition, the remaining material properties used in the simulation are listed in Table 3.3.

Material properties	Numerical values	Unit
k	0.3	W/(mK)
h	30	$W/(m^2K)$
$\bar{c}$	1670	$J/(Kkg)$
ρ_0	1060	kg/m^3
α	10^{-4}	1/K

Table 3.3: Material properties for the simulation model

3.3 Numerical Implementation

Dirichlet boundary conditions are imposed by fixing the corresponding displacement and temperature degrees of freedom at prescribed values. Free surface conditions are equivalent to zero traction and make the equivalent nodal forces equal to zero. Convection heat transfer is implemented as a boundary condition according to Equation (2.62). The convection coefficient h is assumed to be constant throughout the process. However, depending on the cooling mechanism being simulated, free convection and forced convection may lead to dramatically different values for h [79]. In the literature, convection coefficients often come in the order of $10W/m^2K$ [14, 80].

The printing of a layer is treated as a nonstop process in time to emulate the real conditions. As mentioned in Section 2.3.1 and 2.3.2, at the beginning of the deposition, the element addition algorithm decides whether and where to add a full element, a partial element or no element on the reference configuration. The discrete system of PDEs are constructed on the referential mesh at the current time step. It is then solved iteratively for the coupled displacement and temperature given an initial guess for the solution. If a new element is added, its initial displacement in the deformed configuration, as explained in Section 2.3.2, serves as the starting point for the iteration. If no new element is added, solutions from the last time step are used. In this dissertation, the printing is always directed from the left end to the right end of a layer. Once a layer is finished, the extrusion head needs to move back to the starting point of the next layer. A small time interval, around one-fifth of the time for printing one layer, is added between the end of one layer and the beginning of the next to represent this motion. This time interval is discretized with the same time step size. The hanging nodes created from partial elements are treated as redundant degrees of freedom with their values being a linear interpolation of the values at the vertices of the full edge they lie on. Figure 3.2 shows an example of a hanging node and its two neighboring nodes. Let $\boldsymbol{x}_h$, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denote the coordinates of the hanging node, node 1 and node 2, respectively. The length percentage ξ is defined and obtained through

$$\boldsymbol{x} = \xi \boldsymbol{x}_1 + (1 - \xi)\boldsymbol{x}_2. \tag{3.3}$$

This ratio ξ is then used to compute quantities other than length. Assuming linear interpolation for the values at the hanging node, for a quantity f,

$$f(\boldsymbol{x}_h) = \xi f(\boldsymbol{x}_1) + (1 - \xi) f(\boldsymbol{x}_2). \tag{3.4}$$

The element-level calculations for the partial elements are the same as the full elements. Equation (3.4) is imposed similar to Dirichlet boundary conditions when solving the eventual linear system. The degrees of freedom of the hanging nodes are removed, just as the ones with fixed displacement and temperature. Boundary conditions are active on the partial edges that are exposed to them.

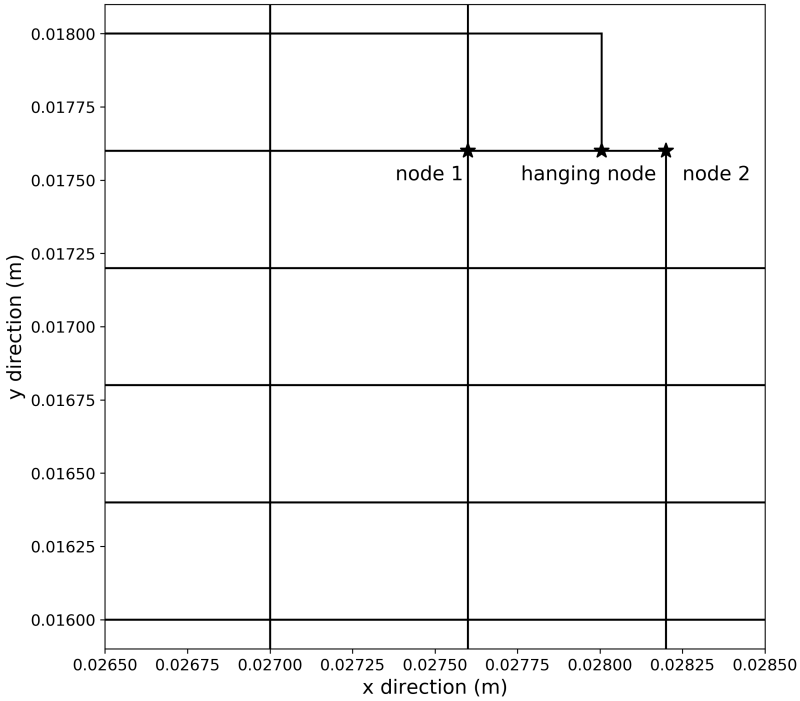


Figure 3.2: Hanging node and its surrounding nodes

As mentioned before, reducing computation time for simulation of additive manufacturing is a major focus of the current research effort [10]. Since our model is presently confined to two-dimension problems, runtime has not reached a critical point where optimization becomes unfeasible. Nevertheless, steps are taken to help reduce computing time and better organize the code's data structure. The simulation model is implemented in both Python and C++ with the C++ version running all problems as it is substantially faster. Element and node additions do not require destruction and reconstruction of new arrays every time. Instead, vectors and matrices are pre-allocated at the beginning with sufficient size to cover the final geometry and avoid expensive memory operations. The linear solver package UMFPACK [81] is used to solve the final asymmetric linear system. Effective solves require sparse matrices generated from the model to be stored in a compressed column data structure [82]. The positions of existing entries in the global sparse matrices stored in compressed column form do not remain the same as new entries from new elements spread out across rows and columns and therefore change the column-wise order. Consequently, in addition to keeping track of the length of current elements and nodes, the sorting indices of element-level matrix entries that would put them into global column-wise ascending order are also stored and updated whenever necessary. This allows us to avoid the expensive sorting operations over the entire global matrices.

Typically, nodal temperature drops from approximately $500K$ (close to melting point coming out of extrusion head) to $300–373K$ ambient temperature, resulting in a $200K$ change in value. At the same time, a small length scale problem, as the ones shown in

this dissertation, might involve an object with the length and width on the scale of $1cm$, and with simulated layer thickness of around $0.3mm$. This, while satisfying the infinitesimal deformation assumption, leads to a displacement in the range of $10^{-7}m$. Since the coupled displacement and temperature vectors are solved together, the huge order of magnitude difference between them might induce a large condition number and accumulate numerical errors as more layers are printed. To improve the condition number, the length dimension is scaled by an appropriate number. Figure 3.3 shows an example where only two elements are added on top of an existing base layer. We calculate its condition number of the eventual sparse matrix $\boldsymbol{A}$ such that $\boldsymbol{AU} = \boldsymbol{b}$, where $\boldsymbol{U} = [\boldsymbol{u} \quad \boldsymbol{\theta}]^T$ at the last time step. Various values of the scaling factor $\tilde{c}$ are chosen so that $\hat{\boldsymbol{X}} = \frac{1}{\tilde{c}}\boldsymbol{X}$ where $\hat{\boldsymbol{X}}$ is the scaled position vectors. The displacement is scaled accordingly as well by $\tilde{c}\hat{\boldsymbol{u}} = \boldsymbol{u}$ where $\hat{\boldsymbol{u}}$ is the scaled displacement. This amounts to selective diagonal preconditioning for the displacement degrees of freedom.

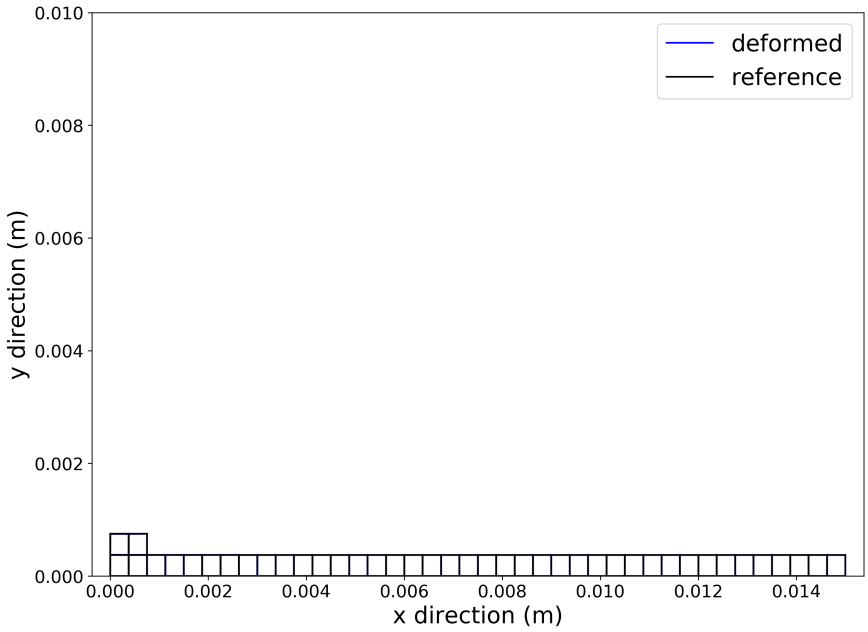


Figure 3.3: Test example for condition number

Figure 3.4 illustrates how the condition number changes with the scaling factor. As expected, a $\tilde{c} < 1$ can help lower the condition number as it brings the length scale closer to the temperature one. Yet if the scaling factor c is too small, such benefits disappear. Hence 10^{-4} is picked for the following simulations.

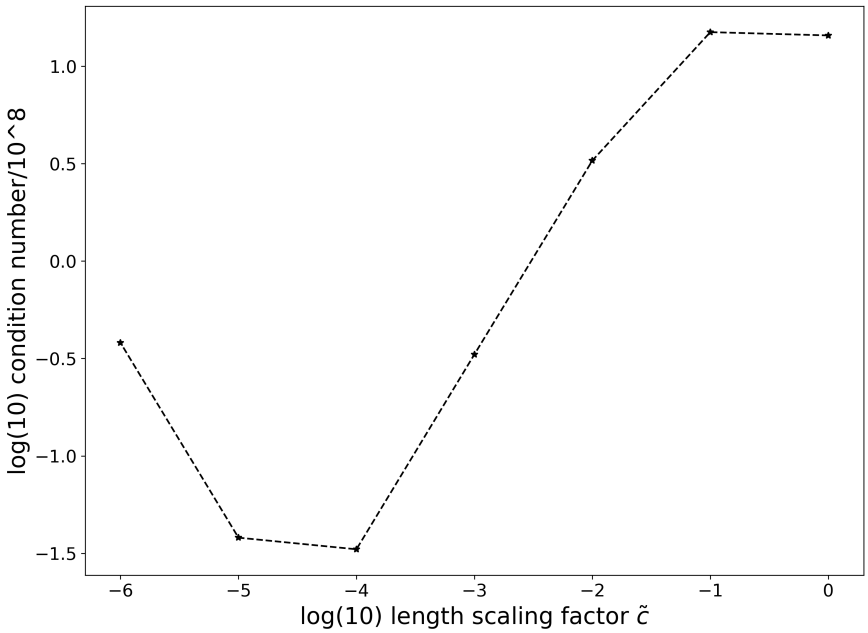


Figure 3.4: Condition number *vs* length scaling

3.4 Simulation Problems and Results

3.4.1 Problem Description and Mesh Test in the x-direction

A typical simulation problem with the proposed model and material constitutive law is described next. Convergence tests with respect to mesh size based on the problem are conducted to ensure accuracy of the model. The object to be printed is a two-dimensional wall, whose shape is given as the reference configuration in Figure 3.5 with width of $20mm$ and height of $10mm$. The wall is first discretized by 40 elements in the x-direction while the number of elements in the y-direction is dictated by the layer thickness and chosen to be 30. A typical layer thickness for FDM can be between $0.25 - 0.4mm$, which the height of an element satisfies. The bottom of the wall is subjected to Dirichlet boundary conditions with zero nodal displacement and constant temperature at ambient value of $373K$. The three other sides of the wall are free of traction and under convection heat transfer conditions with θ_∞ at ambient temperature. The initial temperature of the hot material is set at $500K$ of depostion. A base layer at the bottom whose reference temperature is at the ambient temperature is assumed to be present at the beginning of the simulation with no initial stress or strain. The time step size for one element is set to $0.006s$ and therefore the equivalent printing speed is $83mm/s$. The time interval between layers is set to $0.036s$, roughly one-sixth of the time it takes to print one layer. The simulation stops when the last element is added and the results at that time are plotted from Figures 3.5 to 3.7b. Figure 3.5 plots the deformed and referential boundary at the last time step, where the displacement

is magnified by 10 times to better illustrate the result. Small deformation can be observed while the general shape of the deformed configuration is close to the desired one.

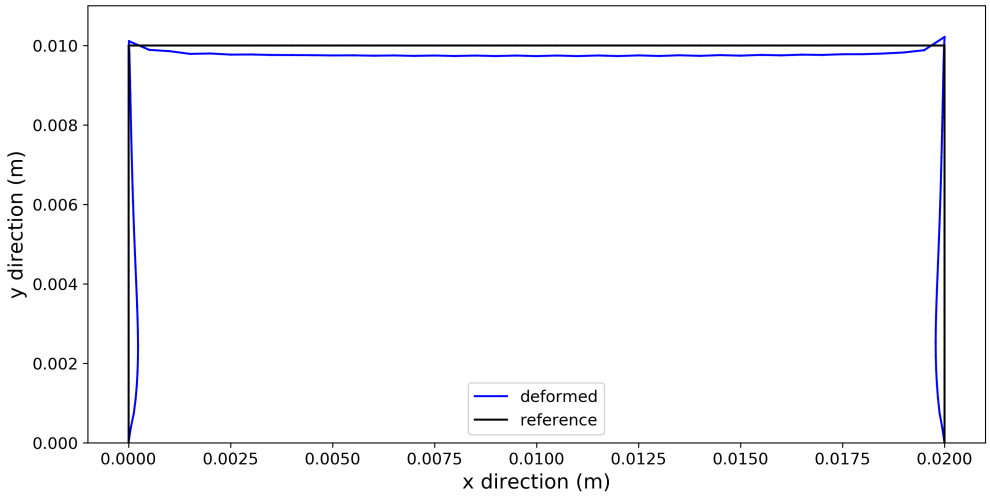


Figure 3.5: Deformed and reference shape of the two-dimensional wall with 10 times the actual displacement

The temperature filled contour after the last element is added is shown in Figure 3.6 where a clear pattern of gradually dropping temperature as height decreases is evident. Based on the simulation condition and stopping time, a large part of the wall remains at a fairly high temperature.

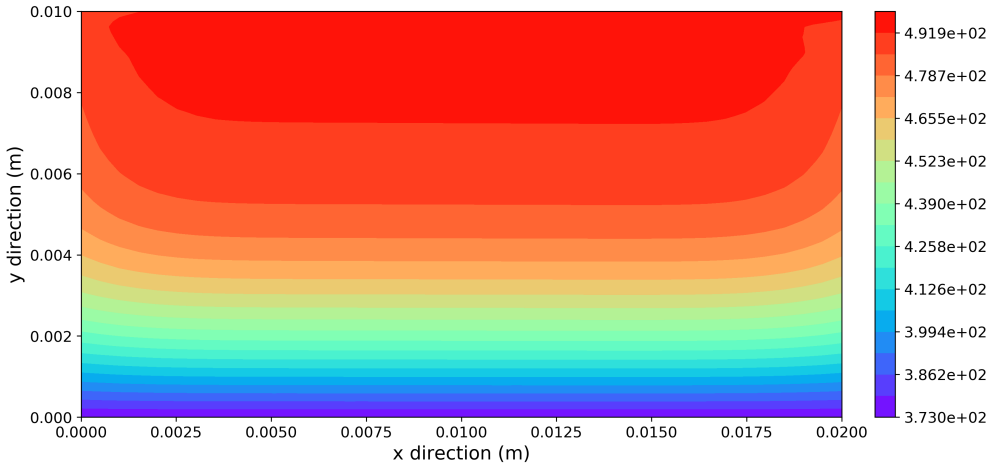
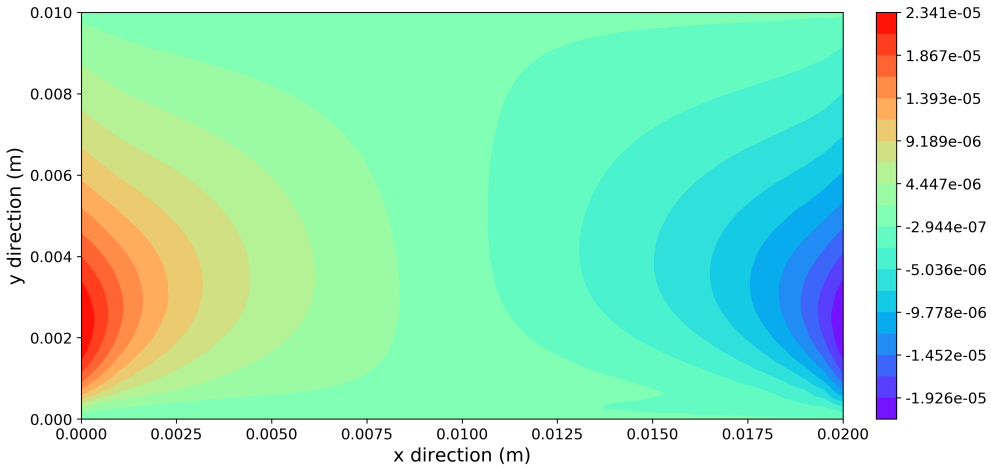


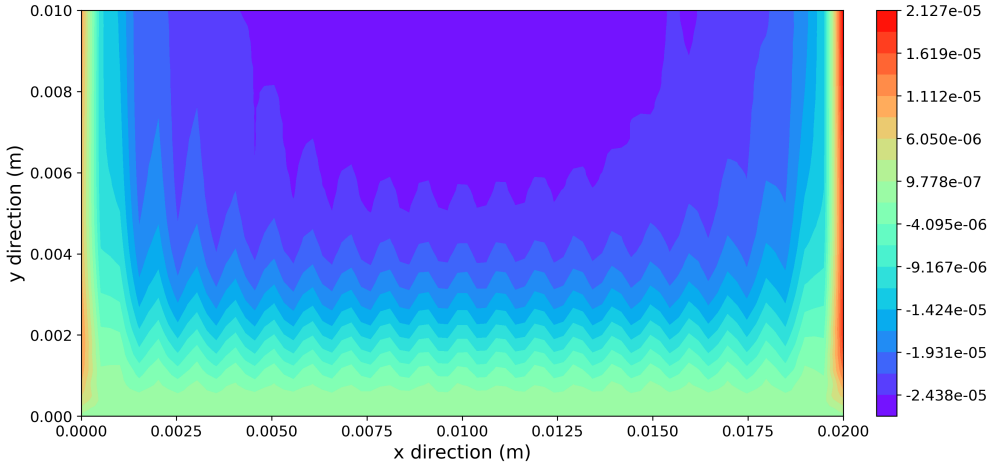
Figure 3.6: Temperature filled contour of the wall

Figure 3.7a illustrates the displacement in the x-direction. Positive displacement occurs on the left side of the wall while on the right side negative displacement can be seen. The displacement decreases to zero towards the bottom as it is fixed. The displacement pattern

in the y-direction is shown in Figure 3.7b. Negative y-direction displacement is present in most part of the object with the exception of the lower part of the two side edges. The cooling during the printing process explains the overall contraction, while the fixed bottom boundary and new elements inheriting displacement from existing layers and elements cause the small positive y-direction displacement.



(a) Displacement in x direction filled contour of the wall



(b) Displacement in y direction filled contour of the wall

Figure 3.7: Displacement filled contour at the end of element addition of the wall

This is demonstrated in the zoomed-in view of the relevant elements in Figure 3.8. First, for the bottom base layer, it has zero initial displacement and is already at ambient temperature. Since it does not experience cooling but rather small amount of heating due to the hot material added on top of it, the base layer experiences no reduction in area. The shrinking of elements above it towards the center of the wall in x-direction forces upper nodes of the base layer to also have positive x-direction displacement on the left side and

negative on the right. Thus its height at the two sides is pushed up to exhibit minor positive y-displacement as the bottom nodes are fixed. Second, beyond the base layer, the leftmost elements shrinking in area due to cooling leads to decrease in their edge lengths in the x-direction. In order to preserve both material continuity and mass/area, new elements added on top initially gain positive y-displacement. Such positive displacement competes with the overall reduction in height due to cooling and is inherited by layers up above. At the current time step, the wall remains mostly at elevated temperature from new elements with high temperature being added continuously. It is worth noting that once the temperature cools down sufficiently, such positive y-displacement vanish at the top part of the side edges as observed in Figure 3.14 and 3.29.

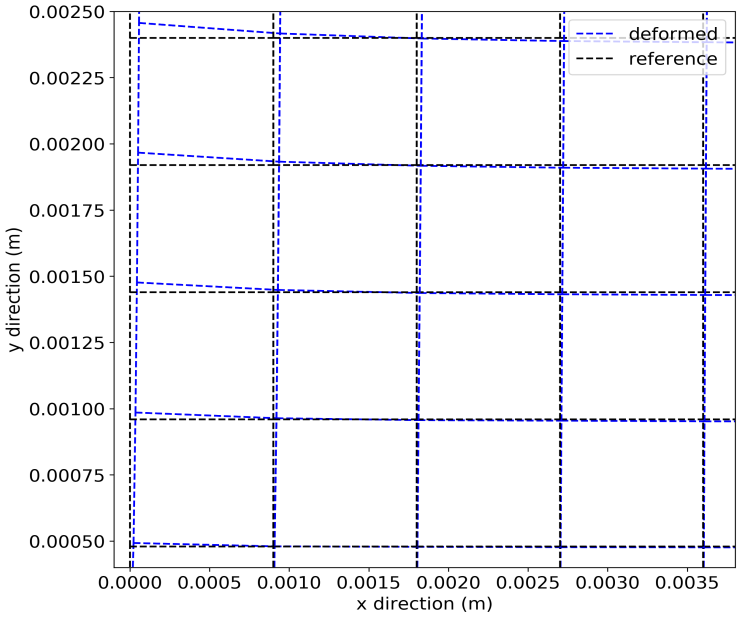


Figure 3.8: Displacement at the bottom leftmost part of the wall

The displacement and temperature distribution pattern matches intuitive considerations as the object contracts with decreasing temperature. The overall shrinking in area agrees with published results in literature [11, 58, 25] as well.

A convergence test on mesh size is conducted to verify the model. In this case, however, changing the mesh size in the y-direction would result in a different dynamic problem with different number of layers. Therefore, the same problem described above is run with varying mesh size in the x-direction to show consistent results can be reached with the choice of number of elements. For consistency, overall the time to print one layer is kept the same and the layer thickness is unchanged as well. Table 3.4 shows the result of maximum temperature and maximum contraction of the top surface when the last element is added. As can be seen in the Figures 3.9, 3.10, and the data from the table, these values changes are minimal as

mesh is refined. Therefore our pick of number of elements in x direction $n_x = 40$ provides good and consistent results.

$\Delta x(m)$	max T (K)	change in max T	max contr (m)	change in max contr
$6.67 * 10^{-4}$	498.203	NA	-2.6558×10^{-5}	NA
$5.00 * 10^{-4}$	498.483	0.056%	-2.6920×10^{-5}	1.3%
$4.00 * 10^{-4}$	498.662	0.036%	-2.7471×10^{-5}	2.0%
$3.33 * 10^{-4}$	498.790	0.026%	-2.7178×10^{-5}	0.7%

Table 3.4: Convergence study of a two-dimensional wall problem

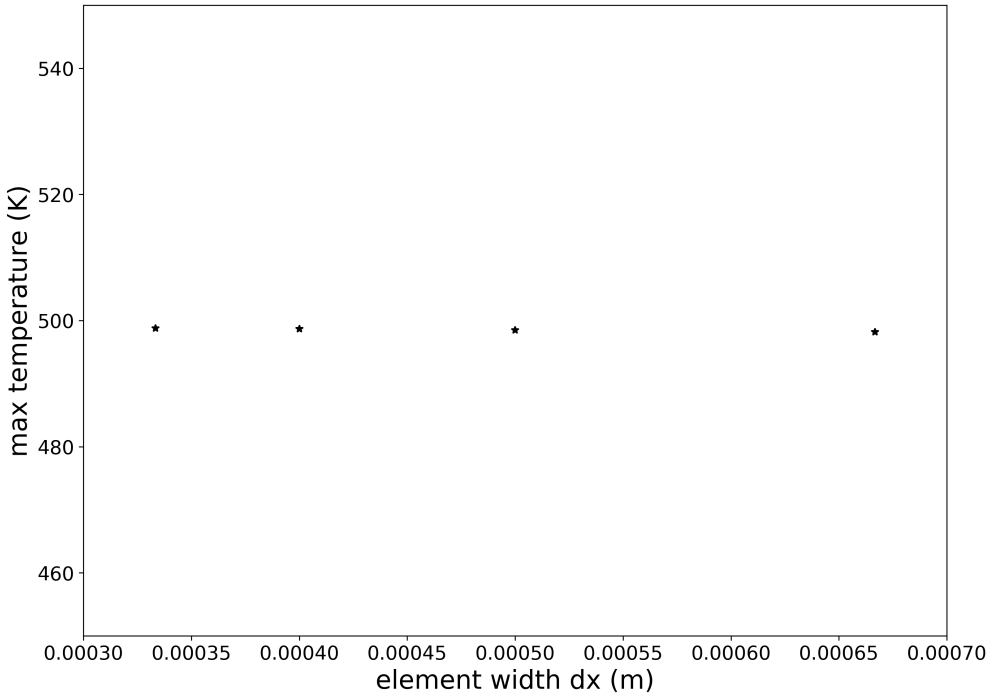


Figure 3.9: Maximum temperature *vs* element width in the x-direction in the reference configuration

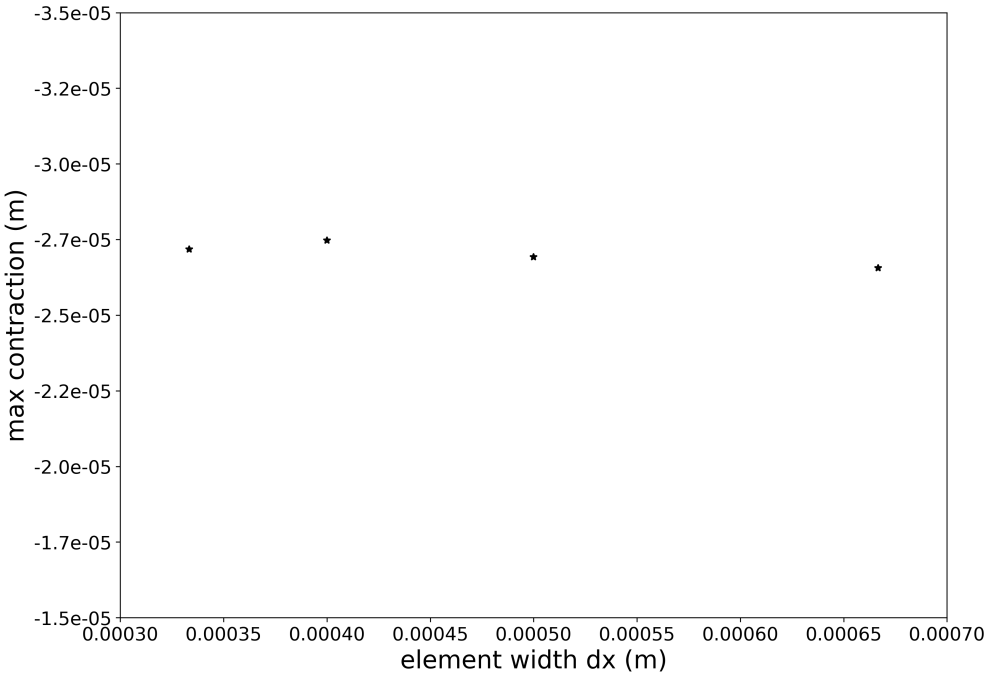


Figure 3.10: Maximum contraction *vs* element width in the x-direction in the reference configuration

3.4.2 Simulation Model Problems and Results

In this section, two simulation problems based on the setup in Section 3.4.1 are described, which also form the foundation for the optimization study in Chapter 4. The first one is a two-dimensional wall problem with width $20mm$ and height $10mm$. The referential domain is discretized with 40 elements in the x-direction and 38 in the y-direction for a layer thickness of $0.263mm$. The deposition time for one element is set to $0.006s$, equivalent to a printing speed of $83mm/s$. After the wall is finished building, another $240s$ of time is simulated to let the temperature of the entire wall cool down reasonably to ambient temperature. This matches a reali-life FDM printing process and provides a comparable condition for measuring shape error subsequently. Figure 3.11 shows the deformed and reference configuration of the wall, where the displacement is 5 times the actual value to better illustrate the deformation. Figure 3.12 displays the temperature change of the *top-right* node of the wall against time after the deposition is completed. In this case, the time this node is first added happens to be the time the deposition ends. Therefore, the total time plotted is $240s$ and the temperature decreases throughout the cooling process to the ambient temperature value. At the designated time of $240s$ after deposition ends, the temperature has sufficiently cooled down with maximum value around $376K$.

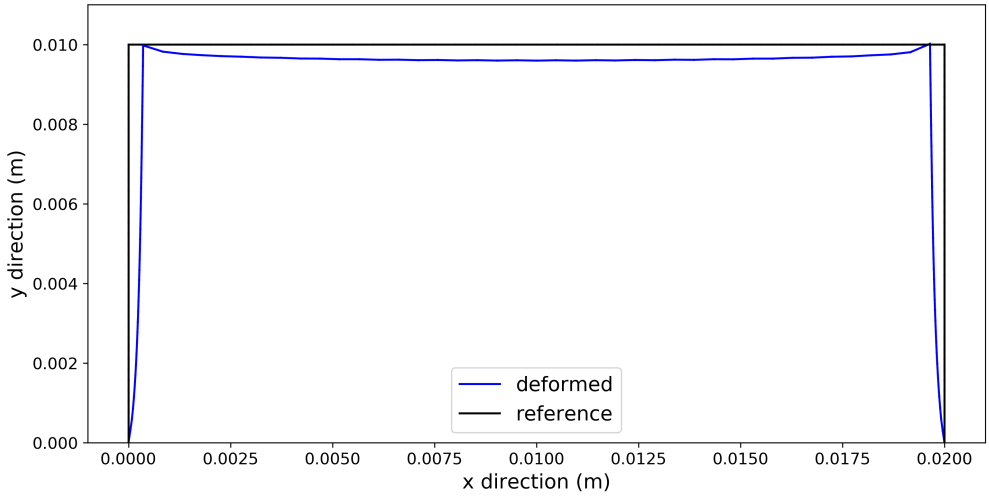


Figure 3.11: Deformed and reference shape of the two-dimensional wall with 5 times the actual displacement

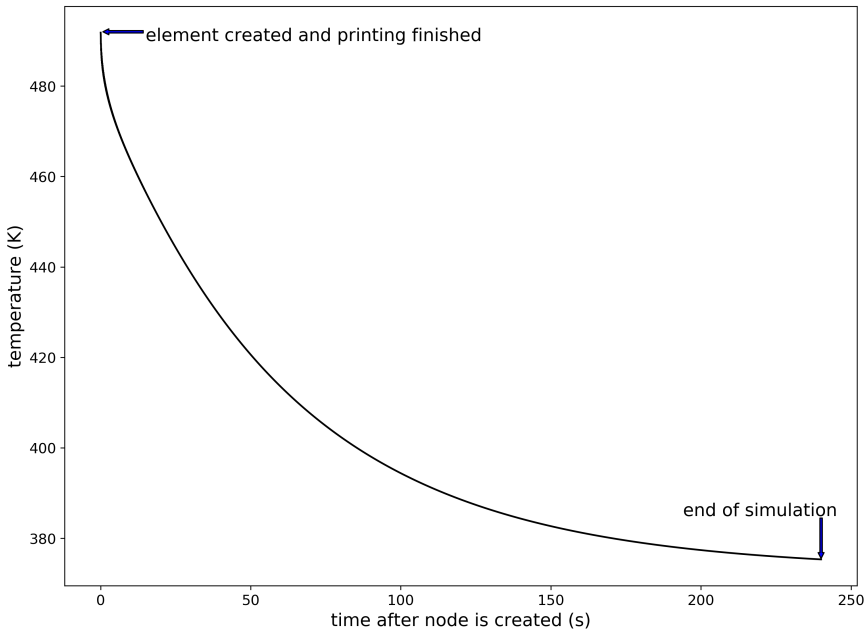


Figure 3.12: Temperature change with time of the *top-right* node of the wall

Figures 3.13 and 3.14 depict the displacement filled contours of the solutions at the last time step. The deformation is more pronounced with similar pattern shown in Section 3.4.1. As an example, the y-displacement change with respect to time of the *middle* node at the top edge is shown in Figure 3.15. The time axis of the curve starts when the deposition is completed and runs $240s$ until the simulation ends. The y-displacement initially drops quickly in the negative as the temperature cools down rapidly before stablizing as time proceeds. The

consistent increase in magnitude of the negative y-displacement further illustrates how the wall shrinks as it cools down. It is worth noting that the infinitesimal deformation assumption is valid in these results. The maximum absolute value in the y-direction displacement is $7.5 \times 10^{-5} m$ which is 0.75% compared to the original y-direction height at $10^{-2} m$.

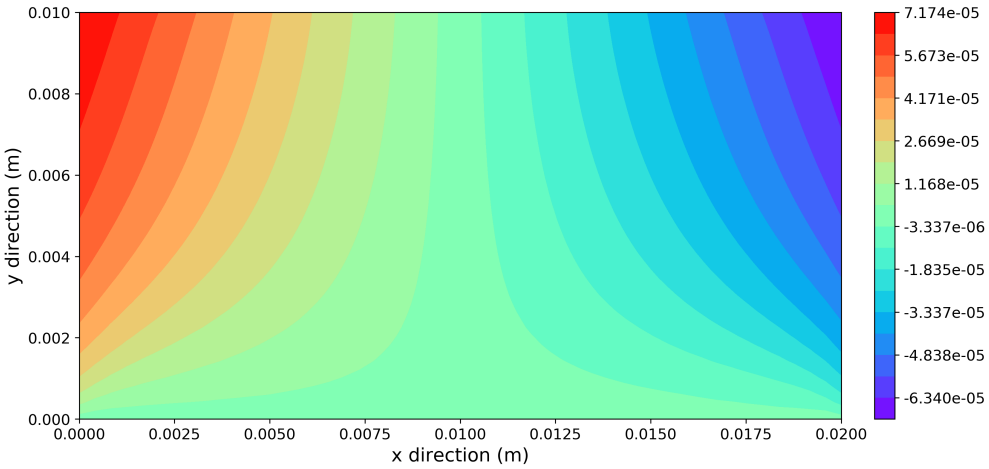


Figure 3.13: Displacement in the x-direction filled contour of the wall

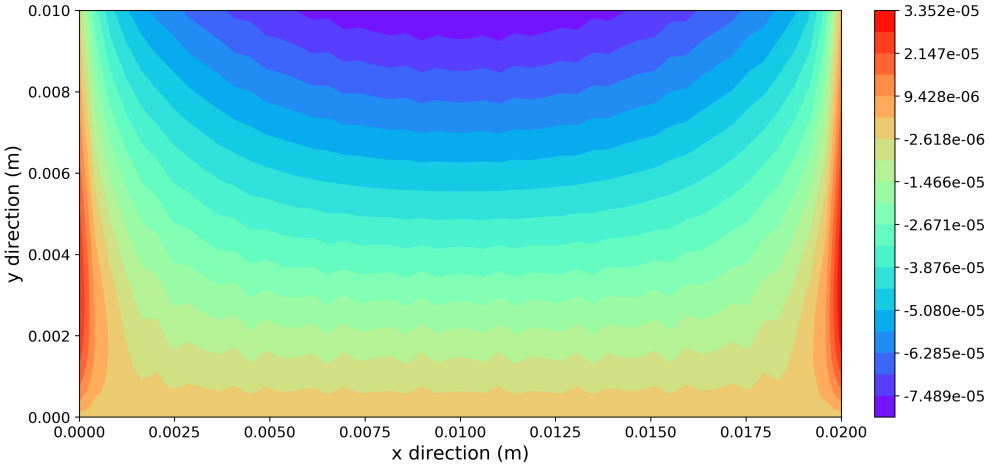


Figure 3.14: Displacement in the y-direction filled contour of the wall

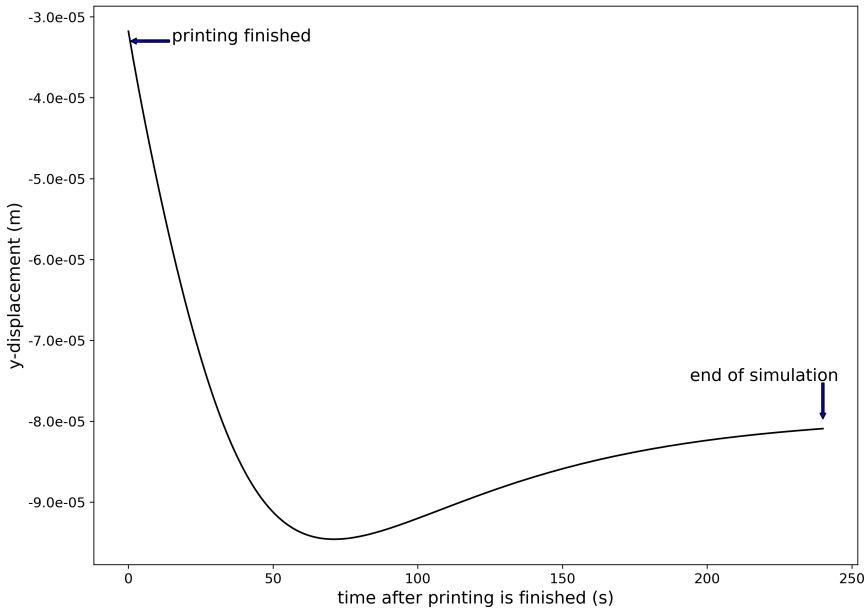


Figure 3.15: Displacement change in the y-direction with time of the *middle* node at the top edge of the wall

The second problem is again a two-dimensional wall (width $15mm$, height $10mm$), but now with a quarter-circular hole of radius $3mm$ at the top right end. The referential domain is discretized with full elements of the width $0.375mm$ and height $0.25mm$, while the partial elements have varying width. The simulation is run with partial elements and the one-sided slicing algorithms described in Chapter 2 until the temperature field becomes uniform and close to ambient temperature with $240s$ after deposition itself is finished. Figure 3.16 shows the temperature history of the *top-right* node, *i.e.*, the last-added node. The time starts when the node is first created and ends when the cooling stops $240s$ after the deposition is finished. Unlike the solid wall case, there is a small gap between when the node is created and the deposition is completed as marked on the figure. Nevertheless, the temperature gradually cools down to the ambient temperature value. The displacement information at the last time step is shown in Figures 3.17 to 3.19.

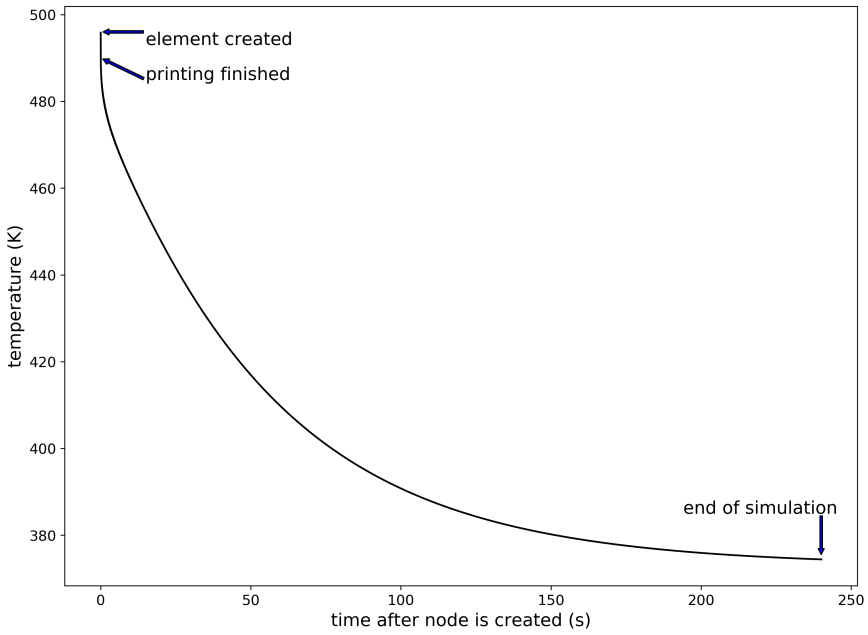


Figure 3.16: Temperature change with time of the last-added node of the wall with hole

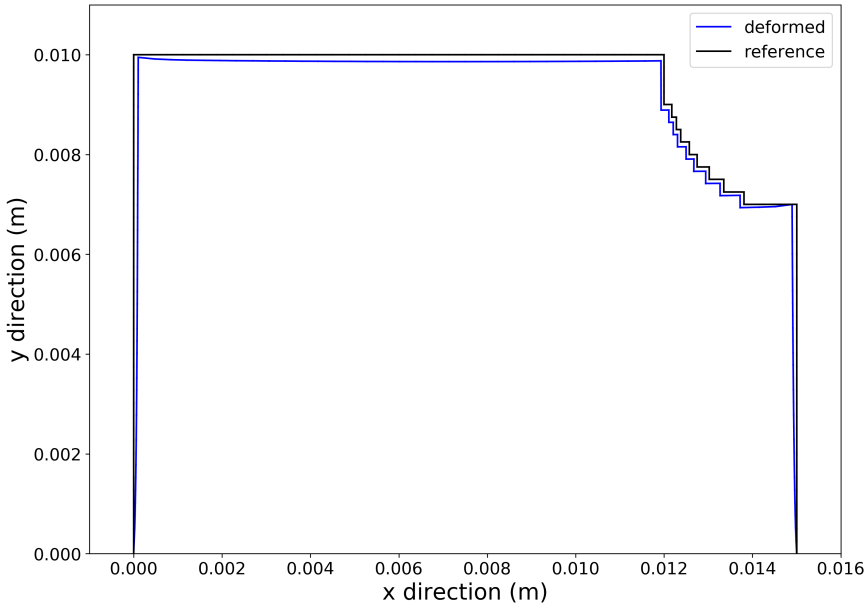


Figure 3.17: Deformed and reference shape of the two-dimensional wall with hole

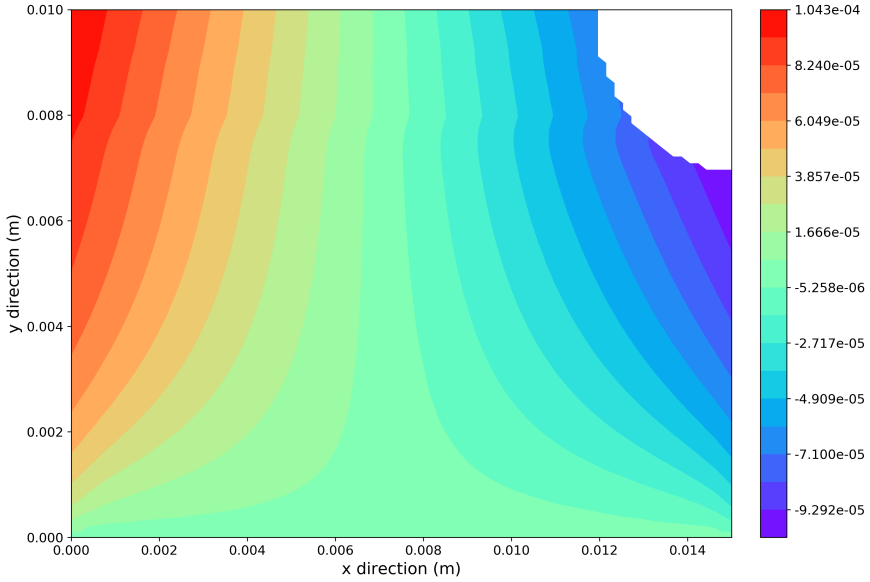


Figure 3.18: Displacement in the x-direction filled contour of the wall with hole

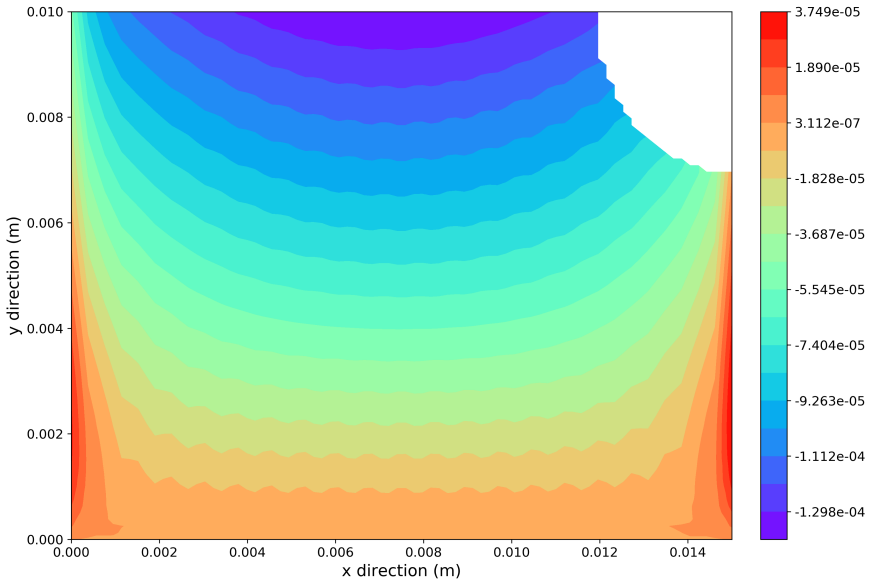


Figure 3.19: Displacement in the y-direction filled contour of the wall with hole

The filled contour plots are made in Python with matplotlib. The interpolation method for plotting is set to 'linear'. Although the geometry is different, the 'wall with hole' problem reflects similar displacement pattern with positive x-displacement on the left part and negative on the right. Negative y-direction displacement is again prominent throughout the object. As an example, the y-displacement of the last-added node is plotted against time

once the node is created, as shown in Figure 3.20. The initial negative displacement resulting from previous layers and elements continues to decrease, which implies the object is shrinking overall.

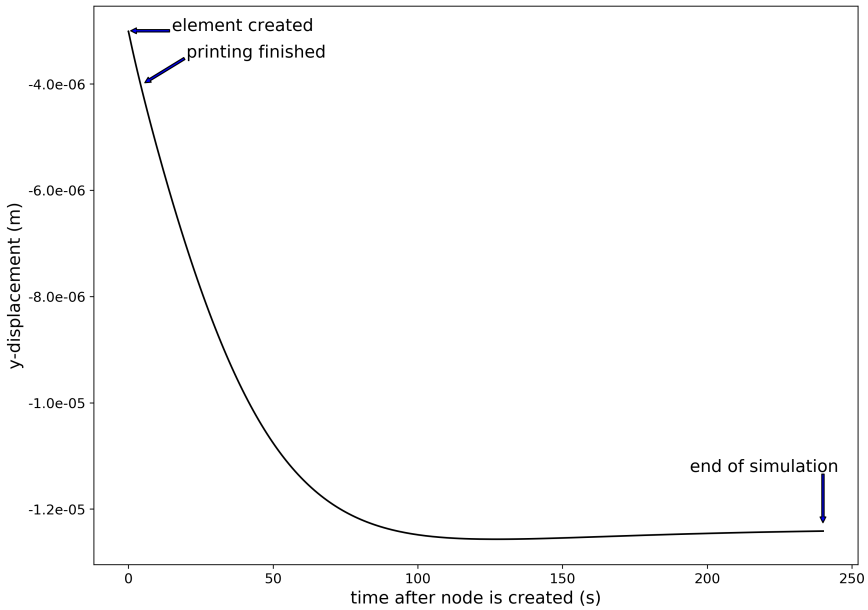


Figure 3.20: Displacement change in the y-direction with time of the last-added node of the wall with hole

The temperature manages to cool down in the end, $240s$ after deposition finishes, to a maximum value of $374K$.

A small oscillation in the y-displacement is also observed in both problems. This is because when a new node is added, its initial position often register opposite displacement to those of the existing nodes of the same element to compensate their displacement so that continuity and mass conservation are maintained. And such displacement passes on to the next element and layer. This stems from both the physical phenomenon of heat-coupled deformation during the printing process and the element addition algorithms of our choice. To show that it is unrelated to time-step size, the first problem discussed in this section, the two-dimensional wall, is simulated with half the time-step size, the same mesh and deposition speed. This translates to two time-steps for one new element while the total time remains the same. Figure 3.21 is the filled y-displacement contour of the same wall problem with this time-step treatment, and displays similar deformation pattern as in Figure 3.14. The problem would not be exactly the same numerically as the new element and nodes are created during the first of the two time-steps and then solved twice. Neverthless, the solution from this different approach of element addition shows that time discretization does not affect the small oscillation in the y-displacement observed.

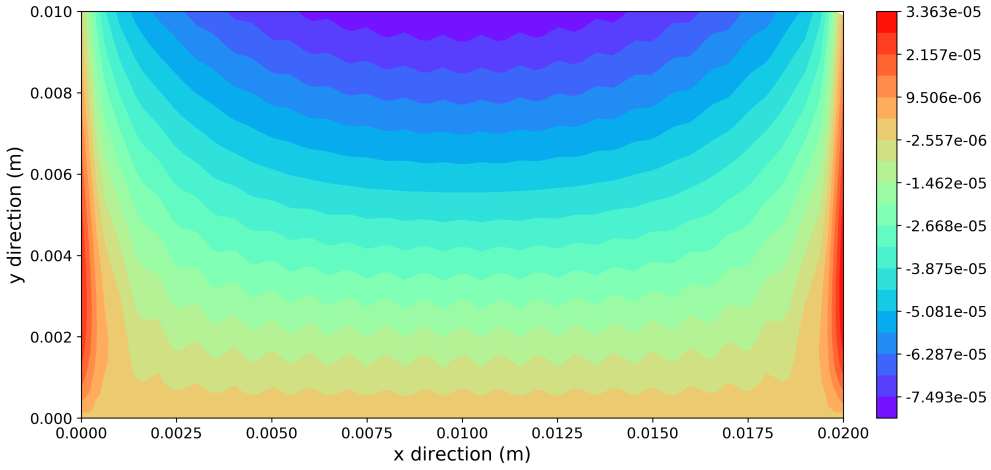


Figure 3.21: Displacement in the y-direction filled contour of the wall with smaller time-step size

3.5 Uncoupled Thermomechanical Model

In this section, an uncoupled thermomechanical model widely used in research and applications is briefly introduced and compared to the model developed in this paper by simulating the same problem. As described in [80], the uncoupled model solves the heat transfer problem on a predetermined Lagrangian mesh with elements being activated when the corresponding material is projected. The heat transfer equation corresponding to a rigid conductor is given as

$$c\dot{\theta} = -\nabla\cdot\boldsymbol{q}_0 + \rho_0 r, \tag{3.5}$$

where c is the volumetric heat capacity, and the external heat supply is represented by $\rho_0 r$. In order to highlight the coupling effect on the solutions, the constitutive relations for both models are kept the same whenever suitable. In addition, the same mesh generation and element addition algorithms introduced in earlier sections are applied. However, since the displacement is not solved at each time step, element addition in the deformed configuration no longer encounters deformation from previous layers and existing elements. Displacement history variables are no longer in use either.

The temperature history computed by this model on the fixed reference configuration is then used to determine the mechanical response. The stress equilibrium is governed by Equation (3.6)

$$\nabla \cdot \boldsymbol{\sigma} = 0, \tag{3.6}$$

where $\boldsymbol{\sigma}$ is the stress. Usually a simple constitutive law for the material is assumed for elastic strain and stress as shown in Equation (3.7),

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\epsilon}_e, \tag{3.7}$$

where $\boldsymbol{\epsilon}_e$ is the elastic strain and $\mathbb{C}$ is the elasticity tensor. Here $\mathbb{C}$ is assumed to be temperature-dependent and can be described in the same way as in Equation (2.43). The

thermally induced strain $\boldsymbol{\epsilon}_T$ is given by

$$\boldsymbol{\epsilon}_T = \alpha \left(\theta - \theta_0\right) \boldsymbol{I}, \tag{3.8}$$

where α the thermal expansion coefficient and θ_0 the reference temperature where no thermal-induced strain occurs. The total strain $\boldsymbol{\epsilon}$ is subsequently

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_e + \boldsymbol{\epsilon}_T. \tag{3.9}$$

The same problem of two-dimensional wall building is simulated with both models. The wall has a width of $20mm$ and a height of $10mm$ with 40 elements in the x-direction and a layer thickness of $0.25mm$ in the reference configuration. The printing speed is set at $100mm/s$ with an interval time of $30ms$ between layers. The wall is allowed to cool down to ambient temperature for $240s$ after deposition is finished. First, the results of the coupled model are shown. Figure 3.22 shows the deformed and reference configuration where small but visible deformation is present. The displacement in the figure is 5 times the actual value to display it clearly. Figure 3.23 and Figure 3.24 are the filled contour plots of the displacement in x and y direction with results showing similar contraction of the wall as the ones presented in Section 3.4.

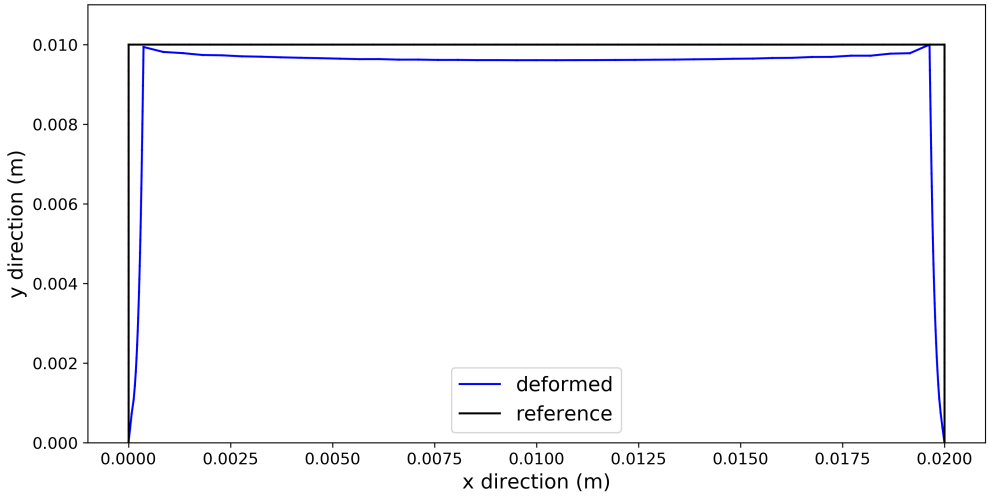


Figure 3.22: Deformed and reference shape of the two-dimensional wall with 5 times the actual displacement: coupled model

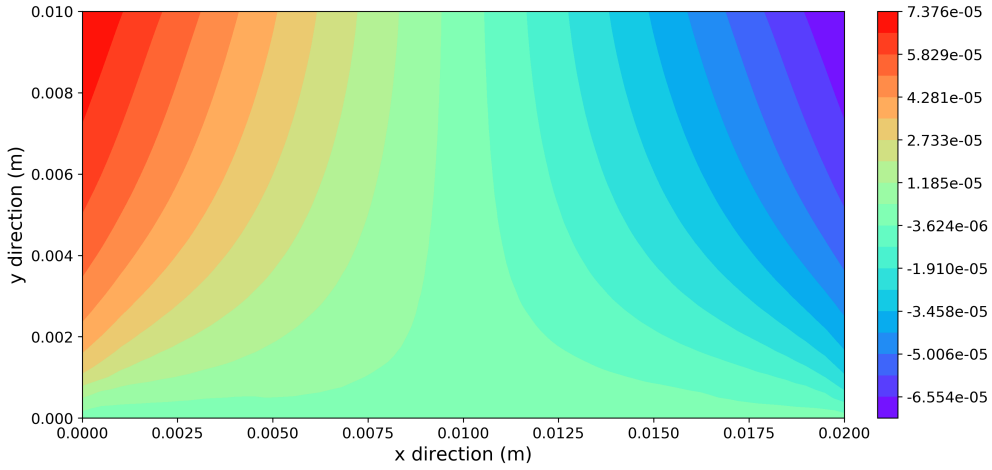


Figure 3.23: Displacement in the x-direction filled contour of the wall: coupled model

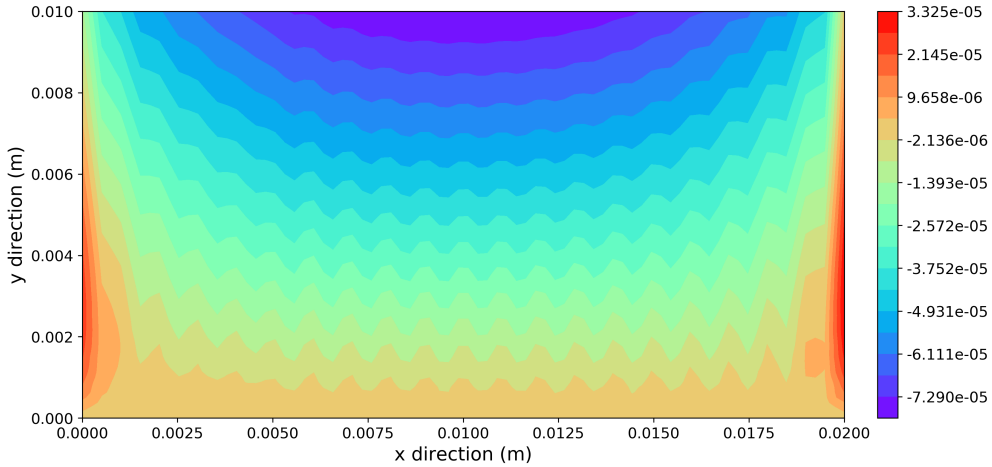


Figure 3.24: Displacement in the y-direction filled contour of the wall: coupled model

Next the same problem is solved with the uncoupled model described earlier in this section. Since enough time steps are taken for the temperature to come down, the final temperature field would naturally be almost identical from both models with maximum values around $375K$ and ambient temperature $373K$. The maximum temperature difference at all nodes $240s$ after printing finished is negligible at $0.06632K$. The deformation solutions, on the other hand, show the critical difference between the two methods. It is noted that the uncoupled method possesses no means of the thermal or mechanical history of the manufacturing process as its deformation is solved statically, which leads to the same strain and stress given the same current temperature distribution regardless of how the temperature or displacement evolved. The temperature field at the final step solved from the heat transfer equation is used to gain the displacement. Figure 3.25 illustrates the deformed boundary

of the wall together with the referential one, where the displacement is 5 times the actual value. Figure 3.26 and Figure 3.27 plot the displacement in x and y direction respectively.

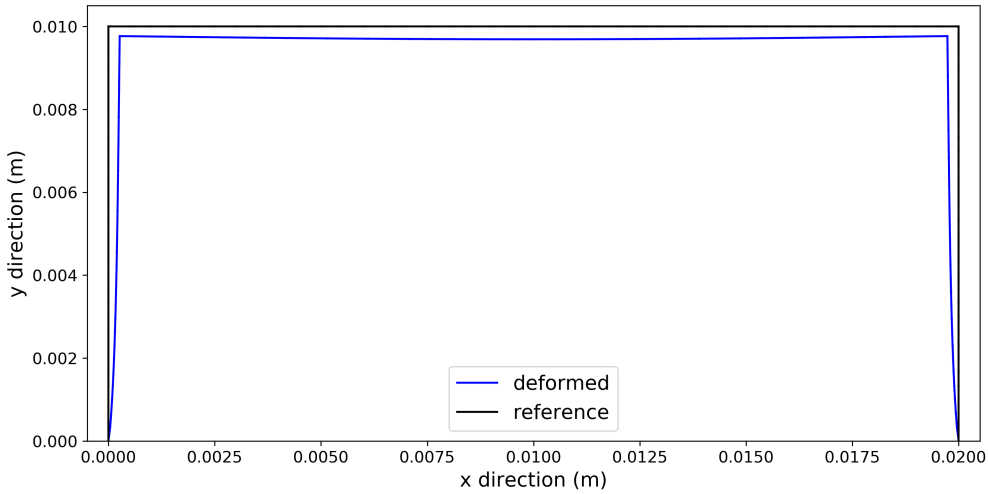


Figure 3.25: Deformed and reference shape of the two-dimensional wall with 5 times the actual displacement: uncoupled model

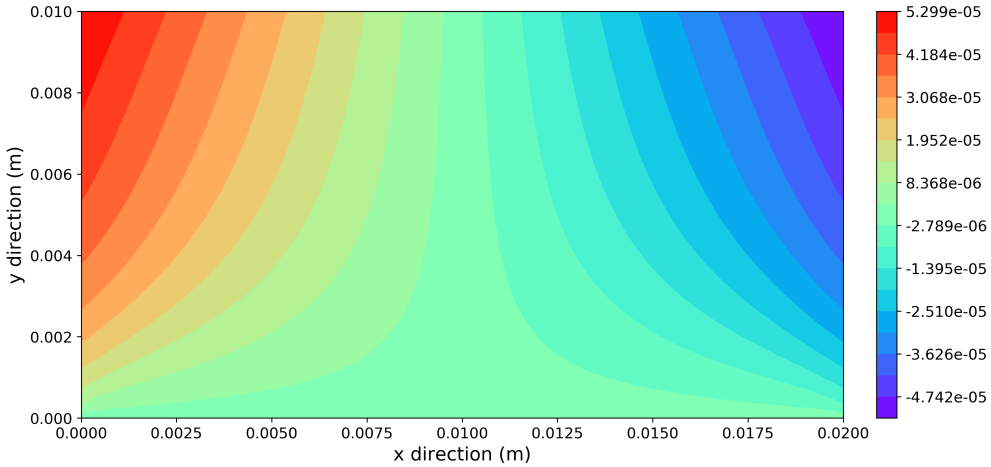


Figure 3.26: Displacement in the x-direction filled contour of the wall: uncoupled model

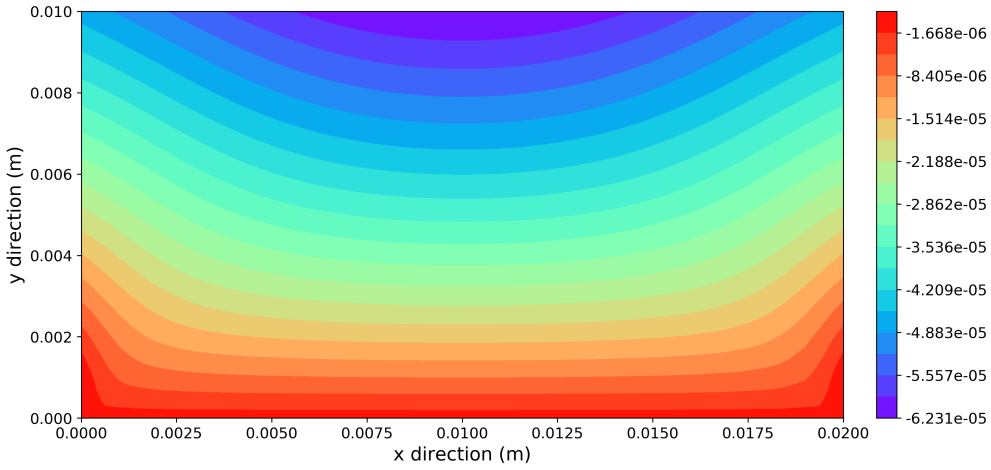


Figure 3.27: Displacement in the y-direction filled contour of the wall: uncoupled model

The zoomed-in views of the deformation from the two models are shown in Figure 3.28 and 3.29. The displacement is five times the actual values in the plots to highlight their differences. It is clear that the uncoupled model produces more even displacement throughout the domain, especially in the y direction across each layer as no temperature or deformation history is taken into account. The coupled model shows larger negative y-displacement at the middle of the wall in x-direction. Deformation history plays an important role in the coupled model as seen in the y-displacement filled contour in Figure 3.24. A pattern of minor oscillation across the x-direction is observed due to new element compensating for the deformation of the existing layers and elements. Such phenomenon is non-existent for the uncoupled model. In all, both models predict shrinkage and small deformation while the deformation patterns differ somewhat. It is worthwhile to explore the coupled model when higher accuracy of deformation is desired. It also allows the small deformation assumption to hold with larger accumulated displacement because of the use of history displacement variables.

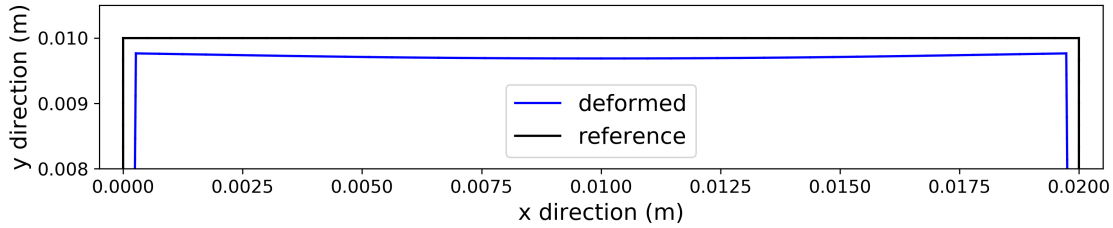


Figure 3.28: Detailed deformed and reference shape of the two-dimensional wall: uncoupled model

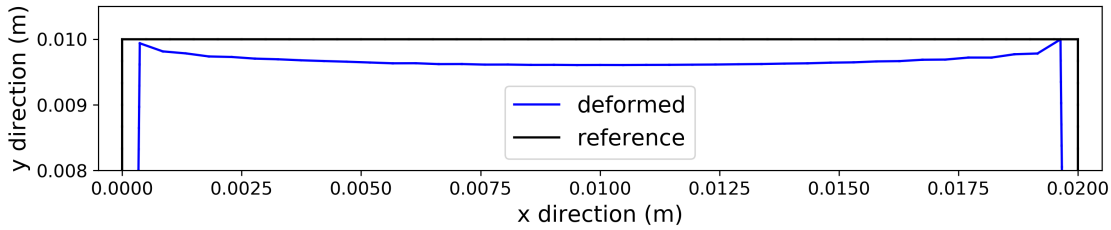


Figure 3.29: Detailed Deformed and reference shape of the two-dimensional wall: coupled model

Chapter 4

Optimization

4.1 Introduction

Fused deposition modeling, whether used for polymer, ceramics or metal sometimes creates undesirable internal and surface defects. Potential errors include overflow at the start and end of a segment due to acceleration and deceleration of material extrusion, curve approximation error from staircase effect *etc.* Many process variables can be optimized to better achieve the design goal [83, 84, 85]. They can be material properties like viscosity and thermal conductivity or machine specific parameters like filament diameter and nozzle size. Based upon the finite element simulation model developed in Chapter 2, focus of our optimization is on the operation parameters such as layer thickness and head speed [86].

In contrast to existing optimization research work as outlined in Chapter 1, a different approach is taken here to tackle the optimization of FDM process as a PDE-constrained optimization problem. In the previous chapters, a general framework for finite element simulation of two-dimensional FDM manufacturing process is developed. The fully discretized finite element model is parameterized to incorporate potential optimization parameters. The system of PDEs as derived in Chapter 2 at time step t_{n+1} can be written as

$$\boldsymbol{r}_{n+1}\left(\boldsymbol{u}_{n+1}, \boldsymbol{u}_n, \boldsymbol{v}_n, \boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n, \boldsymbol{u}^{his}_{n+1}, \boldsymbol{\theta}^{his}_{n+1}, \boldsymbol{\mu}, t_{n+1}\right) = \boldsymbol{0}, \tag{4.1}$$

where $\boldsymbol{r}_{n+1}$ is the set of discretized PDEs that describe the system at t_{n+1}, $\boldsymbol{u}_{n+1}$ the displacement at the current time step, $\boldsymbol{u}_n, \boldsymbol{v}_n$ the displacement and velocity at the previous time step, $\boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n$ the temperature at current and previous time step, $\boldsymbol{u}^{his}_{n+1}$ the displacement history variable, $\boldsymbol{\theta}^{his}_{n+1}$ the temperature history variable, and $\boldsymbol{\mu}$ the optimization parameters. The history variables are introduced in Section 2.3.2.

As mentioned before, various objectives have been studied including surface roughness and residual stress. In this dissertation, attention is focused on a measure of shape error that computes the dimensional deviation caused by deformation during the printing process from the desired shape. Shape error or closely related surface roughness is a major problem of AM processes and efforts have been taken to address it through process map or post-processing

treatment as reviewed above [87]. Here the shape error is defined on the particular edges of interest in two dimensions. The two simulation problems solved in Section 3.4.2 are used to illustrate the shape error definition.

The general definition of our objective function f_0 is

$$f_0(\boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\mu}, t) = \frac{\left[\int_{\Gamma_S^t} \left(d(\boldsymbol{u}) - \bar{d}\right)^2 dl\right]^{\frac{1}{2}}}{L_c \left(\int_{\Gamma_S^t} dl\right)^{\frac{1}{2}}}, \tag{4.2}$$

where Γ_S^t refers to the edges of interest, $d(\boldsymbol{u}) - \bar{d}$ is a difference measure in position at a point and L_c is a critical length dependent on the problem to make f_0 dimensionless. Notice that f_0 depends only on $\boldsymbol{u}$ explicitly. The objective function is evaluated through one-dimensional Gaussian quadrature with consistent spatial discretization with the mesh itself. The discretized form is

$$f_{0,h} = \frac{\left[\sum_{i=1}^{n} \sum_{j=1}^{m} \left(d(\boldsymbol{u}_j) - \bar{d}\right)^2 w_j \frac{\Delta l_i}{2}\right]^{\frac{1}{2}}}{L_c \left(\sum_{i=1}^{n} \Delta l_i\right)^{\frac{1}{2}}}, \tag{4.3}$$

where n is the number of discretized edges and m is the order of quadrature used. Standard Gaussian quadrature with nodes between $[-1, 1]$ is used with $\frac{\Delta l_i}{2}$ being the Jacobian, where linear elements for the edges are assumed in accordance with the bilinear elements.

For the first example problem, the two-dimensional wall problem, Γ_S^t is the top edge, and $d(\boldsymbol{u}) - \bar{d}$ is defined as the difference between the y-coordinate of a quadrature point in the deformed configuration and its designed height with same x-coordinate in the reference configuration as shown in Figure 4.1. The displacement is exaggerated to five times the original values to more clearly display the shape error definition. The reference (black) edges have zero shape error while the deformed ones (blue) deviate from their ideal height due to the thermomechanical deformation. The critical length L_c is chosen to be the height of the wall.

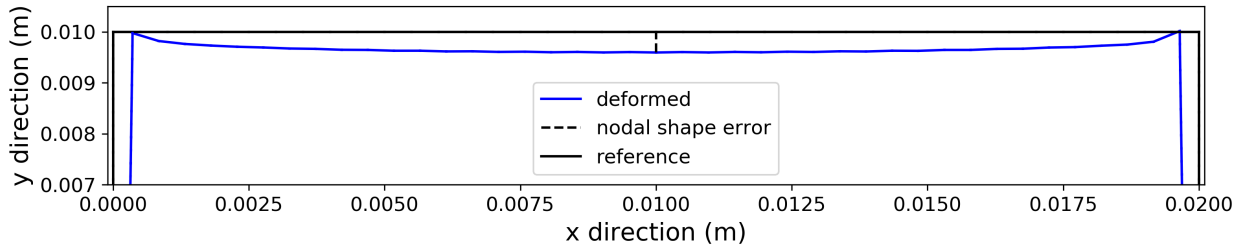


Figure 4.1: Measurement of shape error for two-dimensional wall with 5 times the actual displacement

For the second problem, the wall with a circular hole, $d(\boldsymbol{u}) - \bar{d}$ is defined as $|\boldsymbol{x} - \boldsymbol{x}_0| - r$, where $\boldsymbol{x}_0$ is the center of the circle and r is its radius, and Γ_S^t is the step-edge approximation,

or the 'error edges', of the quarter-circle hole. The 'error edges' are defined as the edges formed by connecting the middle points of the actual printed step edges approximation of the circular hole, as shown in Figure 4.2. This guarantees a total edge length that would converge to $\pi r/2$ instead of $2r$ as the mesh is refined. A zoomed-in view is given in Figure 4.3 with 50 elements in the x-direction for a full layer and 40 elements in the y direction. The critical length L_c here is chosen to be the radius of the circle.

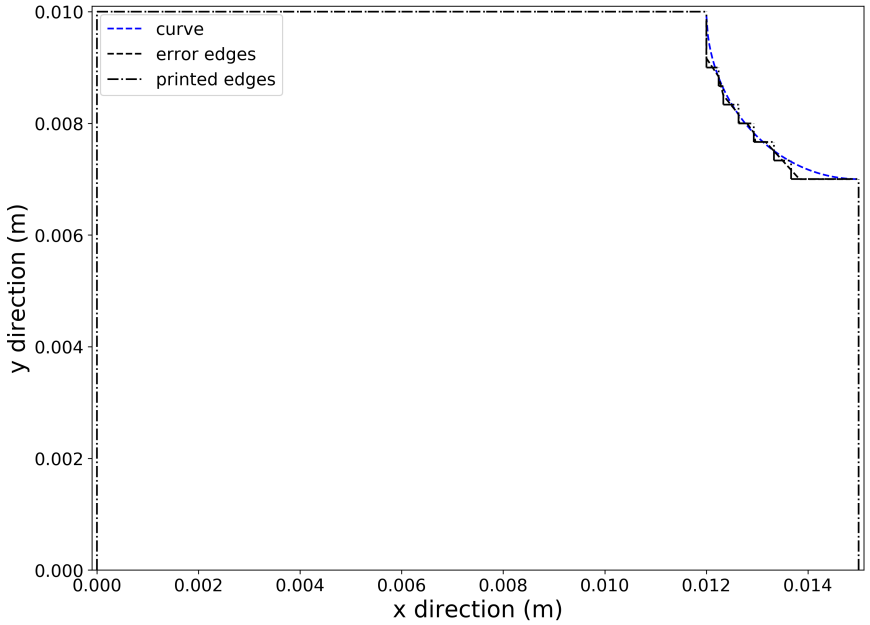


Figure 4.2: Error edges for two-dimensional wall with hole

The discussion on optimization methods is divided into two parts, one focusing on a *discrete parameter*, *i.e.*, the layer thickness, and the other on *continuous parameters*, *e.g.*, convection coefficient, as they show different properties and thus offer different insights into the best approaches to optimization.

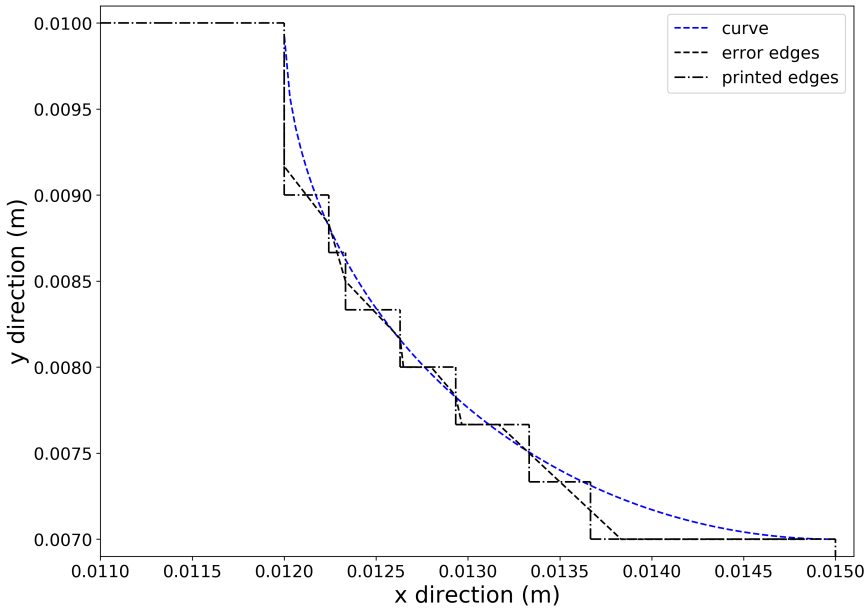


Figure 4.3: Error edges for two-dimensional wall with hole zoomed in

4.2 Layer Thickness Optimization

An obvious optimization parameter is the layer thickness, which presents a challenge to the optimization problem since given the height of the printed object, it corresponds to a discrete variable, which is the number of layers, an integer. This is particularly an issue for approximating curves and measuring the associated shape errors because a major source of the shape error defined in Section 4.1 comes from pure geometric error in the deposition. Therefore, to optimize the number of layers therefore, a good understanding of its impact on the objective function needs to be established.

In order to argue that the geometric error comes from the choice of layer thickness, the artificial shape error caused by discretization in the x-direction needs to be evaluated and reduced as much as possible. This is a non-issue for the two-dimensional wall problem as no slicing algorithm for curvature is needed and the pure geometric error is zero. For the wall with a quarter-circle hole, however, different slicing algorithms result in drastically different results. To appreciate this, consider the geometry shown in Figure 4.2 and fix the number of layers at 34. Without solving the thermomechanical equations, the pure geometric objective function, the shape error, defined in Section 4.1 can be evaluated and plotted against the number of elements in the x-direction for a full layer. The two different slicing algorithms discussed in Chapter 2, one-sided and minimum-area are used to construct the referential mesh. The results are plotted in Figure 4.4 for the one-sided method and Figure 4.5 for the minimum-area method.

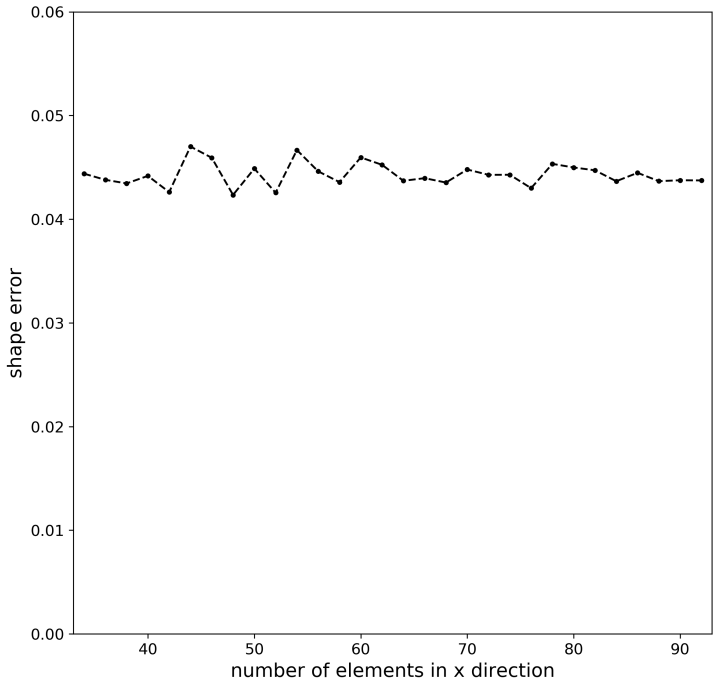


Figure 4.4: Shape error *vs* number of elements in x-direction with one-sided slicing and 34 layers

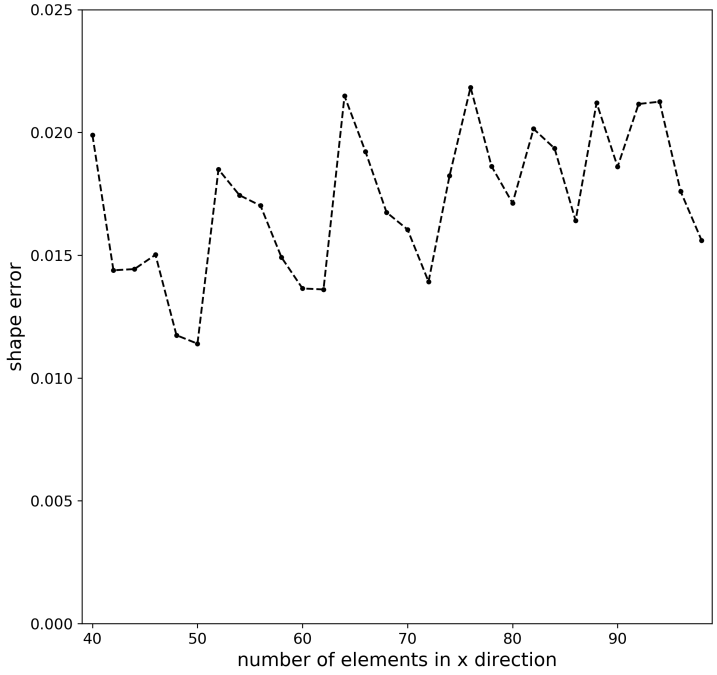


Figure 4.5: Shape error *vs* number of elements in x-direction with maximum-area slicing

It is clear that one-sided slicing provides a near constant shape error as it only relies on y-coordinates of the boundary elements and nodes, given by the number of layers, to determine the width of the partial element and thus the shape error, making it mostly independent on the discretization in the x-direction. However, due to the 40% limit on the area of partial elements, some fluctuations are still visible but nevertheless too small to meaningfully change the potential optimization steps. The minimum-area slicing algorithm, on the other hand, relies on the x-coordinates of the vertices of the boundary elements and therefore is not ideal in isolating the effect on the objective function. The error as seen in Figure 4.5 shows shape error oscillations with respect to the x-size of the mesh. 'Error edges' with selected number of elements in the x-direction of 40 and 50 are plotted in Figure 4.6. Clear differences in the positions of 'error edges' lead to diverging shape errors. An alternative minimum-area slicing algorithm independent of mesh in the x-direction might resolve this issue, but here throughout the optimization of the simulation problems the one-sided slicing algorithm is preferred and chosen to get a better understanding of how layer thickness affects the shape error.

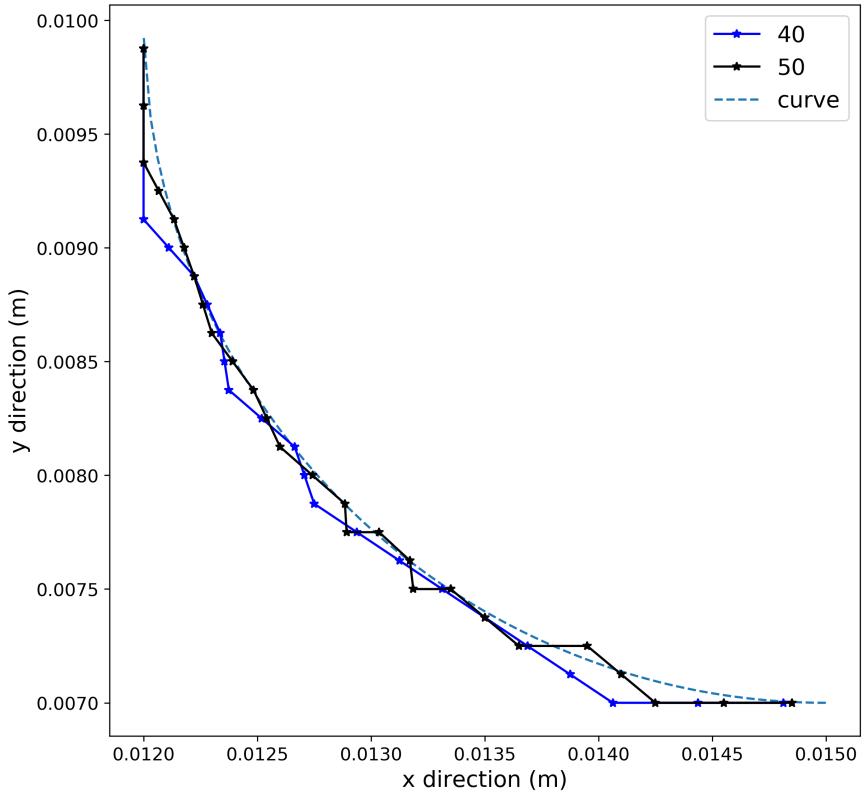


Figure 4.6: Shape error of 40 and 50 elements in the x-direction with 40 layers with minimum-area slicing

The effect of the number of layers on the geometric shape error is examined next. Since the number of elements in the x-direction does not fundamentally alter the pure geometric

shape error, a good aspect ratio for the quadrilateral elements can be maintained while shape error is calculated with different number of layers. To this end, let Δx and Δy be the width and height of a full element in the reference configuration, respectively. Fewer elements in the x direction is more advantageous in terms of computing time but reduces the aspect ratio of $\Delta x/\Delta y$ which can lead to errors if too small. Therefore, a compromised value should be sought. Figure 4.7 shows how the number of elements in both x and y directions change to maintain an aspect ratio of 1.0 and is linear by definition. For this case, Figures 4.8 and 4.9 display the shape error decreasing with smaller elements, where the element size measure Δh is calculated as $\Delta h = (\Delta x^2 + \Delta y^2)^{\frac{1}{2}}$. Particularly, it converges to 0 with an order of $O(\Delta h)$.

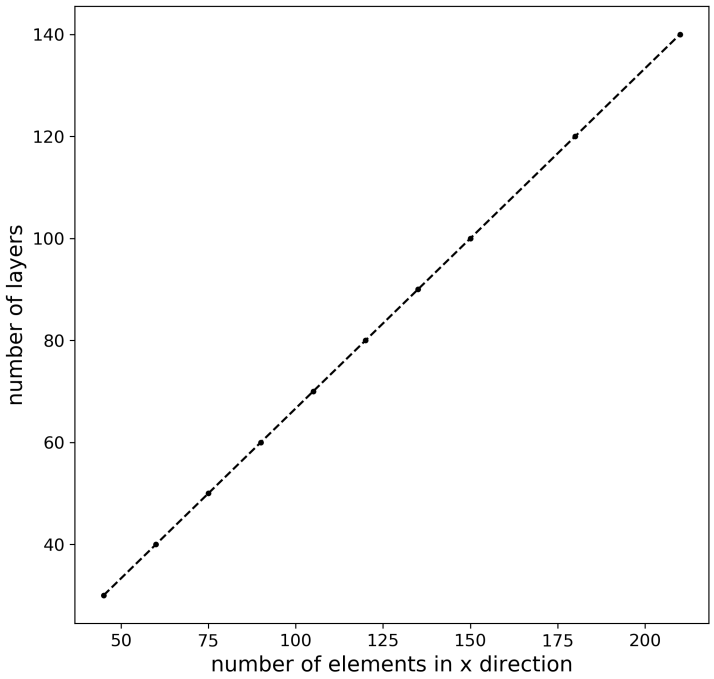


Figure 4.7: Number of elements in x direction *vs* number of layers with one-sided slicing

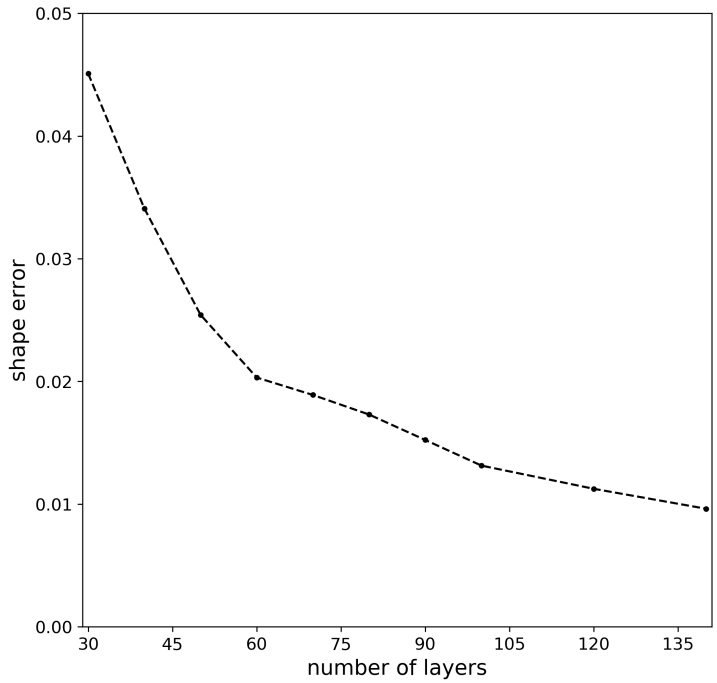


Figure 4.8: Shape error *vs* number of layers with one-sided slicing for aspect ratio of 1.0

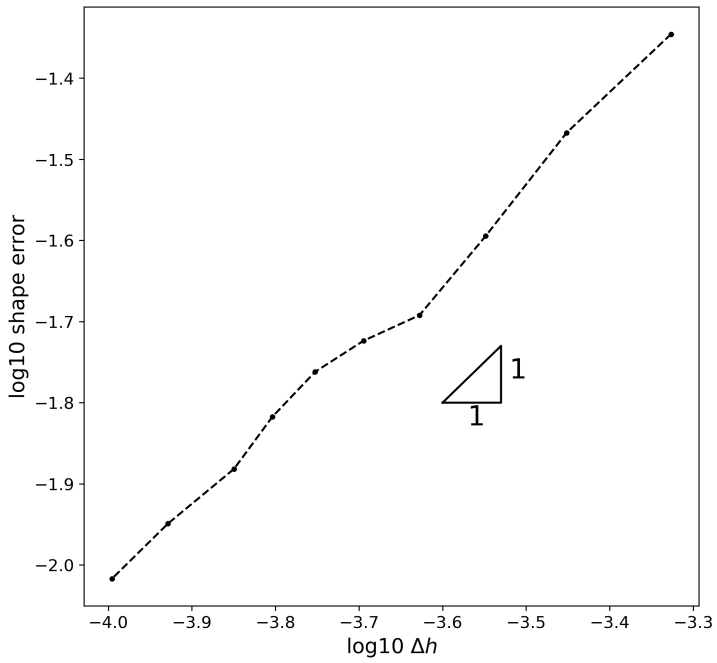


Figure 4.9: Log shape error *vs* log number of layers with one-sided slicing for aspect ratio of 1.0

Taking a closer look at Figure 4.8 between 30 and 60 layers, which are within the range

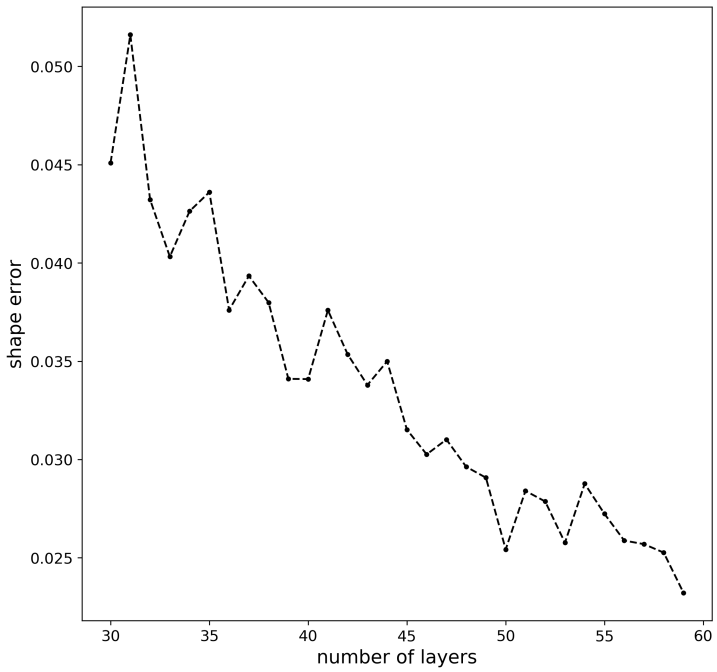


Figure 4.10: Shape error *vs* number of layers with one-sided slicing with smaller range

of the number of layers in real applications given the height of the wall, the finely resolved shape error is plotted in Figure 4.10. Here another key effect of this discrete parameter is exhibited, which is the large number of local minimum and maximum values that are crucial to the optimization outcome. To better understand this phenomenon, the cases of layer number 40 and 41 layers are examined in Figure 4.11 where both the 'error edges' and the target curve are plotted. It is clear that the majority of the shape error occurs around the horizontal part of the approximation of the quarter-circle. The discreteness of the number of layers means that the horizontal approximation varies a lot depending on whether the height of the horizontal curve is an integer multiple of the layer thickness. In turn such errors cause the local shape error to fluctuate. The errors of these two cases are shown in Figure 4.11 and amount to an against-the-trend jump of 10% from 40 to 41 layers.

In addition, the fact that $|\boldsymbol{x} - \boldsymbol{x}_0| - r$ is the integrand along the 'error edges' magnifies such local changes. This is because $|\boldsymbol{x} - \boldsymbol{x}_0| - r$ itself is very small in value by definition and prone to small changes with $\boldsymbol{x}$. These factors contribute to the difficulty of optimization of layer thickness using local methods that are likely to become stuck on local minima. At the same time without effective gradient a more naive and exhaustive optimization method has to be adopted. However, it is possible to design efficient and effective optimization of the number of layers without having to build out the contour of the entire parameter domain. The shape errors maintains a clear trend with respect to the number of layers, allowing the optimization algorithm to potentially "climb out" of the local minima by taking a larger step size. Algorithms that reflect the knowledge gained in this section are proposed in

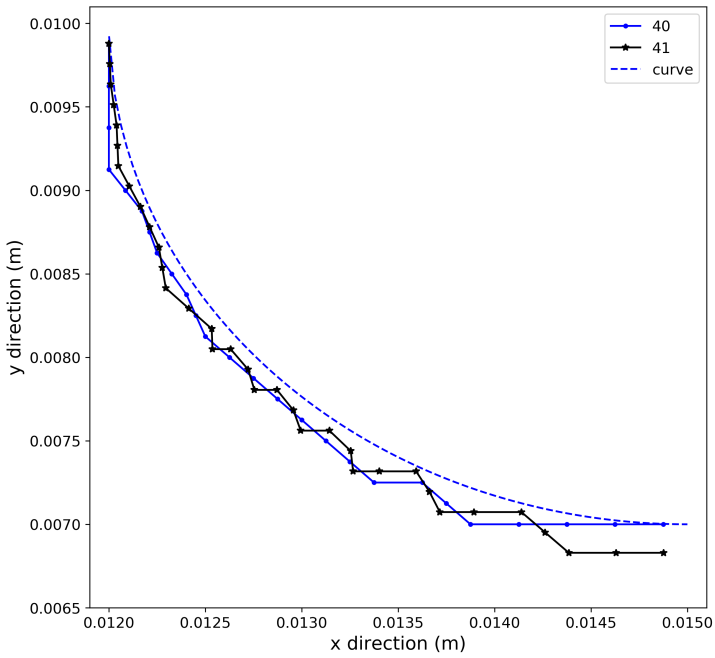


Figure 4.11: Shape error of 40 and 41 layers with one-sided slicing

Sections 4.4.1 and 4.4.2.

4.3 Continuous Parameters

The sensitivities of the continuous optimization parameters $\boldsymbol{\mu}$ can be obtained through careful differentiation of the entire fully discretized PDEs with respect to them. This allows for gradient-based optimization methods to be designed and employed on them. The formal optimization problem is written as

$$\begin{aligned} \underset{\boldsymbol{\mu}}{\text{minimize}} \quad & f_0\left(\boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\mu}, t\right) \\ \text{subject to} \quad & \boldsymbol{r}\left(\boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\mu}, t\right) = \mathbf{0} \\ & f_i(\boldsymbol{\mu}) \leqslant b_i, \ i = 1, \ldots, m, \end{aligned} \tag{4.4}$$

where f_0 is the objective function, r the discretized PDEs, f_i the constraints, $\boldsymbol{u}$ the displacement, $\boldsymbol{\theta}$ the temperature and m the number of constraints. First-order sensitivities of the quantities of interest

$$\frac{df_i}{d\boldsymbol{\mu}}, \qquad i = 0, 1, ...m, \tag{4.5}$$

can be derived for given objective functions and constraints. In our particular case the objective is shape error defined in Equation (4.2).

Following Section 2.2, the sensitivities are calculated based on the fully discretized forms of the governing equations (4.1) both in space and time. At time t_{n+1}, the displacement and temperature variables are written as $\boldsymbol{u}_{n+1}$ and $\boldsymbol{\theta}_{n+1}$. The sensitivity of the shape error objective with respect to parameters can be written as

$$\frac{df_0}{d\boldsymbol{\mu}}\left(\boldsymbol{u}_{n+1}\right) = \frac{\partial f_0}{\partial \boldsymbol{u}_{n+1}} \frac{\partial \boldsymbol{u}_{n+1}}{\partial \boldsymbol{\mu}}, \tag{4.6}$$

since f_0 is dependent only on displacement explicitly. The first term on the right hand side in Equation (4.6) $\frac{\partial f_0}{\partial \boldsymbol{u}_{n+1}}$ can be computed by differentiating Equation (4.3). The state sensitivities $\frac{\partial \boldsymbol{u}_{n+1}}{\partial \boldsymbol{\mu}}$ at time step t_{n+1} however rely implicitly on sensitivities from all the previous time steps that need to be calculated and tracked through the simulation.

From Equation (4.1), differentiate both sides of the equation with respect to $\boldsymbol{\mu}$ to gain

$$\begin{aligned} \frac{d\boldsymbol{r}_{n+1}}{d\boldsymbol{\mu}}\left(\boldsymbol{u}_{n+1}, \boldsymbol{u}_n, \boldsymbol{v}_n, \boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n, \boldsymbol{u}_{n+1}^{his}, \boldsymbol{\theta}_{n+1}^{his}, \boldsymbol{\mu}, t_{n+1}\right) = \boldsymbol{0}, \\ \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_{n+1}}\frac{\partial \boldsymbol{u}_{n+1}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_n}\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{v}_n}\frac{\partial \boldsymbol{v}_n}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_{n+1}}\frac{\partial \boldsymbol{\theta}_{n+1}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{\theta}_n}{\partial \boldsymbol{\mu}} \\ + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_{n+1}^{his}}\frac{\partial \boldsymbol{u}_{n+1}^{his}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_{n+1}^{his}}\frac{\partial \boldsymbol{\theta}_{n+1}^{his}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\mu}} = \boldsymbol{0}. \end{aligned} \tag{4.7}$$

State sensitivities $\frac{\partial \boldsymbol{u}_{n+1}}{\partial \boldsymbol{\mu}}$, $\frac{\partial \boldsymbol{\theta}_{n+1}}{\partial \boldsymbol{\mu}}$ at time t_{n+1} can be calculated implicitly based on Equation (4.7) with state sensitivities from the previous two time steps. Provided with the correct initial conditions for sensitivities, the ones at the last time step needed for optimization can be calculated. From Equation (4.7), all the remaining terms need to be examined and computed. It is noted that $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_{n+1}}$, $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_{n+1}}$ in discretized matrix forms are already derived when solving the PDEs with the Newton-Raphson iteration and given in Equation (2.79) at the element level. The fully discrete sensitivities of $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_n}$, $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_n}$, $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{v}_n}$ can be obtained by substituting the Newmark integration scheme into the fully discretized PDE system as in Equation (2.69) and taking the appropriate differentiation. To deduce the detailed forms, momentum and energy balances are denoted separately as $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. Here, $\boldsymbol{r}_{1,n+1}$ represents the discrete form of linear momentum balance and $\boldsymbol{r}_{2,n+1}$ the discrete energy balance at time step t_{n+1}. Consequently, equation (4.1) can be rewritten as

$$\begin{aligned} \boldsymbol{r}_{1,n+1}\left(\boldsymbol{u}_{n+1}, \boldsymbol{u}_n, \boldsymbol{v}_n, \boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n, \boldsymbol{u}_{n+1}^{his}, \boldsymbol{\theta}_{n+1}^{his}, \boldsymbol{\mu}, t_{n+1}\right) = \boldsymbol{0}, \\ \boldsymbol{r}_{2,n+1}\left(\boldsymbol{u}_{n+1}, \boldsymbol{u}_n, \boldsymbol{v}_n, \boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n, \boldsymbol{u}_{n+1}^{his}, \boldsymbol{\theta}_{n+1}^{his}, \boldsymbol{\mu}, t_{n+1}\right) = \boldsymbol{0}. \end{aligned} \tag{4.8}$$

The fully discretized matrix-vector forms of $\boldsymbol{r}_{1,n+1}$ and $\boldsymbol{r}_{2,n+1}$ are given in Equations (2.63)

and (2.64). The element-level sensitivity matrices for $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_n}$, $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_n}$, $\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{v}_n}$ are

$$\begin{aligned}
\left[\frac{\partial \boldsymbol{r}^e_{1,n+1}}{\partial \boldsymbol{u}_n}\right] &= -\frac{4}{\Delta t_n^2}\left[\boldsymbol{M}^e_u\right], & \left[\frac{\partial \boldsymbol{r}^e_{2,n+1}}{\partial \boldsymbol{u}_n}\right] &= \int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \left[\theta^e_{n+1}\right] \left[\boldsymbol{M}^e_{n+1}\right] \frac{1}{\Delta t_n}\left[\boldsymbol{B}^e_{u2}\right]\, dV, \\
\left[\frac{\partial \boldsymbol{r}^e_{1,n+1}}{\partial \boldsymbol{v}_n}\right] &= -\frac{4}{\Delta t_n}\left[\boldsymbol{M}^e_u\right], & \left[\frac{\partial \boldsymbol{r}^e_{2,n+1}}{\partial \boldsymbol{v}_n}\right] &= \int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \theta^e_{n+1} \left[\boldsymbol{M}^e_{n+1}\right] \left[\boldsymbol{B}^e_{u2}\right]\, dV, \\
\left[\frac{\partial \boldsymbol{r}^e_{1,n+1}}{\partial \boldsymbol{\theta}_n}\right] &= \left[\boldsymbol{0}\right], & \left[\frac{\partial \boldsymbol{r}^e_{2,n+1}}{\partial \boldsymbol{\theta}_n}\right] &= -\int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \frac{c^k_{n+1}}{\Delta t_n}\left[\boldsymbol{N}^e_t\right]\, dV.
\end{aligned} \tag{4.9}$$

Assemble Equations (4.9) over the entire domain, and the global matrix forms of the sensitivities terms

$$\left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_n}\right], \qquad \left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{v}_n}\right], \qquad \left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_n}\right] \tag{4.10}$$

are obtained.

For the so-called history variables, the sensitivities $\frac{\partial \boldsymbol{r}_{i,n+1}}{\partial \boldsymbol{u}^{his}_{n+1}}$, $\frac{\partial \boldsymbol{u}^{his}_{n+1}}{\partial \boldsymbol{\mu}}$, $\frac{\partial \boldsymbol{r}_{i,n+1}}{\partial \boldsymbol{\theta}^{his}_{n+1}}$ $\frac{\partial \boldsymbol{\theta}^{his}_{n+1}}{\partial \boldsymbol{\mu}}$, $i = 1, 2$ are calculated locally within each element to take advantage of their data structures.

Recall that $\boldsymbol{u}^{his,e}_{n+1}$ for an element is created when it is first added and contains the displacement at all vertices. The global matrix $\boldsymbol{u}^{his}_{n+1}$ thus is stored as a two-dimensional array with the dimension of $\{nel, ndim \times nvrt\}$, where nel is the number of elements, $ndim$ is the dimension of the displacement variable and $nvrt$ is the number of vertices for each element. It is then easier to calculate the history variable sensitivities at the element level and then assemble into the global matrices. At the element level, $\frac{\partial \boldsymbol{r}^e_{i,n+1}}{\partial \boldsymbol{u}^{his}_{n+1}}$, $\frac{\partial \boldsymbol{r}^e_{i,n+1}}{\partial \boldsymbol{\theta}^{his}_{n+1}}$ $i = 1, 2$, can be calculated similarly in matrix form by differentiating the fully discretized form in Equations (2.63) and (2.64)

$$\begin{aligned}
\left[\frac{\partial \boldsymbol{r}^e_{1,n+1}}{\partial \boldsymbol{u}^{his}_{n+1}}\right] &= -\left[\boldsymbol{K}^e_{u,n+1}\right] \\
\left[\frac{\partial \boldsymbol{r}^e_{1,n+1}}{\partial \boldsymbol{\theta}^{his}_{n+1}}\right] &= \int_{\Omega^e_0} \left[\boldsymbol{B}^e_u\right]^T \left[\left(\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\theta}^{his}_{n+1}}\right)^e_{n+1}\right] \left[\boldsymbol{N}^e_t\right]\, dV \\
\left[\frac{\partial \boldsymbol{r}^e_{2,n+1}}{\partial \boldsymbol{u}^{his}_{n+1}}\right] &= \int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \left[\left(\frac{\partial c}{\partial \boldsymbol{u}}\right)^e_{n+1}\right] \dot{\theta}^e_{n+1}\left[\boldsymbol{B}^e_{u2}\right]\, dV - \\
&\qquad \int_{\Omega^e_0} \left[\boldsymbol{N}^e_t\right]^T \theta^e_{n+1} \left[\left(\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{\epsilon}}\right)^e_{n+1}\right] \left[\dot{\boldsymbol{\epsilon}}^e_{n+1}\right] \left[\boldsymbol{B}^e_{u2}\right]\, dV \\
\left[\frac{\partial \boldsymbol{r}^e_{2,n+1}}{\partial \boldsymbol{\theta}^{his}_{n+1}}\right] &= \left[\boldsymbol{0}\right].
\end{aligned} \tag{4.11}$$

Equations (4.11) are the element-level fully discretized sensitivities for the history variables.

Assuming now without losing substantial generality, that the dimension of $\boldsymbol{\mu}$ is one, the element level matrix form of

$$\left[\frac{\partial \boldsymbol{u}_{n+1}^{his,e}}{\partial \boldsymbol{\mu}}\right], \left[\frac{\partial \theta_{n+1}^{his,e}}{\partial \boldsymbol{\mu}}\right] \tag{4.12}$$

are arranged in order as

$$\left[\frac{\partial \boldsymbol{u}_{n+1,1}^{his,e}}{\partial \mu} \quad ... \quad \frac{\partial \boldsymbol{u}_{n+1,nvrt}^{his,e}}{\partial \mu}\right]^T, \left[\frac{\partial \theta_{n+1,1}^{his,e}}{\partial \mu} \quad ... \quad \frac{\partial \theta_{n+1,nvrt}^{his,e}}{\partial \mu}\right]^T. \tag{4.13}$$

The overall length of the vectors are $ndim \times nvrt$ and $nvrt$, respectively. If more than one sensitivity parameters are considered, instead of vectors, matrices can be formed in lieu of (4.13). The entries of the vectors in (4.13) $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\mu}}$ at the time step when a particular element is added. Denote on the element level

$$\begin{aligned}
\left[\left(\frac{\partial \boldsymbol{r}_1}{\partial \boldsymbol{\mu}_{his}}\right)_{n+1}^{e}\right] &= \left[\frac{\partial \boldsymbol{r}_{1,n+1}^{e}}{\partial \boldsymbol{u}_{n+1}^{his}}\right]\left[\frac{\partial \boldsymbol{u}_{n+1}^{his,e}}{\partial \boldsymbol{\mu}}\right] + \left[\frac{\partial \boldsymbol{r}_{1,n+1}^{e}}{\partial \boldsymbol{\theta}_{n+1}^{his}}\right]\left[\frac{\partial \theta_{n+1}^{his,e}}{\partial \boldsymbol{\mu}}\right], \\
\left[\left(\frac{\partial \boldsymbol{r}_2}{\partial \boldsymbol{\mu}_{his}}\right)_{n+1}^{e}\right] &= \left[\frac{\partial \boldsymbol{r}_{2,n+1}^{e}}{\partial \boldsymbol{u}_{n+1}^{his}}\right]\left[\frac{\partial \boldsymbol{u}_{n+1}^{his,e}}{\partial \boldsymbol{\mu}}\right] + \left[\frac{\partial \boldsymbol{r}_{2,n+1}^{e}}{\partial \boldsymbol{\theta}_{n+1}^{his}}\right]\left[\frac{\partial \theta_{n+1}^{his,e}}{\partial \boldsymbol{\mu}}\right].
\end{aligned} \tag{4.14}$$

Assembling over the entire domain the two vectors in Equation (4.14), the discretized vector form of

$$\left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_{n+1}^{his}}\frac{\partial \boldsymbol{u}_{n+1}^{his}}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_{n+1}^{his}}\frac{\partial \boldsymbol{\theta}_{n+1}^{his}}{\partial \boldsymbol{\mu}}\right] \tag{4.15}$$

in Equation (4.7) is obtained. Similarly we may express the state sensitivities $\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}}$, $\frac{\partial \boldsymbol{v}_n}{\partial \boldsymbol{\mu}}$, $\frac{\partial \boldsymbol{\theta}_n}{\partial \boldsymbol{\mu}}$ in vector form. For instance, using $\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}}$ as an example, with

$$\left[\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}}\right] = \left[\frac{\partial \boldsymbol{u}_{n,1}}{\partial \mu} \quad ... \quad \frac{\partial \boldsymbol{u}_{n,nnod}}{\partial \mu}\right]^T, \tag{4.16}$$

where $nnod$ is the number of nodes in the entire mesh. Multiplying the global sensitivity matrices from Equation (4.10) with the known sensitivities from previous time steps in the form of (4.16), another part of Equation (4.7) is acquired.

$$\left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{u}_n}\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{v}_n}\frac{\partial \boldsymbol{v}_n}{\partial \boldsymbol{\mu}} + \frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{\theta}_n}{\partial \boldsymbol{\mu}}\right]. \tag{4.17}$$

Once the parameters are specified, $\left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\mu}}\right]$ can be computed directly as well. This framework can be applied potentially to any time-step independent and differentiable parameters. Here,

it is applied to the convection coefficient h, a constant throughout the simulation. As a result, a rather straightforward direct dependence of shape error on the parameter is presented as

$$\begin{aligned}\left[\frac{\partial \boldsymbol{r}^e_{1,n+1}}{\partial h}\right] &= \quad [\mathbf{0}]\,, \\ \left[\frac{\partial \boldsymbol{r}^e_{2,n+1}}{\partial h}\right] &= \int_{\Gamma^t_{N,0}\cap\partial\Omega^e_0} [\boldsymbol{N}^e_t]^T \left(\left[\hat{\boldsymbol{\theta}}_\infty\right] - \left[\hat{\boldsymbol{\theta}}^e_{n+1}\right]\right)\, dA,\end{aligned} \tag{4.18}$$

where $\boldsymbol{\theta}^e_{n+1}$ is the element-level nodal temperature solution vector at time step t_{n+1} after meeting the iteration stopping criterion. Assembling Equation (4.18) over the entire domain and the global vector of $\left[\frac{\partial \boldsymbol{r}_{n+1}}{\partial \boldsymbol{\mu}}\right]$ is obtained. Once all the matrix-vector forms of the terms in (4.7) have been obtained, the state sensitivities at time step t_{n+1} can be solved for both displacement and temperature.

Given the shape error of (4.2) as the objective function, the sensitivity with respect to the convection coefficient is a scalar and many optimization methods work well with this case. Initial conditions for $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\mu}}$ are set to 0 as they do not depend on the convection coefficient at the starting time t_0.

The analytical sensitivities are tested against second-order centered finite difference derivatives to ensure correct derivation. A simple two-dimensional wall example as given in Chapter 3 is run with 40 elements in the x-direction and 30 in the y-direction for $240s$ after the wall is completely built. Analytical sensitives for both $\boldsymbol{u}_{n+1}$ and $\boldsymbol{\theta}_{n+1}$ match well with the ones computed through finite difference calculations based on

$$\begin{aligned}\frac{\partial \boldsymbol{u}_{n+1}(\mu_i)}{\partial \mu_i} &\doteq \frac{\boldsymbol{u}_{n+1}(\mu_i+\Delta\mu_i) - \boldsymbol{u}_{n+1}(\mu_i-\Delta\mu_i)}{2\Delta\mu_i} \\ \frac{\partial \boldsymbol{\theta}_{n+1}(\mu_i)}{\partial \mu_i} &\doteq \frac{\boldsymbol{\theta}_{n+1}(\mu_i+\Delta\mu_i) - \boldsymbol{\theta}_{n+1}(\mu_i-\Delta\mu_i)}{2\Delta\mu_i},\end{aligned} \tag{4.19}$$

where μ_i is the ith component of the parameter $\boldsymbol{\mu}$.

4.4 Optimization Methods

In this section some viable optimization algorithms for both discrete and continuous parameters are discussed.

4.4.1 Gradient-less Optimization Method

Consider first gradient-less optimization methods as number of layers is a discrete and crucial parameter. Based on Polak [88], a version of the method of local variants is proposed as given in Algorithm 1. The algorithm starts with an initial point in the parameter space and takes a series of steps towards the optimal solution. The initial step size and starting point

are important, as the algorithm is proven to be able to reach a local minimum but is not guaranteed to reach a global one.
Here, ρ_0 gives the initial step size and $\boldsymbol{\mu}_0$ represents the initial position. The smallest pos-

Algorithm 1: Method of local variations

Select a $\boldsymbol{\mu}_0 \in \mathbb{R}^n$ such that the set $C(\boldsymbol{\mu}_0) = \{f_0(\boldsymbol{\mu}) \leqslant f_0(\boldsymbol{\mu}_0)\}$ is bounded. Select a $\rho_0 > 0$.
Set $i = 0$, $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ and compute $f_0(\boldsymbol{\mu})$. Let $f_{min} = f_0(\boldsymbol{\mu})$, $\boldsymbol{\mu}_n = \boldsymbol{\mu}$.
while $\rho_i > \rho_{min}$ **do**
    Set $\rho = \rho_i$
    for $j = 0; j < 2n; j++$ **do**
        Compute $f_0(\boldsymbol{\mu} + \rho \boldsymbol{d}_j)$
        if $f_0(\boldsymbol{\mu} + \rho \boldsymbol{d}_j) < f_{min}$ **then**
            $f_{min} = f_0(\boldsymbol{\mu} + \rho \boldsymbol{d}_j)$
            $\boldsymbol{\mu}_n = \boldsymbol{\mu} + \rho \boldsymbol{d}_j$
    if $f_{min} < f_0(\boldsymbol{\mu})$ **then**
        $\boldsymbol{\mu} = \boldsymbol{\mu}_n$, $i+ = 1$
    else
        $\rho_i / = 2$ $(\rho_i - = 1)$

sible step size is indicated by ρ_{min} and can be extended to a vector if required by different parameters. At each iteration, the algorithm starts from the current point z in the parameter space and runs $2n$ simulations on neighboring points, n being the dimension of the parameter space, with the neighboring points in the parameter space given by the step size ρ and directions $\boldsymbol{d}_j$. If any of the neighboring parameter combinations produce a smaller objective, the parameters that produce the smallest one are selected and the algorithm moves to that point. If no such point can be found, the step-size is reduced and new neighboring points are computed until the step-size becomes smaller than ρ_{min}. The quantity $\boldsymbol{d}_j$ are the search directions with one positive and one negative for each parameter dimension. The version given in Algorithm 1 is the one for continuous parameters where the step-size can be repeatedly reduced by half. In the case of integer parameter like the number of layers, we reduce the step-size $|\rho \boldsymbol{d}_j|$ by 1 subsequently.

The method of local variants is first applied to the two-dimensional wall problem introduced in Section 3.4.2. The shape error is optimized over two parameters, namely layer thickness and printing speed. The inequality constraints in Equation (4.4) are specified to be the maximum and minimum values with practical requirements on the two parameters. Specifically, the constraints on layer thickness Δy and printing speed $\frac{\Delta x}{\Delta t}$ are

$$\begin{aligned} 0.2mm \leqslant \Delta y \leqslant 0.35mm \\ 50mm/s \leqslant \frac{\Delta x}{\Delta t} \leqslant 100mm/s. \end{aligned} \tag{4.20}$$

The parameter space for number of layers and printing speed is a well-defined rectangular one upon which the method of local variants is applied, as shown in Figure 4.12. The initial step size is set to 3 for number of layers and $5mm/s$ for the speed. The starting point in the parameter space is set to $n_y = 40$, which corresponds to $0.25mm$ for Δy and $v = 75mm/s$. Here n_y denotes the number of layers and v the speed.

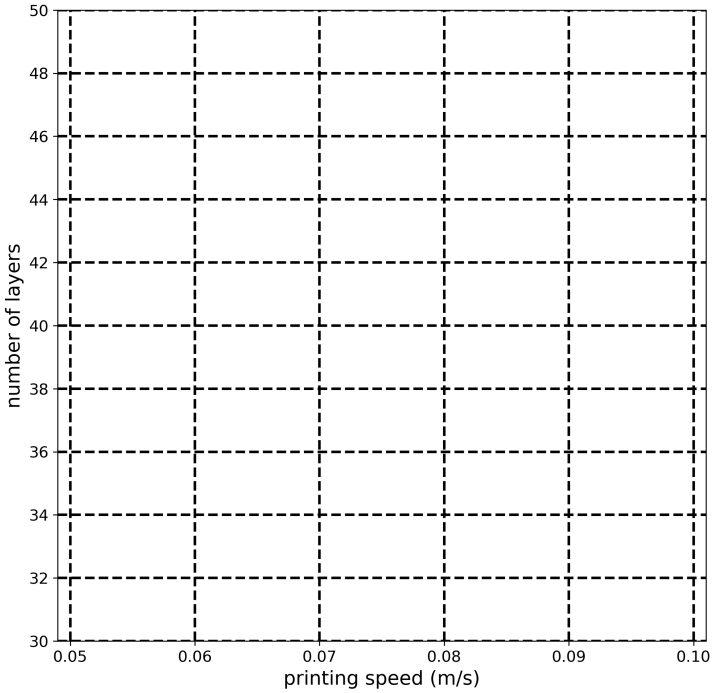


Figure 4.12: Two-dimensional wall parameter space

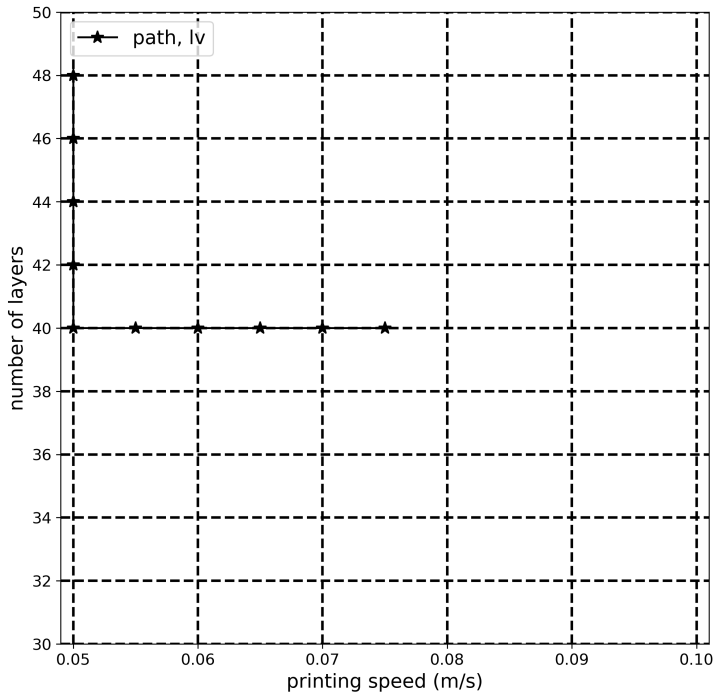


Figure 4.13: Two-dimensional wall optimization path with method of local variants

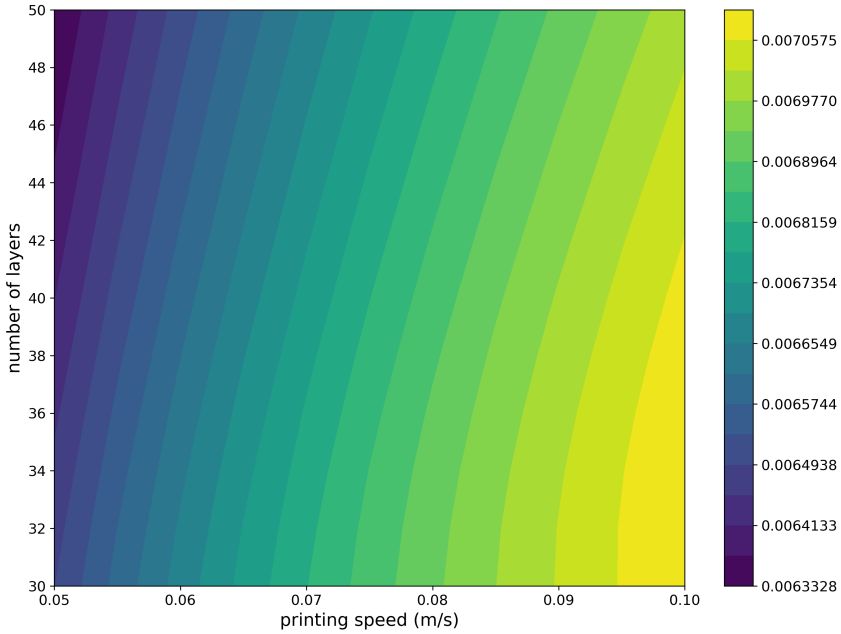


Figure 4.14: Two-dimensional wall: filled contour plot of the objective function with respect to number of layers and printing speed

The simple constant constraints considered are not coupled with the objective calculations and therefore do not interfere with the method of local variants itself. Rather, at each step, after a new potential point in the parameter space is chosen by the algorithm, it is checked against the constraints for viability. If the potential step is out of bound, it will be rejected. A smaller step size that lands a new potential point right on the parameter space boundary would be selected as one of the candidates for the algorithm instead.

The optimization path generated by the algorithm follows the rule of slower speed and larger number of layers n_y and eventually stops at the boundary governed by the constraints as shown in Figure 4.13. The optimization result translates to printing more slowly and slicing the wall more finely in order to reduce the shape error, both of which are intuitively plausible. Meanwhile, as a trade-off it takes longer to manufacture the wall with more layers.

In the wall case the geometric error without any deformation is zero, therefore the thermomechanical response during the building process is responsible for the entirety of the shape error. The proposed method successfully converges to the global minimum for the objective function within the constraints. Figures 4.14 and 4.15 are the filled contour plot and surface plot of the objective function with respect to the two parameters, respectively. Evident in the figures is the smoothness of the objective function which leads to successful convergence of the optimization procedure. Further, the objective function is monotonic in both parameter directions within the given range, making it easy to find neighboring points with a smaller objective function value.

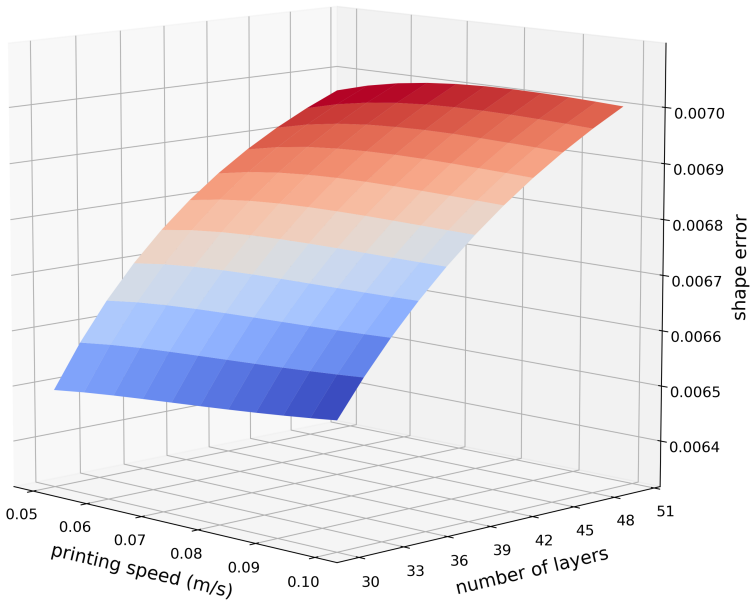


Figure 4.15: Two-dimensional wall: surface plot of the objective function with respect to number of layers and printing speed

Next the method is applied to the problem of the two-dimensional wall with quarter-circular hole described in Section 3.4.2. The constraints are set to

$$\begin{aligned} 0.2083mm \leqslant \Delta y \leqslant 0.35mm \\ 50mm/s \leqslant \frac{\Delta x}{\Delta t} \leqslant 100mm/s. \end{aligned} \tag{4.21}$$

The layer thickness $0.2083mm$ corresponds to 48 layers. The optimization path is shown in Figure 4.16 where the initial step-size is $5mm/s$ for printing speed and 4 for the number of layers. Starting from $(0.075m/s, 36)$ ($\Delta y = 0.28$ mm), the optimization method creates a path to $(0.075m/s, 40)$ followed by continuously smaller speed until the boundary is hit at $(0.05m/s, 40)$. The step-size keeps decreasing to the smallest threshold (1 for number of layers) as no neighboring point produces smaller objective function values and eventually the optimization stops. Therefore, the optimal parameters given by the method of local variants therefore are $(0.05m/s, 40)$ which is the optimal solution, as can be seen in the filled contour plot in Figure 4.17.

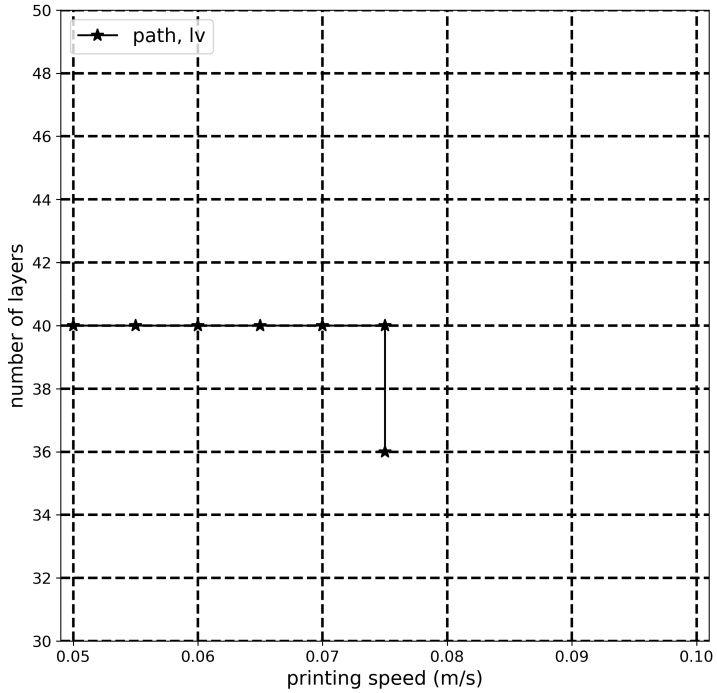


Figure 4.16: Two-dimensional wall with hole: optimization path with method of local variations

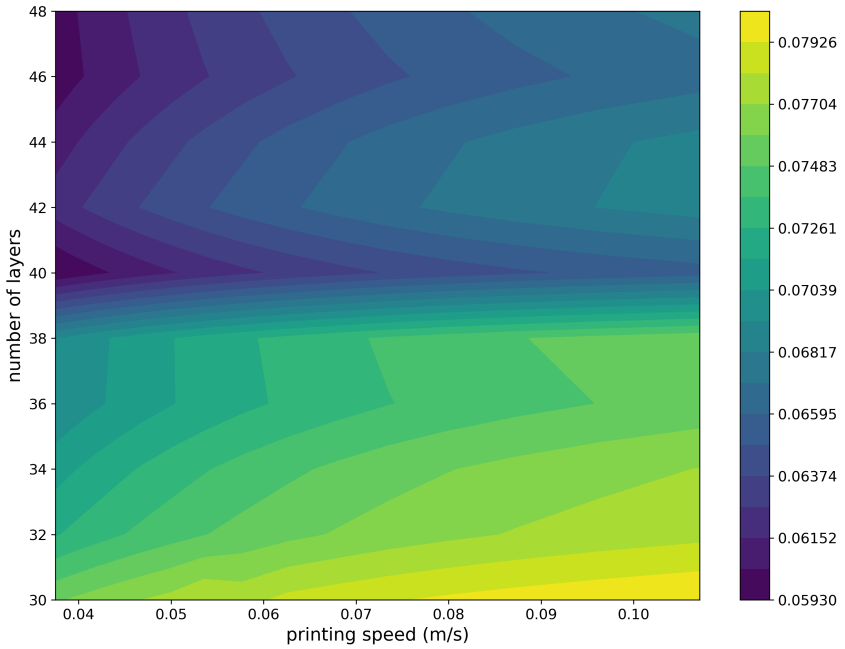


Figure 4.17: Two-dimensional wall with hole: filled contour plot of the objective function with respect to number of layers and printing speed

The geometric error no doubt plays a big role since the objective is no longer monotonic with respect to layer thickness as discussed in Section 4.2. Such error is intrinsic to the layered manufacturing process and is biased towards more layers with local minimums and maximums throughout the domain. Hence the minimum within the constrained domain might no longer be at the maximum number of layers on the boundary, as it turns out to be with the choice of constraints. The geometric error patterns are inherited in the final shape error even with the thermomechanical building process accounting for a bigger source of the shape error. This is evident in their objective function surface plots in Figure 4.18 and Figure 4.19. The pure geometric error matches the observed results in Figure 4.10. With a bigger initial choice of step size and a reasonable or rather lucky choice of initial parameters, the optimization method manages to advance to the correct solution.

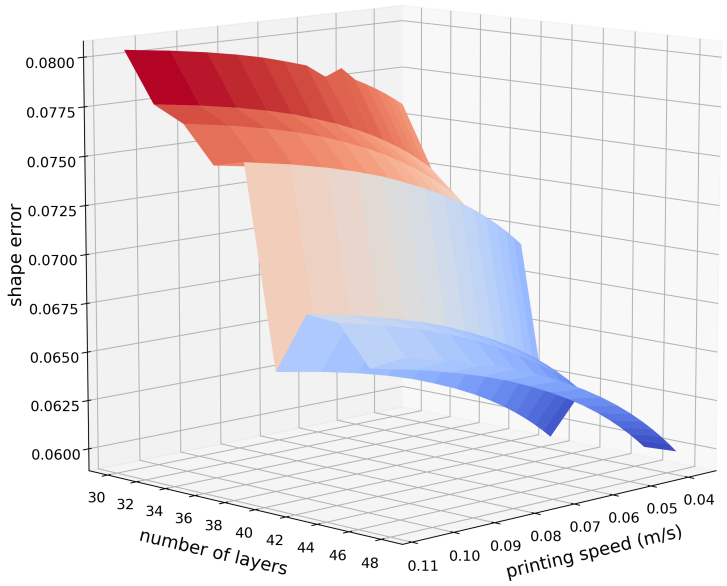


Figure 4.18: Two-dimensional wall with hole: surface plot of the objective function with respect to number of layers and printing speed

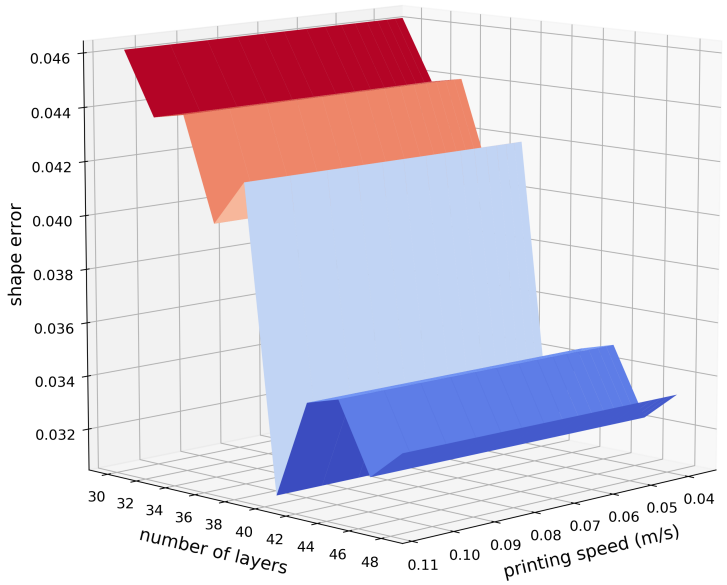


Figure 4.19: Two-dimensional wall with hole: surface plot of the objective function with respect to number of layers and printing speed with no deformation

However, if the initial starting point is different at $(0.075m/s, 44)$, with initial step size of 2 for number of layers, the optimization path ends up at a different point and the path is shown in Figure 4.20, arriving at a local minimum of $(0.075m/s, 46)$ which produces larger objective function value than the previous result. In this case, the method of local variants fails to converge to a global minimum.

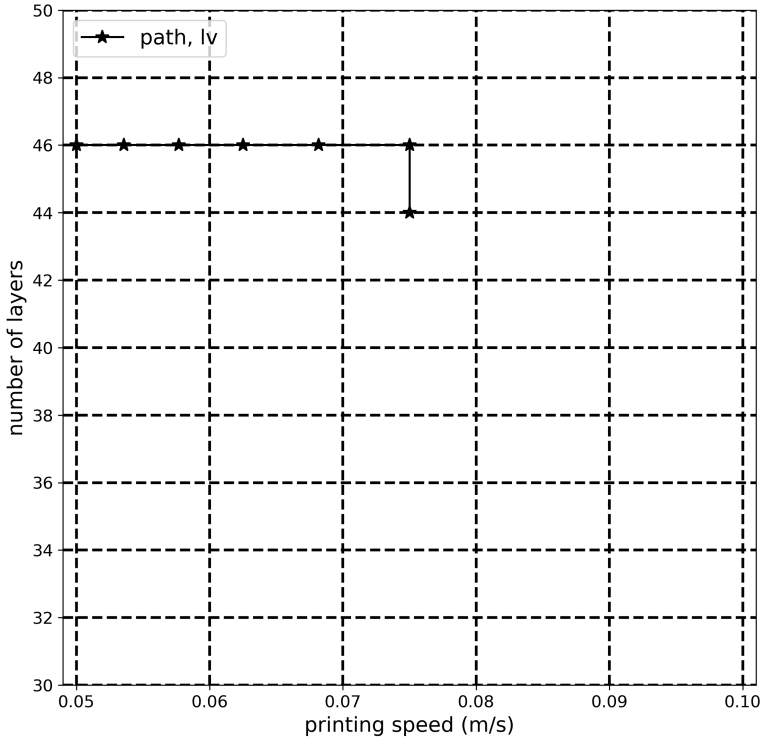


Figure 4.20: Two-dimensional wall with hole optimization path with method of local variations from another initial position

In order to avoid being stuck on a local minimum with a relatively global large value, several steps can be taken. First, multiple starting points in the parameter space can be chosen and the best result out of their optimization solution gets selected. Second, a bigger initial step-size can be employed. This works based on the observation that the local maxima and minima generated by the geometric error do not disrupt the overall trend of the objective function and can be overcome by a reasonably large step-size.

Regardless, to our knowledge there exists no rigorous proof when such discrete parameter and subsequent non-smooth objective function can converge quickly with the method of local variants, therefore this method must be employed carefully depending on the specific parameter being considered. Consequently, no particular algorithm is proposed for discrete parameters beyond the number of layers which has been discussed in details in this section. On the other hand, if differentiable parameters with gradients are considered, the vast and well-researched optimization methods [89] can be tapped to help solve the problem.

4.4.2 Fixed Discrete Parameter with Gradient-based Optimization

As seen in Section 4.4.1, problems with curved geometries cause the number of layers parameter to behave undesirably with respect to the objective function and increase the difficulty in finding a global minimum. Hence, another set of optimization approaches is explored where the discrete parameter is fixed and the continuous parameters are optimized first. Algorithm 2 provides the details, where μ denotes the single discrete parameter and $\boldsymbol{y}$ are the continuous ones. The parameters are constrained, with μ_{min} and μ_{max} representing the initial range on the discrete parameter while $\boldsymbol{y}_{max}$ and $\boldsymbol{y}_{min}$ for continuous ones. Taking advantage of parallel processing, optimization on the continuous parameters is run at m distinct fixed discrete parameter values at the same time, with m being the number of processor cores used. The search range of the discrete parameter is then narrowed by comparing the optimized results and selecting the two smallest value of objective function produced. If the maximum or minimum values of the discrete parameter happen to give the two smallest objectives, the range is not narrowed but different values within the range are chosen and run with gradient-based continuous optimization until either a smaller range can be found or the search exhausts all internal values and the boundary values are thus picked.

Algorithm 2: Discrete parameter search

For the discrete parameter μ and m processors, set μ_{min} and μ_{max} based on physics.
  $s = 0$
while $\mu_{max} - \mu_{min} > 1$ **do**
  $l = \mu_{max} - \mu_{min}$
  if $s >= l/(m-1)$ **then**
    // Exhausted all values within the range, minimums at μ_{max} or μ_{min}
    Break
  Select m $\mu_i \in \mathbb{R}, i = 0, 1, 2, 3...m-1$ and
   $\mu_i = i * (\mu_{max} - \mu_{min})/(m-1) + \mu_{min} + s$,
  For each μ_i, run optimization on the remaining parameters $\boldsymbol{y}$ with
   gradient-based algorithm
  Compare $f_0(\mu_i, \boldsymbol{y})$, where $\boldsymbol{y}$ are optimized for μ_i. Choose two μ_i that give the
   smallest f values and set them to be the new μ_{min} and μ_{max}.
  if $\mu_{max} - \mu_{min} = l$ **then**
    s += 1
  else
    s = 0
  if $\mu_{max} - \mu_{min} < m$ **then**
    $m = \mu_{max} - \mu_{min}$.
Optimal solution is either μ_{max} or μ_{min}

The algorithm here is designed based on the gained knowledge of how the discrete parameter, in this case the number of layers would affect the shape-error objective function. The main assumption is that the local minima and maxima will not change the general direction of the shape error's declining with increasing layers. This should allow the algorithm to narrow the search range of the number of layers to a relatively small one and avoid the high cost of an exhaustive search. The algorithm would still fail to efficiently find a steep global minimum where its immediate neighbors produce much higher values without having to search through all number of layers.

The parallelized code is implemented with Message Passing Interface (MPI). Building on the work in Section 4.3, gradient-based optimization methods are explored for the continuous parameters. The optimality conditions for constrained problems as given in Equation (4.4) based on optimization theory requires solving the Karush–Kuhn–Tucker (KKT) conditions [89]. Dealing with a small number of parameters in the current work, adjoint variables are not used in the calculations here. Instead the optimality condition is written in a straightforward way as

$$\frac{\partial f_0}{\partial \boldsymbol{U}} \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\mu}} + \frac{\partial f_0}{\partial \boldsymbol{\mu}} = 0, \tag{4.22}$$

where $\boldsymbol{U}$ represents the coupled unknown variables that include both displacement $\boldsymbol{u}$ and temperature $\boldsymbol{\theta}$. To get to the optimal solution, iterative methods are often used. Since second-order derivatives are unavailable while only a small number of parameters are being optimized, we opt for gradient descent method with line search to find the path to the optimal solution. The search direction given by gradient descent is $-\frac{df_0}{d\boldsymbol{\mu}}$ and the algorithm is given in Algorithm 3 where a popular and effective backtracking approach is used in the line search [89, 90]. The line search terminates when the sufficient decrease Wolfe condition, also known as the Armijo rule, is satisfied.

$$f_0\left(\boldsymbol{\mu}_k + \alpha p_k\right) \leqslant f_0\left(\boldsymbol{\mu}_k\right) + c\alpha \left[\frac{df_0\left(\boldsymbol{\mu}_k\right)}{d\boldsymbol{\mu}_k}\right]^T \boldsymbol{p}_k. \tag{4.23}$$

Algorithm 3: Gradient descent with line search

for $k = 0, 1, 2, \ldots$ **do**
  $\boldsymbol{p}_k = -\frac{df_0(\boldsymbol{\mu}_k)}{d\boldsymbol{\mu}_k} / \left|\frac{df_0(\boldsymbol{\mu}_k)}{d\boldsymbol{\mu}_k}\right|$
  choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$
  Set $\alpha \leftarrow \bar{\alpha}$
  while $f_0\left(\boldsymbol{\mu}_k + \alpha \boldsymbol{p}_k\right) > f_0\left(\boldsymbol{\mu}_k\right) + c\alpha \left[\frac{df_0(\boldsymbol{\mu}_k)}{d\boldsymbol{\mu}_k}\right]^T \boldsymbol{p}_k$ **do**
    $\alpha \leftarrow \rho\alpha$
  $\alpha_k = \alpha$
  $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \boldsymbol{p}_k$

Here c is a constant chosen based on the problem, ρ is typically $\frac{1}{2}$ to half the step size each iteration and α marks the length of the step. The constraints considered here form simple rectangular boundaries in the parameter space. They are checked whenever a new parameter combination is reached. Optimization steps that result in breaching the boundary of the parameter space are either rejected or adjusted so the parameters land exactly on the boundary. For more complicated constraints which are dependent on both the process and the parameters, this algorithm would not suffice.

This optimization method is first applied to two-dimensional wall building with the number of layers and the convection coefficient h as parameters. The width of the wall is $20mm$ and the height is $10mm$ with 40 elements in the x direction and print speed of $71.4mm/s$. The constraints on the two parameters are given as

$$\begin{aligned} 0.20mm \leqslant \Delta y \leqslant 0.35mm, \\ 20W/m^2K \leqslant h \leqslant 40W/m^2K. \end{aligned} \tag{4.24}$$

The parameter space in this case is shown in Figure 4.21.

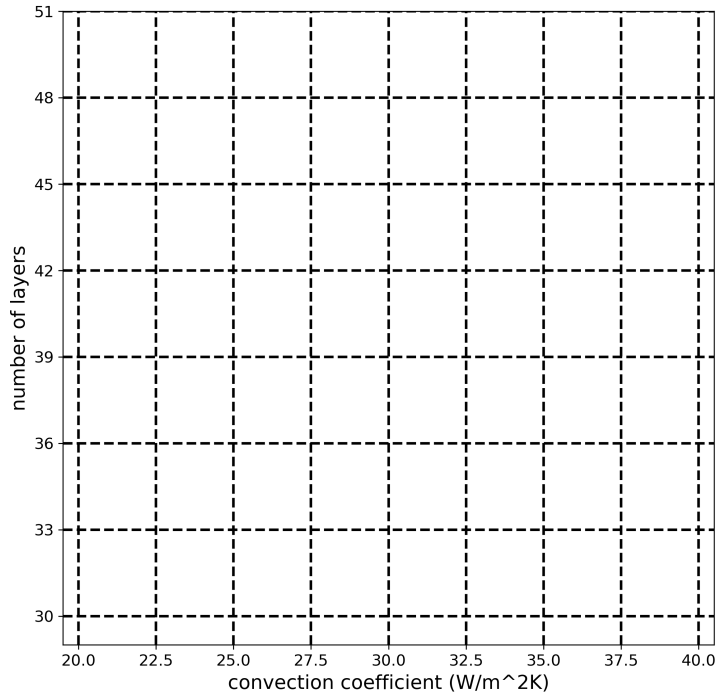


Figure 4.21: Two-dimensional wall parameter space with convection coefficient

Taking a closer look at an example step where the parameters are $(30W/m^2K, 30)$, note that based on derivations in Section 4.3, the gradient $\frac{df_0}{dh}$ is found to be -1.33493×10^{-5}. The step is taken therefore based on the algorithm with $\bar{\alpha} = 1$, $\rho = 0.5$ and $c = 0.1$. Through simulation $f_0(31, 30) = 0.00710$. Together with $f_0(30, 30) = 0.00718$, $\alpha = 1$ already satisfies

the Wolfe condition. The step length is then set at 1 and the next convection coefficient is chosen to be 31.

Applying the line search gradient descent algorithm to five different number of layers in parallel, optimized convection coefficients can be obtained for each of them separately. The range for number of layers is set as $(30, 50)$ initially at the constraint boundary and layer numbers $30, 34, 40, 46, 50$ are chosen as the z_i in Algorithm 2. The optimized parameters in the parameter space are shown in Figure 4.22 marked with objective function values.

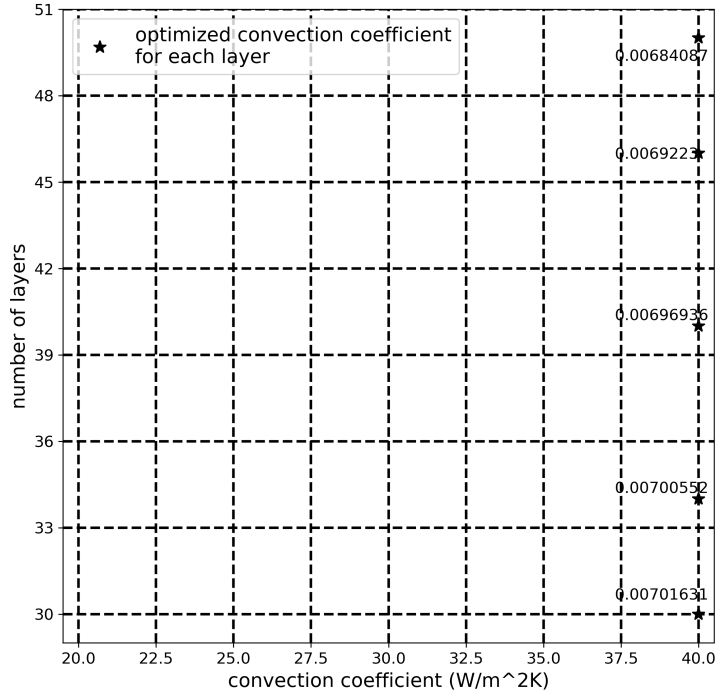


Figure 4.22: Two-dimensional wall optimization path with convection coefficient at fixed number of layers

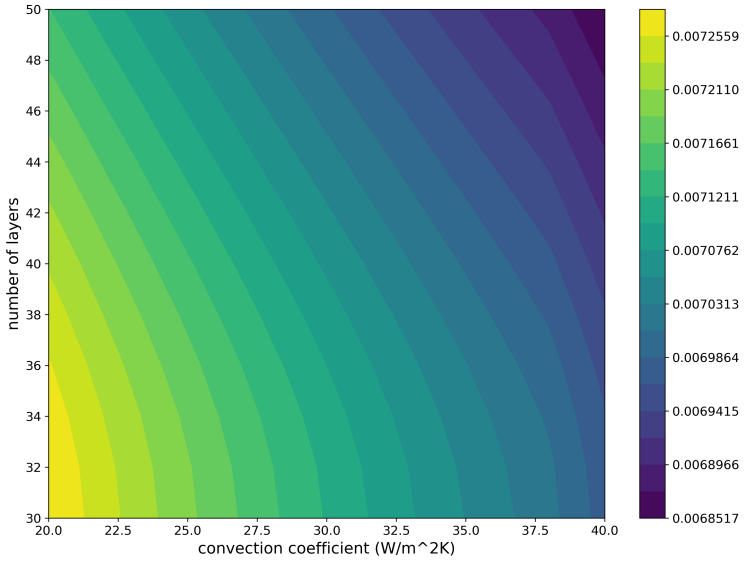


Figure 4.23: Two-dimensional wall optimization: filled contour plot of number of layers and convection coefficient

It is clear then that the cases of 46 and 50 layers provide the smallest shape errors and thus become the new upper and lower bound for the number of layers parameter. Optimization is run for $47, 48, 49$ layers and by comparing results among them eventually the algorithm arrives at the optimal solution $(40W/m^2K, 50)$. The filled contour plot and the surface plot are shown in Figures 4.23 and 4.24, respectively. The smooth surface and its monotonic relation with both parameters again explain why the optimal solution appears at the boundary. The algorithm successfully converges to the optimal solution.

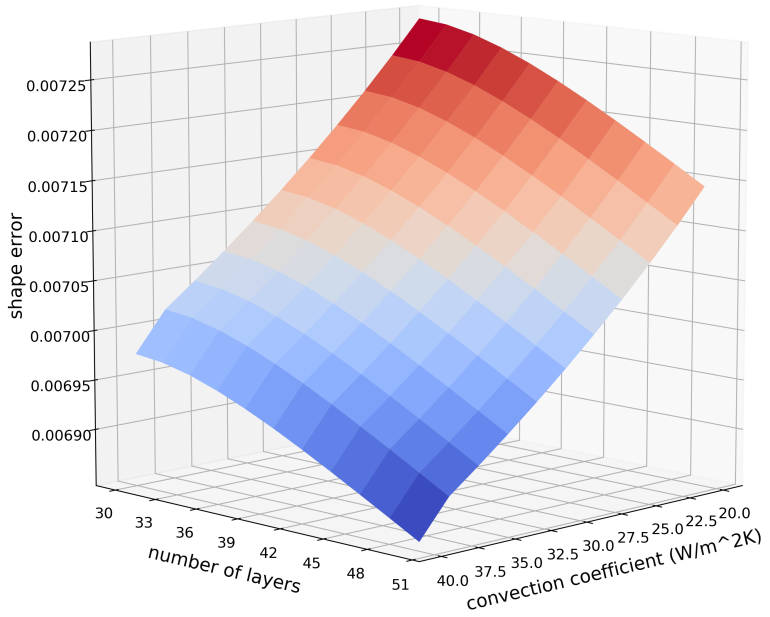


Figure 4.24: Two-dimensional wall optimization: surface plot of number of layers and convection coefficient

The optimization algorithm is also applied to the problem of the two-dimensional wall with a quarter-circular hole with the number of layers and convection coefficient as optimization parameters. The width of the wall is again $15mm$ and the height is $10mm$ with 40 elements in the x-direction for a full layer and $3mm$ radius quarter-circle hole at the top right, as in Chapter 3. The printing velocity is fixed at $85mm/s$ within the admissible range. The constraints on the optimization parameters are given as

$$\begin{aligned} 0.2083mm \leqslant \Delta y \leqslant 0.35mm, \\ 20W/m^2K \leqslant h \leqslant 40W/m^2K. \end{aligned} \tag{4.25}$$

An example optimization step is described next with parameters $(30W/m^2K, 30)$. The analytically calculated gradient with the continuous variable $\frac{df_0}{dh}$ is -3.213×10^{-5}. Based on the algorithm, $f_0(31, 30)$ is taken as a trial step and its value is 0.07995, while $f_0(30, 30)$ is 0.07999. Therefore $\alpha = 1$ satisfies the stopping criterion again and the next step in the optimization path is $(31W/m^2K, 30)$. Continuing the process, for the five cases of numbers of layers 30, 34, 40, 44, 48 chosen within the $(30, 48)$ range, an optimized convection coefficient is found for each one, as shown in Figure 4.25.

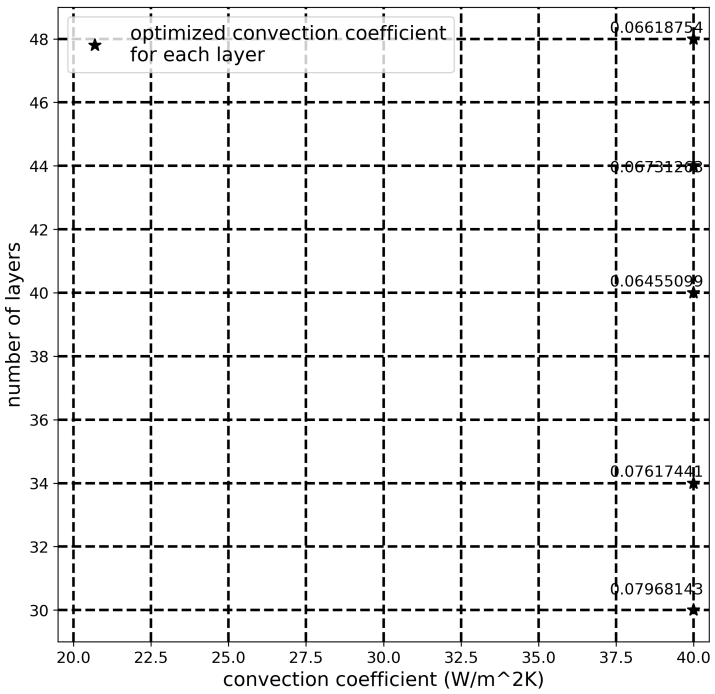


Figure 4.25: Two-dimensional wall with hole: optimization path with convection coefficient 1

Next, the process is repeated for a narrowed range of number of layers in the $(40, 48)$ range and the optimization problem is solved for layer numbers $42, 44, 46$. The range narrows further down to $(40, 46)$ before it stops shrinking and the remaining numbers of layers $41, 43, 45$ are chosen and optimized. Eventually the optimization result is obtained at 40 layers and $40 W/m^2K$ as they give the smallest objective. The optimal solution again is not at maximum number of layers due to the geometric error of approximating curvature and the discreteness of the number of layers. It is shown in the filled contour plots in Figures 4.26 and 4.27 that indeed the minimum objective is reached within the constraints.

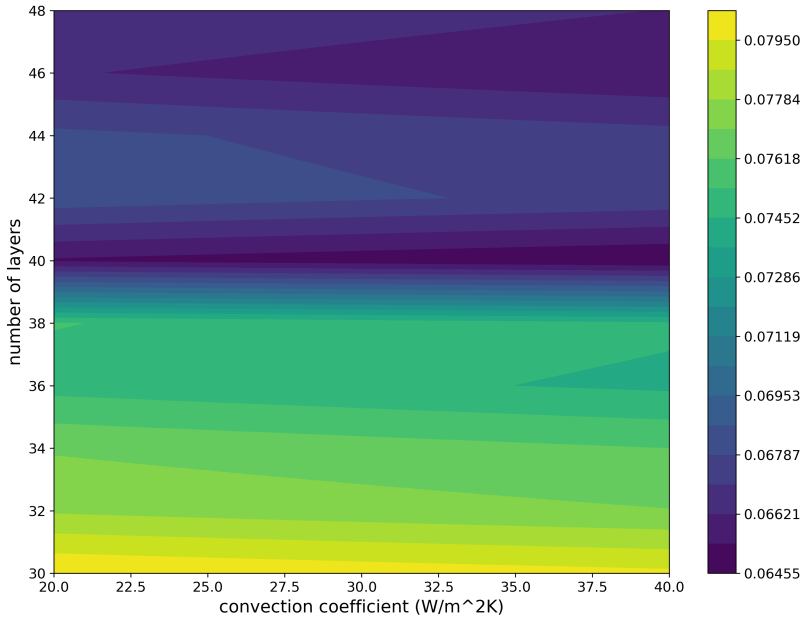


Figure 4.26: Two-dimensional wall with hole optimization: filled contour plot of number of layers and convection coefficient

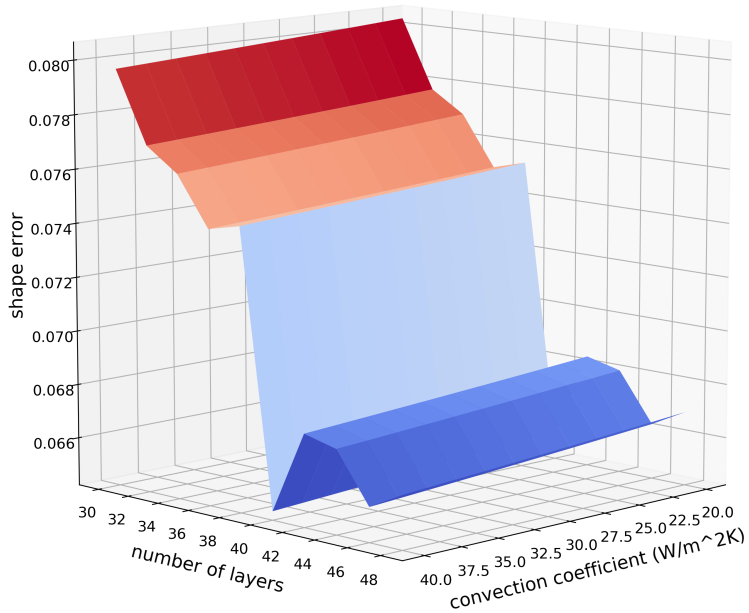


Figure 4.27: Two-dimensional wall with hole optimization: surface plot of number of layers and convection coefficient

The result here suggests that a larger convection coefficient, enabled by forced convection helps the material to cool more rapidly and reduces the eventual shape error developed during the printing and cooling process. Meanwhile, more layers in general would improve the dimensional quality by shrinking the geometric error. However, it is cautioned that cooling down faster does not necessarily leads to an improved geometry because it might cause the newly added elements and layers to be distorted due to large deformation from previous layers. Maintaining a temperature gradient during the printing process itself might lead to a superior result. Further research with a wider range of parameters is needed to address this. At the same time if every layer is cooled all the way down to ambient temperature before another layer is printed, the continuity assumption between different layers might no longer be a reasonable one.

Chapter 5

Conclusion

In this dissertation, a coupled thermomechanical finite element simulation model for fused deposition modeling, an additive manufacturing method is developed. The main contributions of this model are as follows:

- Simultaneous solutions of displacement and temperature to better understand the coupling effect and underlying physics behind the FDM process,
- Evolving geometry that accounts for deformation history during the printing process to better represent the printed geometry and more accurately capture mechanical behaviors,
- Complete implementation of the model in both Python and C++ with flexible constitutive models.

This simulation model is then applied to an optimization study of the process parameters on the shape error objective. To the best of our knowledge, parameterized optimization for additive manufacturing has not been studied in details before. This work contributes to the endeavor by:

- Establishing parameterized PDE-constrained optimization problem for FDM finite element model,
- Studying in detail the effect of number of layers on shape error, particularly the multifaceted geometric error that arises from both slicing algorithms and layer number choices,
- Deriving and computing sensitivities for continuous parameters to take advantage of well-known optimization methods,
- Proposing both gradient-less and gradient-based optimization algorithms and implementing them,

- Discussing optimal results from proposed algorithms and how they can lead to improved design for both the manufacturing and design processes.

There exists many potential areas for further work that would expand on the base model developed here. These include:

- Extension the model to 3D for real-life applications
- Study of additional optimization parameters like printing pattern
- Implementation of more detailed material models with matched experimental data.

Bibliography

[1] O.A. Mohamed, S.H. Masood, and J.L. Bhowmik. Optimization of fused deposition modeling process parameters: A review of current research and future prospects. *Advances in Manufacturing*, 3(1):42–53, March 2015.

[2] I. Gibson, D.W. Rosen, and B. Stucker. *Additive Manufacturing Technologies*. Springer, New York, 2014.

[3] C.K. Chua, K.F. Leong, and C.S. Lim. *Rapid prototyping: principles and applications (with companion CD-ROM)*. World Scientific Publishing Company, Singapore, 2010.

[4] S. Upcraft and R. Fletcher. The rapid prototyping technologies. *Assembly Automation*, 23(4):318–330, December 2003.

[5] S. Mansour and R. Hague. Impact of rapid manufacturing on design for manufacture for injection moulding. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 217(4):453–461, April 2003.

[6] N. Hopkinson, R.J. Hague, and P.M. Dickens. *Rapid manufacturing: An Industrial Revolution for the Digital Age*. John Wiley and Sons, England, 2006.

[7] B. Thavornyutikarn, N. Chantarapanich, K. Sitthiseripratip, G.A. Thouas, and Q. Chen. Bone tissue engineering scaffolding: Computer-aided scaffolding techniques. *Progress in Biomaterials*, 3(2-4):61–102, December 2014.

[8] S.H. Ahn, M. Montero, D. Odell, S. Roundy, and P.K. Wright. Anisotropic material properties of fused deposition modeling ABS. *Rapid Prototyping Journal*, 8(4):248–257, October 2002.

[9] M.M. Francois, A. Sun, W.E. King, N.J. Henson, D. Tourret, C.A. Bronkhorst, N.N Carlson, C.K. Newman, T.S. Haut, J. Bakosi J, and J.W. Gibbs. Modeling of additive manufacturing processes for metals: Challenges and opportunities. *Current Opinion in Solid State and Materials Science*, 21(LA-UR-16-24513):SAND–2017–6832J, January 2017.

[10] W. King, A.T. Anderson, R.M. Ferencz, N.E. Hodge, C. Kamath, and S.A. Khairallah. Overview of modelling and simulation of metal powder bed fusion process at Lawrence

Livermore National Laboratory. *Materials Science and Technology*, 31(8):957–968, June 2015.

[11] B. Schoinochoritis, D. Chantzis, and K. Salonitis. Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(1):96–117, January 2017.

[12] F. Hajializadeh and A. Ince. Short review on modeling approaches for metal additive manufacturing process. *Material Design and Processing Communications*, 2(2):e56, March 2019.

[13] S.J. Kalita, S. Bose, H.L. Hosick, and A. Bandyopadhyay. Development of controlled porosity polymer-ceramic composite scaffolds via fused deposition modeling. *Materials Science and Engineering: C*, 23(5):611–620, October 2003.

[14] P. Michaleris. Modeling metal deposition in heat transfer analyses of additive manufacturing processes. *Finite Elements in Analysis and Design*, 86:51–60, September 2014.

[15] E.R. Denlinger, J. Irwin, and P. Michaleris. Thermomechanical modeling of additive manufacturing large parts. *Journal of Manufacturing Science and Engineering*, 136(6), December 2014.

[16] E.R. Denlinger, M. Gouge, J. Irwin, and P. Michaleris. Thermomechanical model development and in situ experimental validation of the laser powder-bed fusion process. *Additive Manufacturing*, 16:73–80, August 2017.

[17] S. Kolossov, E. Boillat, R. Glardon, P. Fischer, and M. Locher. 3D FE simulation for temperature evolution in the selective laser sintering process. *International Journal of Machine Tools and Manufacture*, 44(2-3):117–123, February 2004.

[18] I.A. Roberts, C.J. Wang, R. Esterlein, M. Stanford, and D.J. Mynors. A three-dimensional finite element analysis of the temperature field during laser melting of metal powders in additive layer manufacturing. *International Journal of Machine Tools and Manufacture*, 49(12-13):916–923, October 2009.

[19] Y. Zhang and Y.K. Chou. Three-dimensional finite element analysis simulations of the fused deposition modelling process. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 220(10):1663–1671, October 2006.

[20] T. Keller T, G. Lindwall, S. Ghosh, L. Ma, B.M. Lane, F. Zhang, U.R. Kattner, E.A. Lass, J.C. Heigel, Y. Idell, and M.E. Williams. Application of finite element, phase-field, and calphad-based methods to additive manufacturing of Ni-based superalloys. *Acta Materialia*, 139:244–253, October 2017.

[21] Q. Yang, P. Zhang P, L. Cheng, Z. Min, M. Chyu, and A.C. To. Finite element modeling and validation of thermomechanical behavior of Ti-6Al-4V in directed energy deposition additive manufacturing. *Additive Manufacturing*, 12:169–177, October 2016.

[22] L. Costa, R. Vilar, T. Reti, and A.M. Deus. Rapid tooling by laser powder deposition: Process simulation using finite element analysis. *Acta Materialia*, 53(14):3987–3999, August 2005.

[23] D. Hu and R. Kovacevic. Modelling and measuring the thermal behaviour of the molten pool in closed-loop controlled laser-based additive manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 217(4):441–452, April 2003.

[24] L.B. Ji and T.R. Zhou. Finite element simulation of temperature field in fused deposition modeling. *Advanced Materials Research*, 97:2585–2588, 2010.

[25] Y. Zhang and Y.K. Chou. 3D FEA simulations of fused deposition modeling process. In *ASME 2006 International Manufacturing Science and Engineering Conference*, pages 1121–1128, January 2006.

[26] A. Foroozmehr, M. Badrossamay, E. Foroozmehr, and S.I. Golabi. Finite element simulation of selective laser melting process considering optical penetration depth of laser in powder bed. *Materials and Design*, 89:255–263, January 2016.

[27] A.P. West, S.P. Sambu, and D.W. Rosen. A process planning method for improving build performance in stereolithography. *Computer-Aided Design*, 33(1):65–79, January 2001.

[28] G. Vastola, G. Zhang, Q.X. Pei, and Y.W. Zhang. Controlling of residual stress in additive manufacturing of Ti-6Al-4V by finite element modeling. *Additive Manufacturing*, 12:231–239, October 2016.

[29] H.S Cho, W.S. Park, B.W. Choi, and M.C. Leu. Determining optimal parameters for sterolithography processes via genetic algorithm. *Journal of Manufacturing Systems*, 19(1):18, 2000.

[30] J. Gockel, J. Beuth, and K. Taminger. Integrated control of solidification microstructure and melt pool dimensions in electron beam wire feed additive manufacturing of Ti-6Al-4V. *Additive Manufacturing*, 1:119–126, October 2014.

[31] G. Strano, L. Hao, R.M. Everson, and K.E. Evans. Surface roughness analysis, modelling and prediction in selective laser melting. *Journal of Materials Processing Technology*, 213(4):589–597, April 2013.

[32] A.H. Nickel, D.M. Barnett, and F.B. Prinz. Thermal stresses and deposition patterns in layered manufacturing. *Materials Science and Engineering: A*, 317(1-2):59–64, October 2001.

[33] T.A. Krol, M.F. Zaeh, and C. Seidel. Optimization of supports in metal-based additive manufacturing by means of finite element models. *23rd Annual International Solid Freeform Fabrication Symposium - An Additive Manufacturing Conference, SFF 2012*, pages 707–718, 01 2012.

[34] A. Vasinonta, J.L. Beuth, and M.L. Griffith. A process map for consistent build conditions in the solid freeform fabrication of thin-walled structures. *Journal of Manufacturing Science and Engineering*, 123(4):615–622, August 2000.

[35] R. Anitha, S. Arunachalam, and P. Radhakrishnan. Critical parameters influencing the quality of prototypes in fused deposition modelling. *Journal of Materials Processing Technology*, 118(1-3):385–388, December 2001.

[36] T. Nancharaiah, D.R. Raju, and V.R. Raju. An experimental investigation on surface quality and dimensional accuracy of FDM components. *International Journal on Emerging Technologies*, 1(2):106–111, January 2010.

[37] D. Horvath, R. Noorani, and M. Mendelson. Improvement of surface roughness on abs 400 polymer using design of experiments (DOE). *Materials Science Forum*, 561:2389–2392, 2007.

[38] C.C. Wang, T.W. Lin, and S.S. Hu. Optimizing the rapid prototyping process by integrating the Taguchi method with the gray relational analysis. *Rapid Prototyping Journal*, 13(5):304–315, October 2007.

[39] K.P. Thrimurthulu, P.M. Pandey, and N.V. Reddy. Optimum part deposition orientation in fused deposition modeling. *International Journal of Machine Tools and Manufacture*, 44(6):585–594, May 2004.

[40] T. Mukherjee, W. Zhang, and T. DebRoy. An improved prediction of residual stresses and distortion in additive manufacturing. *Computational Materials Science*, 126:360–372, October 2017.

[41] A.K. Sood, R.K. Ohdar, and S.S. Mahapatra. Improving dimensional accuracy of fused deposition modelling processed part using grey Taguchi method. *Materials and Design*, 30(10):4243–4252, December 2009.

[42] G. Onwubolu and F. Rayegani. Characterization and optimization of mechanical properties of ABS parts manufactured by the fused deposition modelling process. *International Journal of Manufacturing Engineering*, 2014, November 2014.

[43] A.K. Sood, R.K. Ohdar, and S.S. Mahapatra. Experimental investigation and empirical modelling of FDM process for compressive strength improvement. *Journal of Advanced Research*, 3(1):81–90, January 2012.

[44] J.M. Chacón, M.A. Caminero, E. García-Plaza, and P.J. Núñez. Additive manufacturing of PLA structures using fused deposition modelling: Effect of process parameters on mechanical properties and their optimal selection. *Materials and Design*, 124:143–157, June 2017.

[45] M. Domingo-Espin, J.M. Puigoriol-Forcada, A.A. Garcia-Granada, J. Llumà, S. Borros, and G. Reyes. Mechanical property characterization and simulation of fused deposition modeling polycarbonate parts. *Materials and Design*, 83:670–677, October 2015.

[46] B.H. Lee, J. Abdullah, and Z.A. Khan. Optimization of rapid prototyping parameters for production of flexible abs object. *Journal of Materials Processing Technology*, 169(1):54–61, October 2005.

[47] K. Chin Ang, K. Fai Leong, C. Kai Chua, and M. Chandrasekaran. Investigation of the mechanical properties and porosity relationships in fused deposition modelling-fabricated porous structures. *Rapid Prototyping Journal*, 12(2):100–105, March 2006.

[48] A.K. Sood, R.K. Ohdar, and S.S. Mahapatra. Parametric appraisal of mechanical property of fused deposition modelling processed parts. *Materials and Design*, 31(1):287–295, January 2010.

[49] G. Percoco, F. Lavecchia, and L.M. Galantucci. Compressive properties of fdm rapid prototypes treated with a low cost chemical finishing. *Research Journal of Applied Sciences, Engineering and Technology*, 4(19):3838–3842, October 2012.

[50] F. Rayegani and G.C Onwubolu. Fused deposition modelling (FDM) process parameter prediction and optimization using group method for data handling (GMDH) and differential evolution (DE). *The International Journal of Advanced Manufacturing Technology*, 73(1-4):509–519, July 2014.

[51] S.H. Masood, K. Mau, and W.Q. Song. Tensile properties of processed FDM polycarbonate material. *Materials Science Forum*, 654-656:2556–2559, June 2010.

[52] A. Arivazhagan, S. Masood, and I. Sbarski. Dynamic mechanical analysis of fused deposition modelling processed polycarbonate. *Annual Technical Conference - ANTEC, Conference Proceedings*, 1:950–955, January 2011.

[53] A. Arivazhagan and S.H. Masood. Dynamic mechanical properties of ABS material processed by fused deposition modelling. *International Journal of Engineering Research and Applications*, 2(3):2009–2014, May 2012.

[54] A. Alafaghani, A. Qattawi, B. Alrawia, and A. Guzmana. Experimental optimization of fused deposition modelling processing parameters: A design-for-manufacturing approach. *Procedia Manufacturing*, 10:791–803, December 2017.

[55] Y. Zhang and K. Chou. A parametric study of part distortions in fused deposition modelling using three-dimensional finite element analysis. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(8):959–968, August 2008.

[56] R. Rezaie, M. Badrossamay, A. Ghaie, and H. Moosavi. Topology optimization for fused deposition modeling process. *Procedia CIRP*, 6:521–526, January 2013.

[57] D. Jankovics, H. Gohari, M. Tayefeh, and A. Barari. Developing topology optimization with additive manufacturing constraints in ANSYS. *IFAC*, 51(11):1359–1364, January 2018.

[58] H. Bikas, S. Panagiotis, and G. Chryssolouris. Additive manufacturing methods and modelling approaches: A critical review. *The International Journal of Advanced Manufacturing Technology*, 83(1-4):389–405, March 2016.

[59] N.E. Hodge, R.M. Ferencz, and J.M. Solberg. Implementation of a thermomechanical model for the simulation of selective laser melting. *Computational Mechanics*, 54(1):33–51, July 2014.

[60] A. Sengupta, P. Papadopoulos, and R.L. Taylor. A multiscale finite element method for modeling fully coupled thermomechanical problems in solids. *International Journal for Numerical Methods in Engineering*, 91(13):1386–1405, September 2012.

[61] M.E. Gurtin, E. Fried, and L. Anand. *The Mechanics and Thermodynamics of Continua*. Cambridge University Press, Cambridge, 2010.

[62] C. Trusdell and R. Toupin. *The Classical Field Theories*. Springer-Verlag, Berlin, 1960.

[63] T. Osswald and J.P. Hernández-Ortiz. *Polymer Processing Modeling and Simulation*. Hanser Publications, Munich, 2006.

[64] G. Chryssolouris. *Manufacturing Systems: Theory and Practice*. Springer-Verlag, New York, 2013.

[65] M. Nikzad, S.H. Masood, and I. Sbarski. Thermo-mechanical properties of a highly filled polymeric composites for fused deposition modeling. *Materials and Design*, 32(6):3448–3456, June 2011.

[66] S. Reese and S. Govindjee. Theoretical and numerical aspects in the thermo-viscoelastic material behaviour of rubber-like polymers. *Mechanics of Time-Dependent Materials*, 1(4):357–396, December 1997.

[67] N.M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(EM3):67–94, 1959.

[68] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, Cambridge, 1999.

[69] A. Bernard and A. Fischer. New trends in rapid product development. *CIRP Annals*, 51(2):635–652, January 2002.

[70] A. Gebhardt. *Understanding Additive Manufacturing*. Hanser Publications, Munich, 2011.

[71] M. Montero, S. Roundy, D. Odell, S.H. Ahn, and P.K. Wright. Material characterization of fused deposition modeling (FDM) ABS by designed experiments. In *Proceedings of Rapid Prototyping and Manufacturing Conference*, July 2001.

[72] A. Bellini, S.U. Guceri, and M. Bertoldi. Liquefier dynamics in fused deposition. *Journal of Manufacturing Science and Engineering*, 126(2):237–246, May 2004.

[73] R.S. Crockett. The liquid-to-solid transition in stereodeposition techniques. In *International Solid Freeform Fabrication Symposium*, 1996.

[74] N. Venkataraman, S. Rangarajan, M.J. Matthewson, B. Harper, A. Safari, S.C. Danforth, G. Wu, N. Langrana, S. Guceri, and A. Yardimci. Feedstock material property–process relationships in fused deposition of ceramics (FDC). *Rapid Prototyping Journal*, 6, December 2000.

[75] H.S. Ramanath, C.K. Chua, K.F. Leong, and K.D. Shah. Melt flow behaviour of poly-ϵ-caprolactone in fused deposition modelling. *Journal of Materials Science: Materials in Medicine*, 19(7):2541–2550, July 2008.

[76] R.B. Dupaix and M.C. Boyce. Constitutive modeling of the finite strain behavior of amorphous polymers in and above the glass transition. *Mechanics of Materials*, 39(1):39–52, January 2007.

[77] T.S. Gates and C.T. Sun. Elastic/viscoplastic constitutive model for fiber reinforced thermoplastic composites. *AIAA Journal*, 29(3):457–463, March 1991.

[78] P. Brindley, R. Goodridge, M. East, and R. Hague. Material properties: Preliminary investigation in to the mechanical properties of stratasys polycarbonate and m30abs materials. Technical report, Loughborough University, Loughborough, UK, April 2008.

[79] F.P. Incropera, A.S. Lavine, T.L. Bergman, and D.P. DeWitt. *Fundamentals of Heat and Mass Transfer*. John Wiley and Sons, New Jersey, 2007.

[80] J.C. Heigel. *Thermo-Mechanical Model Development and Experimental Validation for Directed Energy Deposition Additive Manufacturing Processe*. PhD thesis, The Pennsylvania State University, 2015.

[81] T.A. Davis. Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, 30(2):196–199, June 2004.

[82] T.A. Davis. *UMFPACK Version 5.1 User Guide*, 05 2007.

[83] S.H. Masood. Intelligent rapid prototyping with fused deposition modelling. *Rapid Prototyping Journal*, 2(1):24–33, March 1996.

[84] J.R. Groza and J.F. Shackelford. *Materials Processing Handbook*. CRC press, Boca Raton, 2007.

[85] C. Bellehumeur, L. Li, Q. Sun, and P. Gu. Modeling of bond formation between polymer filaments in the fused deposition modeling process. *Journal of Manufacturing Processes*, 6(2):170–178, January 2004.

[86] M.K. Agarwala, V.R. Jamalabad, N.A. Langrana, A. Safari, P.J. Whalen, and S.C. Danforth. Structural quality of parts processed by fused deposition. *Rapid Prototyping Journal*, 2(4):4–19, December 1996.

[87] P.M. Pandey, N.V. Reddy, and S.G. Dhande. Improvement of surface finish by staircase machining in fused deposition modeling. *Journal of Materials Processing Technology*, 132(1-3):323–331, January 2003.

[88] E. Polak. *Computational Methods in Optimization: A Unified Approach*, volume 77. Academic press, New York, 1971.

[89] J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, New York, 2006.

[90] K. Svanberg. *MMA and GCMMA-two methods for nonlinear optimization*, 2007.

Appendix A

Numerical Algorithms

Algorithm 4 selects boundary elements based on nodal level-set function ϕ. When a node is outside the region enclosed by the given curve $\phi > 0$, while $\phi < 0$ for a node inside. Therefore if an element is entirely outside or inside the curve the values of ϕ at its four vertices should maintain the same sign. When an element is at the boundary being cut by the curve, a change of sign is expected. Here, $\boldsymbol{X}$ is nodal position and $\Delta x, \Delta y$ are the width and height of the element. This algorithm is applied onto the reference configuration.

Algorithm 4: Level-set based boundary element

```
1 Given Δx and Δy, discretize the domain with linear quadrilateral elements.
2 for each potential element in the domain do
3     Find φ(x, y) based on the geometry function f(x, y)
4     if φ changes sign then
5         Mark this element as boundary element
6         Design partial element with X, φ, Δx and Δy
```

Algorithm 5 describes how a new element is added based on the position of the extrusion head and the amount of material printed during a time step. Adding a partial element requires more computations dependent on the slicing algorithms.

Algorithm 5: Adding element

```
Decide whether two new nodes or three new nodes need to be added
if adding full element then
    if adding two new nodes then
        Find two existing nodes x_{e1} and x_{e2}
        Given x_1 and x_2 for the new node, find y_1 and y_2 based on area Δx Δy s.t. its
          edge is parallel to the existing edge
    else
        Find the three existing nodes x_{e1}, x_{e1} and x_{e2}
        Calculate the triangular area a_1 based on x_{e1}, x_{e1} and x_{e2}
        Given x_1 for the new node, find y_1 based on area Δx Δy and a_1
else
    adding partial element
    if adding two new nodes then
        Find two existing nodes x_{e1} and x_{e2}
        Given x_1 and x_2 for the new node, find y_1 and y_2 based on the area of the
          partial element a_2 s.t. its edge is parallel to the existing edge
    else
        // Adding one node
        Find the three existing nodes x_{e1}, x_{e1} and x_{e2}
        Calculate the triangular area a_1 based on x_{e1}, x_{e1} and x_{e2}
        Given x_1 for the new node, find y_1 based on area of the partial element a_2
          and a_1
```

Algorithm 6 and 7 detail the slicing algorithms of one-sided and minimum-area methods in two dimension. The cases used are given in Figures 2.9 and 2.10. Here pe is the indicator of whether an element is a partial element with $pe = 0$ for a complete element, $pe = 1$ for partial element connected on the left side and $pe = 2$ for partial element connect on the right. The x_1 and x_2 are the intermediate variables for calculating the *width* of the partial element. The boolean *is_adding* determines whether an element should be added or not. The nodal level-set function values defined in Section 2.3.1 at the vertices are denoted as *siv* where $i = 0, 1, 2, 3$ are the index of the four vertices of the two-dimensional quadrilateral element. And *tol_area* is the minimum area of a partial element allowed to ensure accuracy and convergence of the solutions. The scalar ϵ is on the scale of 10^{-14} to better control numerical issues caused by scalar values hitting machine precision.

Algorithm 6: One-sided slicing algorithm

```
1  if case 1 then
2      pe = 1
3      width = s3v/(s3v − s2v)Δx
4  else if case 2 then
5      x1 = −s0v/(s1v − s0v)Δx
6      pe = 2
7      width = Δx − x1
8  else if case 3 then
9      x1 = −s3v/(s2v − s3v)Δx
10     pe = 2
11     width = Δx − x1
12 else if case 4 then
13     x1 = s0v/(s0v − s1v)Δx
14     pe = 1
15     width = x1
16 else if case 5 or 6 or 9 or 10 or 11 or 12 then
17     is_adding = 0
18 else if case 7 then
19     x1 = −s0v/(s1v − s0v)Δx
20     x2 = −s3v/(s2v − s3v)Δx
21     pe = 2
22     width = (Δx − max(x1, x2))
23     if area < tol_area then
24         is_adding = 0
25 else if case 8 then
26     x1 = s0v/(s0v − s1v)Δx
27     x2 = s3v/(s3v − s2v)Δx
28     pe = 1
29     width = min(x1, x2)
30     if area < tol_area then
31         is_adding = 0
32 area = widthΔy
```

Algorithm 7: Minimum area slicing

Set area to $\Delta x \Delta y$, width to Δx. X direction position is given.
if *case 1* **then**
    $x_1 = s3v/(s3v - s2v)\Delta x$, $y_1 = s1v/(s1v - s2v)\Delta y$.
    if $y_1 < \Delta y/2 - \epsilon$ **then**
        $pe = 1$, $width = (x_1\Delta y + \Delta x\Delta y - 2x_1y_1)/2/(\Delta y - y_1)$.
if *case 2* **then**
    $x_1 = -s0v/(s1v - s0v)\Delta x$, $y_1 = -s0v/(s3v - s0v)\Delta y$.
    if $y_1 > \Delta y/2 + \epsilon$ **then**
        $pe = 2$, $width = (\Delta y/2 - y_1)/(\Delta y - y_1)x_1$.
if *case 3* **then**
    $x_1 = -s3v/(s2v - s3v)\Delta x$, $y_1 = s0v/(s0v - s3v)\Delta y$.
    if $y1 < dy/2 - \epsilon$ **then**
        $pe = 2$, $width = (\Delta x - (\Delta y/2 - y_1)/(\Delta y - y_1) * x_1)$.
if *case 4* **then**
    $x_1 = s0v/(s0v - s1v)\Delta x$, $y_1 = -s1v/(s2v - s1v)\Delta y$.
    if $y_1 > \Delta y/2 + \epsilon$ **then**
        $pe = 1$, $width = x_1 + (\Delta x - x_1)/2/y_1\Delta y$.
if *case 5* **then**
    $y_1 = -s0v/(s3v - s0v)\Delta y$, $y_2 = -s1v/(s2v - s1v)\Delta y$
    if $y_1 + y_2 > \Delta y$ **then**
        $is\_adding = 0$.
if *case 6* **then**
    $y_1 = s0v/(s0v - s3v)\Delta y$, $y_2 = s1v/(s2v - s1v)\Delta y$.
    if $y_1 + y_2 < \Delta y$ **then**
        $is\_adding = 0$.
if *case 7* **then**
    $x_1 = -s0v/(s1v - s0v)\Delta x$, $x_2 = -s3v/(s2v - s3v)\Delta x$, $pe = 2$, $width = (\Delta x - (x_1 + x_2)/2)$.
if *case 8* **then**
    $x_1 = s0v/(s0v - s1v)\Delta x$, $x_2 = s3v/(s3v - s2v)\Delta x$, $pe = 1$, $width = (x1 + x2)/2$.
if *case 9* **then**
    $x_1 = s3v/(s3v - s2v)\Delta x$, $y_1 = -s0v/(s3v - s0v)\Delta y$.
    if $y_1 > \Delta y/2 - \epsilon$ **then**
        $is\_adding = 0$
    else
        $pe = 1$, $width = (\Delta y/2 - y_1)/(\Delta y - y_1)x_1$.
        $width = (\Delta y/2 - y_1)/(\Delta y - y_1)x_1$.
if *case 10* **then**
    $x_1 = s0v/(s0v - s1v)\Delta x$, $y_1 = s0v/(s0v - s3v)\Delta y$.
    if $y_1 < \Delta y/2 + \epsilon$ **then**
        $is\_adding = 0$.
    else
        $pe = 1$, $width = (y_1 - \Delta y/2)/y_1x_1$.
if *case 11* **then**
    $x_1 = -s0v/(s1v - s0v)\Delta x$, $y_1 = s1v/(s1v - s2v)\Delta y$.
    if $y_1 < \Delta y/2 + \epsilon$ **then**
        $is\_adding = 0$.
    else
        $pe = 2$, $width = (\Delta x - \Delta y(\Delta x - x_1)/2/y_1 - x_1)$.
if *case 12* **then**
    $x_1 = -s3v/(s2v - s3v)\Delta x$, $y_1 = -s1v/(s2v - s1v)\Delta y$.
    if $y_1 > \Delta y/2 - \epsilon$ **then**
        $is\_adding = 0$.
    else
        $pe = 2$, $width = ((\Delta y/2 - y_1)/(\Delta y - y_1) * (x_1 - \Delta x) + \Delta x)$.
if $area < tol\_area$ **then**
    $is\_adding = 0$
$area = width\Delta y$