A SCALABLE FRAMEWORK FOR CONTACT-AWARE THERMAL SIMULATION
OF ADDITIVE MANUFACTURING PROCESSES

by

Yaqi Zhang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Mechanical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2020

Date of final oral examination: 05/07/20

The dissertation is approved by the following members of the Final Oral Committee:
Vadim Shapiro, Professor, Mechanical Engineering
Tom Krupenkin, Associate Professor, Mechanical Engineering
Wenxiao Pan, Assistant Professor, Mechanical Engineering
Xiaopin Qian, Professor, Mechanical Engineering
Krishnan Suresh, Professor, Mechanical Engineering
Dan Negrut, Professor, Computer Science
ACKNOWLEDGMENTS

It is a great pleasure to spend the past six years at the University of Wisconsin-Madison. I wish to thank my advisor, Professor Vadim Shapiro, for his patience, insight, advice, rigorous attitude to research and years of support and guidance. Working with him not only has been a rewarding and memorable experience but also teaches me what it takes to succeed as a researcher. Without his direction, this thesis would never have been written. I also want to thank Professor Xiaoping Qian, for his support during my first year at the University of Wisconsin-Madison.

I feel fortunate to have Professors Krishnan Suresh, Xiaoping Qian, Wenxiao Pan, Tom Krupenkin, and Dan Negrut on my dissertation committee. Their insightful suggestions, constructive criticisms, and general guidance are vital in the completion of this work. Working at the Spatial Automation Laboratory (SAL) has given me the opportunity to interact with my colleagues Xingchen Liu, Vaidyanathan Thiagarajan, Goldy Kumar, Brian McCarthy, Randi Wang, Parag Pathak, and Xin Liu. I appreciate their friendship and many fruitful discussions. I wish them the very best in their endeavors.

Finally, I would like to thank my family for their love, support and the financial assistance they have given me over the past 30 years. I want to thank my wife Shu and my son Liaoran for their accompany and support. I wish to thank my parents-in-law for taking care of my son. I especially thank my parents for valuing education and supporting me to pursue happiness.
# CONTENTS

<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Proposed approach</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Outline</td>
<td>5</td>
</tr>
<tr>
<td>2 Related Work</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Fused deposition modeling</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Powder bed fusion</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Discussion</td>
<td>16</td>
</tr>
<tr>
<td>3 General Framework</td>
<td>18</td>
</tr>
<tr>
<td>3.1 Contact-aware path-level (CAPL) discretization</td>
<td>18</td>
</tr>
<tr>
<td>3.1.1 Path level discretization</td>
<td>19</td>
</tr>
<tr>
<td>3.1.2 Contact graph</td>
<td>22</td>
</tr>
<tr>
<td>3.2 Lumped-capacitance model</td>
<td>25</td>
</tr>
<tr>
<td>3.3 Scalability and active body</td>
<td>28</td>
</tr>
<tr>
<td>3.4 Simulation</td>
<td>30</td>
</tr>
<tr>
<td>3.5 Discussion</td>
<td>33</td>
</tr>
<tr>
<td>4 Fused Deposition Modeling</td>
<td>35</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>35</td>
</tr>
<tr>
<td>4.2 Formulation</td>
<td>37</td>
</tr>
<tr>
<td>4.2.1 CAPL discretization</td>
<td>37</td>
</tr>
</tbody>
</table>
4.2.1.1 Tool path discretization ...................................... 37
4.2.1.2 Contact graph .................................................. 40
4.2.2 Lumped-capacitance model ....................................... 43
4.2.3 Active body .......................................................... 49
4.2.4 Simulation ............................................................ 52
4.3 Numerical experiments ............................................... 56
  4.3.1 Simple accuracy test .............................................. 56
  4.3.2 Influence of active body ........................................ 59
  4.3.3 Realistic numerical simulation ................................. 63
4.4 Discussion .............................................................. 68

5 Powder Bed Fusion ..................................................... 69
  5.1 Introduction .......................................................... 69
  5.2 Formulation ........................................................... 71
    5.2.1 CAPL discretization ........................................... 72
      5.2.1.1 Scan path discretization ................................. 72
      5.2.1.2 Contact graph .............................................. 74
    5.2.2 Lumped-capacitance model ................................... 76
    5.2.3 Material properties ........................................... 86
    5.2.4 Element growth ................................................ 90
    5.2.5 Active body ..................................................... 92
    5.2.6 Simulation ....................................................... 94
  5.3 Numerical experiments ............................................. 95
    5.3.1 Single path ..................................................... 96
    5.3.2 Influence of active body ..................................... 98
    5.3.3 Further validation ............................................ 104
    5.3.4 Realistic numerical simulation ............................. 108
  5.4 Discussion ........................................................... 110

6 Data-Driven Model - Melt Pool Prediction and Control .............. 111
  6.1 Introduction ........................................................ 111
6.2 Data driven model ........................................ 113
  6.2.1 Experimental Data .................................. 113
  6.2.2 Data exploration .................................... 117
  6.2.3 Feature engineering and neural network .......... 117
  6.2.4 Train and test ..................................... 121
  6.2.5 Ensemble methods .................................. 122
  6.2.6 Model interpretation ................................. 124
6.3 Application ............................................ 128
  6.3.1 Laser power design ................................. 128
  6.3.2 Hybrid model ...................................... 132
6.4 Discussion ............................................. 133

7 Conclusions and Open Issues .............................. 135

References .................................................. 137
LIST OF TABLES

4.1 Properties of ABS P400 ..................................................... 56

5.1 Material properties for 316L stainless steel as a function of temperature [25] ................................................................. 87

5.2 Comparison of melt pool width results expanded from [57] ............ 98
# LIST OF FIGURES

1.1 Diagram of the general framework. ........................................ 4

2.1 Schematic of deposition of a road in FDM process. ......................... 9

2.2 Schematic of deposition and cooling of a filament in the FDM process. 11

2.3 Schematic of powder bed fusion [29]. ................................... 14

3.1 An example of G-code file used in FDM process. ........................ 20

3.2 Discretization example of one layer in a octopus part. ................... 21

3.3 Illustration of binning algorithm. .......................................... 24

3.4 Example of contact graph. .................................................. 24

4.1 Schematic of a typical fused deposition modeling set up [1]. ............ 36

4.2 G-code file (left) and tool path (right) of a single layer part. .......... 38

4.3 Discretization of a single layer part in Figure 4.2. ....................... 39

4.4 Diagram of a single road. .................................................. 41

4.5 Microphotograph of the cross-sectional area of a FDM part, where $W$ is the road’s width, $H$ is the layer height, and $2y$ is the neck length between adjacent roads within the same layer [77]. ............................. 42

4.6 Cross-sectional shape of road. ............................................. 43

4.7 Top view of contact between two elements in the same layer: adjacent parallel roads. .................................................. 44

4.8 Top view of contact between two elements in the same layer: end-road contact ($L' = W$). .................................................. 44

4.9 Top view of contact between two elements in vertically adjacent layers. 45

4.10 Illustration of different thermal effects on element boundary. ........ 46

4.11 Active body $\mathbb{A}^t = \mathbb{A}_{\text{temp}}^t \cup \mathbb{A}_{\text{spat}}^t$. .................. 50

4.12 Illustration of interacting elements in a given time-step of the proposed numerical scheme. .................................................. 54

4.13 Schematic of deposition of a road in FDM process ....................... 57
4.14 Comparison of simulation ($\Delta t = 0.1$s results with the analytic solution from [6]) ................................................................. 58
4.15 Tool path model of the octopus part. .................................................. 59
4.16 Influence of active body on simulation time of FDM process. .............. 60
4.17 Influence of active body on simulation time of FDM process (loglog scale). 61
4.18 Influence of active body on computation results of octopus part. ......... 63
4.19 Influence of active body on computation results of the “mobius arm” part. 64
4.20 3D model of a “mobius arm” part (Courtesy of Stratasys, Ltd.) ........... 65
4.21 Influence of active body on computation results of an aerospace rocker part. ................................................................. 65
4.22 3D model of an aerospace rocker part (Courtesy of Stratasys, Ltd.) ... 66
4.23 Tool path of the “mobius arm” part. .................................................... 66
4.24 Snapshots of thermal simulation for the manufacturing of mobius arm part, color indicates the temperature. ................................. 67

5.1 Schematic of a typical powder bed fusion set up [2]. ............................. 69
5.2 Scan path of a single layer part. ............................................................. 73
5.3 Discretization of part in Figure 5.2. ....................................................... 74
5.4 Illustration of binning algorithm used in PBF simulation. ...................... 76
5.5 Illustration of contacts between adjacent elements. ............................. 77
5.6 Gaussian intensity profile. ................................................................. 82
5.7 Error function. ............................................................. 84
5.8 Illustration of coordinate transformation when computing H. ............... 86
5.9 Conductivity of 316L stainless steel. .................................................. 88
5.10 Density of 316L stainless steel. ......................................................... 89
5.11 Specific heat capacity of 316L stainless steel. .................................... 89
5.12 Active body example of PBF simulation. .......................................... 94
5.13 Experimental micrograph conducted by Khairallah and Anderson [36]. 97
5.14 Temperature and width evolution of an element. ................................ 99
5.15 Tool path of parts with different size. ............................................... 100
5.16 Influence of active body on simulation time of PBF process. .............. 101
5.17 Influence of active body on simulation time of PBF process (loglog scale). 102
5.18 Element position. ................................................................. 102
5.19 Influence of active body on accuracy ...................................... 103
5.20 Snapshots of thermal simulation of a part with four layers. ....... 103
5.21 Schematic of AMMT system. ................................................. 105
5.22 Raw melt pool images. ............................................................ 106
5.23 Experimental melt pool field. ............................................... 106
5.24 Simulation melt pool field. .................................................... 107
5.25 3D model of Stanford Bunny [83]. ....................................... 108
5.26 Scan path model of the Stanford Bunny part. ......................... 109
5.27 Snapshots of thermal simulation of Stanford bunny. .............. 109

6.1 laser scan path. ................................................................. 114
6.2 Raw melt pool image (128 pixel × 120 pixel). ....................... 115
6.3 The pipeline of preprocessing melt pool images. ...................... 115
6.4 Experimental data. .............................................................. 116
6.5 Melt pool area distribution. ................................................... 116
6.6 Pairplot of laser-power/scan-speed/energy-density and melt pool length-width/area. .................................................... 117
6.7 The paradigm of supervised learning. .................................... 118
6.8 Structure of the neural network. .......................................... 121
6.9 Comparison of melt pool prediction and experimental data. ...... 123
6.10 Performance of bagging. ..................................................... 124
6.11 Performance of stacking. .................................................... 125
6.12 Influence of laser power on melt pool length/width. ............... 126
6.13 Influence of scan speed on melt pool length/width. ................ 127
6.14 Designed laser power field. ................................................. 130
6.15 Resultant melt pools of the designed laser power plan. ........... 131
6.16 Resultant Melt pool field of constant laser power and scan speed. 132
A SCALABLE FRAMEWORK FOR CONTACT-AWARE THERMAL SIMULATION OF ADDITIVE MANUFACTURING PROCESSES

Yaqi Zhang
Under the supervision of Professor Vadim Shapiro
At the University of Wisconsin-Madison

Many additive manufacturing (AM) processes are driven by a moving heat source. For these processes, thermal field evolution during the manufacturing process plays an important role in determining both geometric and mechanical properties of the fabricated parts. Thermal simulation of AM processes is computationally challenging due to the geometric complexity of the manufacturing process and inherent computational complexity that requires a numerical solution at every time increment of the process.

We propose a new general computational framework that supports scalable contact-aware thermal simulation of any AM process driven by a moving heat source. The proposed framework has three novel ingredients. First, the “process-aware” path-level discretization is based on tool path, and the thermal model is formulated directly in terms of manufacturing primitives. Second, a spatial data structure, called contact graph, is used to represent the discretized domain and capture all possible thermal interactions during the simulation. Finally, the simulation is localized based on specific physical parameters of the manufacturing process, requiring at most a constant number of temperature updates at each time step. The latter implies that the constructed simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance. The “process-aware” path-level discretization and contact graph representation form the basis of a novel discretization called contact-aware path-level (CAPL) discretization, which can capture all possible contact conditions in the as-manufactured geometry. To demonstrate the efficacy and generality of the framework, it has been successfully applied to build thermal simulations of two different AM processes, fused deposition modeling (FDM) process and powder bed fusion (PBF) process. Finally, CAPL discretization and
the idea of locality are further utilized to incorporate the neighboring effect of
the scan path into a data-driven model, which can predict melt pool size based
on the process plan. We demonstrated how the trained surrogate model can be
used as a forward solver for developing novel laser power design algorithms. The
trained surrogate model could also be integrated into our thermal simulation of
PBF process to guide element width initialization.
1 INTRODUCTION

1.1 Motivation

Additive manufacturing (AM) has become an important technology for producing a wide variety of structures and components [61]. The AM processes build a three-dimensional (3D) part from a computer-aided design (CAD) model, usually by successively adding material layer by layer. Many AM processes (particularly those used in production environments) are driven by a moving heat source, which creates new solid material through some phase-changing process. Whether it is a polymer AM process, such as fused deposition modeling (FDM) [98], or a metal AM process, such as powder bed fusion (PBF) [7], microstructure, distortion and mechanical properties of the final printed parts depend heavily on the temperature field evolution during the manufacturing processes. Temperature field evolution plays an important role in determining both the geometric and mechanical properties of the fabricated components. In order to improve quality control and optimize manufacturing parameters, it is critical to have a quantitative understanding of the relationship between process inputs (e.g., manufacturing parameters and tool path) and temperature field evolution developed during the manufacturing process. Computational models and simulations are useful in quantifying and understanding heat transfer during the manufacturing process. Such models and simulations are usually built for a specific AM process on a case by case basis, often using existing or specialized finite elements codes.

Thermal simulation of AM processes is challenging at least for two reasons. The first one is the inherent geometric complexity of the manufacturing process that discretizes the design model and leads to numerous artifacts, such as voids between roads in FDM process [103]. Additively manufactured shapes are constructed by fusing together individually printed primitives (scans and/or layers). Any accurate representation of the solution domain must correctly capture all possible contact conditions, including numerous non-manifold regions in the interior of the domain. Most traditional discretization techniques (e.g., conforming and non-conforming
meshes, discrete elements, uniform Cartesian meshes, etc.) are not suitable for this purpose. For example, common continuum-based thermal methods rely on voxelization of the design geometry, which ignore the obvious fact that design geometry differs significantly from the manufactured shape [49]. So a good discretization of the as-manufactured shape-material model should be “process-aware”, designed in a manner that reflects the nature of geometric and material models implied by the manufacturing process. The second challenge is the high computational complexity, stemming from the fact that the transient thermal simulation requires a numerical solution at every time increment of the process. Traditional numerical methods require temperature updates of all the elements at every time step, the complete thermal simulation of an AM process with \( n \) elements and \( m \) time steps requires \( O(mw) \) computational cost, where \( w \) is the total cost in any one time step and could be as high as \( O(n^3) \). This dramatically limits the size of feasible simulations both in time and in space. Such simulations are clearly not scalable. To be scalable, a simulation should grow linearly with increased complexity in geometry (space) and time (number of steps).

In contrast, we consider the thermal simulation of AM processes driven by a moving heat source from a higher level of abstraction and develop a general framework to build scalable thermal simulation based on the as-manufactured geometry for any such process.

## 1.2 Proposed approach

We propose a novel general framework that acts as a template to generate a scalable thermal simulation of any AM process driven by a moving heat source. As illustrated in Figure 1.1, the framework generates a scalable thermal simulator for

\[1\text{ Solving a system of dense linear equations has a complexity of at most } O(n^3). \text{ The best algorithm known to date was developed by Don Coppersmith and Shmuel Winograd [27] which has a complexity of } O(n^{2.376}). \text{ However, it is of little practical use. It should be noted that the linear systems generated by traditional numerical methods (e.g., finite element methods) are usually sparse, which are cheaper to solve. For example, Gaussian elimination for a matrix of bandwidth } b \text{ takes } O(nb^2) \text{ operations.}

\[2\text{ It would be more accurate to say “as planned to manufacture.”} \]
any AM process driven by a moving heat source as input, through instantiating of four generic components: path-level discretization algorithm, contact graph data structure, lumped-capacitance heat-transfer model, and active body localization algorithm. The simulation generated from the framework distinguishes itself from the prior state of art by three novel ingredients.

1. The “process-aware” path-level discretization is based on the manufacturing primitives described by the tool path plan and the thermal model is formulated directly in terms of manufacturing primitives;

2. A spatial data structure, called contact graph, is used to represent the discretized domain and capture all possible thermal interaction during the simulation;

3. The simulation is localized based on specific physical parameters of the manufacturing process, requiring at most a constant number \(O(1)\) of temperature updates at each time step.

The “process-aware” path-level discretization and contact graph representation form the basis of a novel discretization called contact-aware path-level (CAPL) discretization. Since the notion of the element also depends on the process, different processes (may) have different CAPL discretizations. Our approach does not restrict itself to simulations on the design geometry of a part, but instead simulates the formation of the geometry based on the process plan of a part. In addition, localization of computation ensures that the number of updates at each time step does not depend on the total number of elements, which implies that the constructed simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance.

Also, it should be noted that discretization based on path level does not imply the simulation is “low fidelity”. It is as good (or maybe even better) than any other simulation at the same scale, for instance, at the scale where part geometry is represented in terms of manufacturing primitives and material properties are assumed to be known for each element. We call this mesoscale. The so-called “high
“High fidelity” models simulate the process at finer (micro) scale, e.g., powder particle level, which deals with phase transitions, and predicts homogenized properties. As such, high fidelity models are intended to predict local behaviors and are not intended for simulation of the whole part/process. In contrast, our path-level simulation is designed to simulate realistic process plans.

1.3 Contributions

This thesis introduces a scalable framework for the thermal simulation of additive manufacturing processes, which are driven by a moving heat source. The thesis
has four major contributions:

1. **Simulation based on manufacturing primitives**, in which the discretization is based on the manufacturing primitives described by the tool path plan.

2. **Contact graph representation**, in which a data structure called contact graph is used to represent the discretized domain and capture all possible thermal interaction during the simulation.

3. **Scalability (in both time and space)** is achieved through exploiting temporal and spatial locality with active body, which allows updating at most constant number of elements in each time step.

4. The proposed framework has been successfully applied to build scalable thermal simulations of two different AM processes: fused deposition modeling (FDM) and powder bed fusion (PBF).

### 1.4 Outline

The remainder of this thesis is organized as follows\textsuperscript{3}. Chapter 2 describes the current research space - thermal simulation of AM processes. A variety of thermal models have been proposed based on different assumptions and purposes, with physics fidelity ranging from low to high. Broadly speaking, most simulations fall into one of the three categories of models, in the order of increasing fidelity: (semi-)analytical models that rely on closed-form expressions, continuum-based methods, which are usually approximated by finite elements, and micro-scale methods, which attempt to capture the phase-changes of the manufacturing process. We focus on thermal simulations of the two most common AM processes: fused deposition modeling (FDM) process, which is the most widely used polymer AM process and powder bed fusion (PBF) process, which is the dominant metal AM process. The related work on thermal simulations of the FDM process is summarized in Chapter

\textsuperscript{3}This dissertation is written in a way that each chapter is self-contained.
2.1. The related work on thermal simulations of the PBF process is described in Chapter 2.2.

Chapter 3 introduces the general framework. In it, the CAPL discretization, which includes path-level discretization algorithm and contact graph data structure, is introduced in Chapter 3.1. Lumped-capacitance heat transfer model is introduced in Chapter 3.2. Chapter 3.3 discusses the scalability issue and active body localization algorithm. In order to demonstrate the efficacy and generality of the framework, we applied the proposed framework to build thermal simulations of two different AM processes (FDM and PBF) in the following two chapters.

The general framework is applied to build a linear time thermal simulation of the FDM process in Chapter 4. In order to construct CAPL discretization of the FDM process (Chapter 4.2.1), the manufacturing primitive - deposited road, is discretized into elements according to the tool path and a contact graph is constructed to represent adjacency information between elements. A lumped-capacitance heat transfer model (Chapter 4.2.2), which accounts for most relevant thermal effects including heat convection and radiation to the ambient environment, heat conduction along the road, heat conduction to build platform, heat conduction between contacting elements, is built to conduct a transient thermal simulation. The thermal simulation achieves linear time complexity by exploiting spatial and temporal locality with active body (Chapter 4.2.3), which is based on known empirical data. This allows updating at most a constant number of elements at each time step. With the help of active body, the simulation achieves linear time complexity. The simulation is fully implemented, validated against known analytic solutions, and is tested on realistic complex shapes in Chapter 4.3.

In Chapter 5, the general framework is applied to build a scalable thermal simulation of the PBF process. Chapter 5.2.1 describes how to construct CAPL discretization of PBF process. The scan path discretization algorithm is introduced in Chapter 5.2.1.1, in which the scan path is discretized into many sub-paths, and each element is defined as newly melt/sintered material by laser scan along the corresponding sub-path. Contact graph (Chapter 5.2.1.2), which captures connectivity between adjacent elements, is initialized in pre-processing stage and updated
during the simulation. A lumped-capacitance heat transfer model (Chapter 5.2.2), which accounts for most relevant thermal effects including laser heating, heat convection and radiation to the ambient environment, heat conduction along the path, heat conduction to build platform, heat conduction to powder bed and heat conduction to contacting elements, is built to conduct a transient thermal simulation. An element growth mechanism is introduced in Chapter 5.2.4. The generation and growth of elements during the simulation capture the laser scan and powder melting phenomenon during the manufacturing process. The simulation is localized by active body (Chapter 5.2.5) which allows updating at most constant number of elements at any time step. This implies that the proposed mesoscale simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance. The full-fledged simulation is implemented and validated against experimental data and other simulation results in Chapter 5.3.

In Chapter 6, the CAPL discretization and the idea of temporal and spatial locality are applied to build a neighborhood-based neural network for melt pool prediction and control of the PBF process. The detailed process of building the data-driven model is described in Chapter 6.2. The experiment set up is introduced in Chapter 6.2.1. Chapter 6.2.2 explore the experimental data. Feature engineering and the structure of the model is discussed in Chapter 6.2.3. A neural network is trained and tested using experimental data in Chapter 6.2.4. Different ensemble methods (including bagging, boosting, and stacking) are implemented to improve the performance of the trained surrogate model in Chapter 6.2.5. The surrogate model is interpreted in Chapter 6.2.6. Two applications of the surrogate model are introduced Chapter 6.3. A laser power design algorithm, which uses the trained surrogate solver as a forward model and can keep melt pool size as constant as possible for any given scan path, is proposed in Chapter 6.3.1. It could also be integrated into our mesoscale thermal simulation of PBF process to guide the element width initialization (Chapter 6.3.2).

Finally, we summarize the contributions of this thesis and discuss extensions and open issues in Chapter 7.
In this chapter, we review the thermal simulation of AM processes driven by a moving heat source. Thermal field evolution during the manufacturing process has a significant influence on both geometric and mechanical properties of printed components. As far as the author knows, no general framework for thermal simulation of AM processes has been proposed before. A variety of thermal models have been proposed based on different assumptions and purposes, with physics fidelity ranging from low to high. However such models and simulations are usually built for a specific AM process on a case by case basis. Broadly speaking, most simulations fall into one of the three categories of models, in the order of increasing fidelity: (semi-)analytical models relying on closed-form expressions, continuum-based methods, which are usually approximated by finite elements, and micro-scale methods, which attempt to capture the phase-changes of the manufacturing process. Required simulation time also varies widely, ranging from few seconds for low fidelity models to hundreds or thousands of CPU hours for high fidelity simulations. Here we focus on simulation approaches of the two most common AM processes: fused deposition modeling (FDM) and powder bed fusion (PBF).

2.1 Fused deposition modeling

Fused deposition modeling (FDM) process is driven by a moving heat source. A schematic of the FDM extrusion process is shown in Figure 2.1. As the printer head moves during the manufacturing process, a semi-molten filament is extruded through a heated nozzle. For each layer, the nozzle moves following a piecewise linear path horizontally. The material deposited along each line segment is commonly referred as a road.

The formation of bonds between adjacent roads is induced by the thermal energy of semi-molten material [76], and the repeated heating and cooling of the materials
can aggregate nonuniform thermal gradient and cause stress accumulated that consequently result in part distortion [104]. Thermal analysis not only is important to understand (and predict) different manufacturing failures but also provides the basis for the design of the manufacturing process plan. Generally speaking, the previous research on thermal simulation of FDM can be divided into two categories, in the order of increasing fidelity: 1) analytical models which simplify the problem such that it can be solved analytically; 2) continuum-based models which rely on voxelization of designed geometry and usually approximated by finite element methods. The two categories are discussed below.

Several analytical models have been developed to predict the temperature history of a single road [60]. Thomas and Rodriguez [81] presented a simplified two-dimensional thermal model that treated the roads as rectangle in shape. The analytic solution for the temperature averaged over the width of the road is obtained. The resulting equation for the road temperature averaged over the width of the road for filament width ($W$) and height ($H$) is
\[
T_{\text{ave}}(x, y, t) = T_E \left[ 1 + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} (a_{mn} \sin(\lambda_m y) \cos(\beta_n x)) \times \exp \left( -\left( \frac{k}{C \rho} \right)^2 (\lambda_m^2 + \beta_n^2) t \right) \right],
\]

(2.1)

where

\[
a_{mn} = \frac{4T^*_E}{E_m^2 F_n^2 \lambda_m \beta_n} \sin\left( \frac{9\lambda_m H}{2} \right) \sin\left( \frac{\lambda_m H}{2} \right) \sin\left( \frac{\beta_n W}{2} \right)
\]

(2.2)

\[
E_m^2 = \frac{1}{2} \left( 5H - \frac{\sin(10\lambda_m H)}{2\lambda_m} \right)
\]

(2.3)

\[
F_n^2 = \frac{1}{2} \left( \omega - \frac{\sin(\lambda_n \beta_n W)}{\beta_n} \right)
\]

(2.4)

In the above equations, \( t, C, k \) and \( \rho \) are time, heat capacity, thermal conductivity and density, respectively. And the eigenvalues are the roots of the equations [81]:

\[
\lambda_m \cot(5\lambda_m H) = -\frac{h}{k}
\]

(2.5)

and

\[
\beta_n \tan\left( \frac{\beta_n W}{2} \right) = \frac{h}{k}
\]

(2.6)

This model neglected the effects of conduction to the build platform and any contact resistance between filaments, the former of which is the predominant heat transfer mechanism in the system [6]. Observation that the two-dimensional
analysis showed temperature gradients that rapidly become negligible along the width and height of the filament led Bellehumeur et al to proposed a lumped capacity model which assumed uniform temperature distribution of the cross-sectional area of filament, semi-infinite filament length, and constant heat transfer and convection coefficients [6, 77]. The cooling process is thus simplified into a one-dimensional (1D) transient heat transfer model (Figure 2.2).

A single deposition road is modeled as a simple block (1D sweep with ellipse cross section). The governing equation is simplified to a 1D transient heat transfer equation,

$$\rho c A \frac{\partial T}{\partial t} = A \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) - h P (T - T_\infty),$$  \hspace{1cm} (2.7)

with

$$T = \begin{cases} T_0 & x = 0 \text{ and } t \geq 0 \\ T_\infty & x = \infty \text{ and } t \geq 0, \end{cases}$$  \hspace{1cm} (2.8)
where $T_0$ is the deposition temperature and $T_\infty$ is the environment temperature. The origin is moving at a velocity $v \,(x = vt)$, and the time dependence term $\frac{\partial T}{\partial t}$ is transformed to

$$\frac{\partial T}{\partial t} = \frac{\partial T}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial T}{\partial x} v$$ (2.9)

Then the governing equation is reduced to an ordinary differential equation

$$\rho c A v \frac{\partial T}{\partial x} = A \frac{\partial (k \frac{\partial T}{\partial x})}{\partial x} - hP(T - T_\infty), \quad (2.10)$$

The above equation could be solved analytically with the boundary conditions defined in Equation 2.8.

$$T = T_\infty + (T_0 - T_\infty) e^{-mx}, \quad (2.11)$$

where $m = \frac{\sqrt{1 + 4 \alpha \beta} - 1}{2 \alpha}$ and $x = vt$ with $\alpha = \frac{k}{\rho c v}$ and $\beta = \frac{hP}{\rho c A v}$. This approach has the advantage of simplicity but does not extend to simulations of realistic process plans with complex tool paths. Costa et al. [12, 14] proposed an analytical solution for the transient heat transfer during filament deposition, taking into account contacts between filaments and assuming a relatively simple deposition sequence. Both radial and axial heat conduction are ignored in their model. In another paper [13], the same authors examined the contribution of various thermal phenomena during the manufacturing process to overall heat transfer. Even though analytical models are usually easy to implement and efficient to run, their weakness is that simplified closed-form solutions are limited to simple cases (e.g., cooling condition of single road or simulation of a simple deposition strategy) and cannot be applied to simulate realistic process plans with complicated geometries.

Approaches based on voxelization of the design geometry rely on voxel-based numerical methods (e.g., finite element method) to solve a transient heat transfer
problem in each time-step. Zhang and Chou [102] developed a finite element analysis model using “element activations” to simulate coupled mechanical and thermal phenomena in FDM as well as resulting stresses and distortions. In their model, all the elements are deactivated before the simulation begins. During the simulation, elements are activated according to the process plan. Ji and Zhou [33] proposed a three-dimensional transient thermal finite element model taking into account temperature-dependent thermal conduction and heat capacity. Yardimci and coworkers [4, 92] proposed a 1D thermal model in which each road is modeled as a 1D array of blocks. The thermal interactions with the environment and between roads are considered by including sink terms in the enthalpy form energy equation. Finite volume method was utilized for spatial and temporal discretization. More recently, a number of researchers adopted voxelization of the design geometry as a basis for simulating moving heat source problems common in additive manufacturing [100, 41, 69, 105]. As we explained earlier, it is difficult for such approaches to account for significant difference between design and as-manufactured geometries and tend to be computationally prohibitive, which implies that these approaches can not scale to handle 3D printed components of arbitrary complexity.

### 2.2 Powder bed fusion

Powder bed fusion (PBF) process is often used in producing metal parts which includes the following commonly used printing techniques: direct metal laser sintering (DMLS), electron beam melting (EBM), selective heat sintering (SHS), selective laser melting (SLM) and selective laser sintering (SLS). PBF process is also driven by a moving heat source. A schematic of the process is shown in Figure 2.3. The laser scans on the top of the powder bed according to a prescribed scan path. The heat transfer process involves radiation and convection from power bed to the ambient environment, heat conduction in the powder bed and between the powder bed and substrate, laser energy absorption, and latent heat release/absorption due to phase change. The complexity brought by the variation of thermal properties and powder phase change also complicated the heat transfer problem. Since thermal
field evolution has a significant impact on geometric and mechanical properties of the final printed parts, many researchers have made their effort to understand the physical mechanism of PBF [82, 84, 72] and a variety of thermal models of PBF process have been developed based on different assumptions and purposes, however, the fidelity of these models ranges from low to high. Next, we discuss different approaches in the order of increasing fidelity.

Semi-analytical thermal models of PBF process assume a thermal homogeneous semi-infinite domain [70, 18, 51, 74, 73]. Due to the simplified assumptions, these models only estimate the steady-state temperature field in single track cases. The calculation of these models can be done in seconds. Continuum-based models are usually discretized on the voxelization of the design geometry. The most common formulation considers thermal evolution as a heat transfer process utilizing the Fourier heat conduction theory. Carslaw and Jaeger [11] used the below governing equation to describe heat conduction in the moving medium.

$$\lambda \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) + q = \rho c \frac{\partial T}{\partial t}$$ (2.12)

Figure 2.3: Schematic of powder bed fusion [29].
The initial condition is set as $T(x, y, z, t = 0) = T_0$, boundary condition at free surface consists of radiation and convection, which is expressed as

$$-\lambda \frac{\partial T}{\partial z} = \epsilon \sigma_{SB}(T^4 - T_\infty^4) + h(T - T_\infty),$$

where $T$ is temperature, $\lambda$ is the conductivity coefficient, $q$ is the heat source term, $\rho$ is the density, $c$ is the heat capacity, $T_0$ denotes the initial temperature of the powder bed, $T_\infty$ is the environment temperature, $h$ is the convection heat transfer coefficient, $\sigma_{SB}$ is the Stefan-Boltzmann constant and $\epsilon$ is the emissivity coefficient. Heat loss at the bottom is ignored $-\lambda \frac{\partial T}{\partial z}|_{z=0} = 0$.

Dai et al. [16, 17] used the above governing equation to study the thermal field of dental porcelain using SLM. Matsumoto et al. [54] applied finite element methods to study the distribution of temperature and stress of single layer forming of SLS. Dong et al. [20] and Kolossov et al. [40] created three-dimensional finite element (FE) models for the temperature evolution during laser sintering. The later model also predicts densification. Hodge et al. [31] applied a 3D coupled thermal-mechanical FE model to conduct part-scale simulations and investigate optimal conditions for the printing of overhanging sections. Antony et al. [3] used FE and experimental analysis to study the SLM process of 316L stainless steel powders. An investigation was made into the effects of wetting angle and balling, where the melt pool tends to ball up rather than forming one continuous track if the length to width ratio of the track is too high.

Gusarov et al. [30, 29, 28, 82] have conducted some work on modeling radiation transfer in powder bed fusion. Members of the Zaeh research group at TU Munich have multiple publications regarding part level modeling of both SLM and electron beam melting, including modeling of complete parts, although the actual size of the parts is difficult to determine [45, 95, 96, 44, 43, 94, 42, 97, 22]. Stucker et al., at the University of Louisville, have designed various component technologies intended to speed up the simulation of SLM manufactured parts, with the promise of sub real-time simulations, but experience with this for real-life part geometries
has not yet been published [63, 62, 64, 99]. Other examples using finite element methods are in [41, 88, 86, 87, 20, 40]. In principle, most of the commercial FEM codes can be used to perform such simulations, but the computation times tend to be prohibitively high. In particular, it takes tens to hundreds of CPU hours to simulate the thermal history of a manufacturing process for a small component.

Finer scale models (i.e., micro-scale) often refer to high-fidelity models at the powder scale. Lawrence Livermore National Laboratory [38, 35] proposed a multi-scale modeling strategy of PBF which includes a model at the scale of powder that simulates single track builds and provides powder bed and melt pool thermal data and a second model computationally builds a complete part and predicts manufactured properties. Their in-house code ALE3D is used for micro-scale simulation. The Zohdi group at the University of California at Berkeley has applied the discrete element method (DEM) to study selective laser sintering process [107, 109, 106, 108, 24]. In their simulation, the material was treated as discrete particles instead of a continuum medium. The thermal transfer is modeled through conduction between particles as well as thermal interactions between the laser beam and particles. The Michopoulos group at Naval Research Laboratory has applied a multi-physics discrete element method (MDEM) to simulate metal AM processes [75, 56]. In their simulation, a technique of deactivating previous layers are used to improve the computational efficiency. Korner et al [5, 71] applied their in-house code based on the Lattice Boltzmann method to simulate melt pool dynamics during selective electron beam melting. The authors used 2D simulation to decrease simulation time. The computation time of these (powder scale) models could be extremely high (thousands of CPU hours and more), thus the simulation domain is usually limited to a small space scale (several millimeters in each dimension) and a short time scale (several milliseconds).

2.3 Discussion

In summary, none of the above approaches to thermal simulation of AM processes are for realistically complex components. (Semi-)analytical models are limited by
their simplified assumptions, which are not applicable. Continuum-based models based on voxelization of design geometry can not account for significant differences between as-designed and as-manufactured geometry and tend to be computationally prohibitive. High fidelity powder-based models can capture the most physical factors at micro-scale, but only locally due to extremely high computational cost. Also, most of the classical discretization methods are not suitable for simulating AM shapes, which are constructed by fusing together individually printed primitives. For example, a voxel-based finite element model may be able to approximate the geometry of primitives (e.g., roads in FDM) with extremely high resolution and very high computational cost. However, voxelization cannot capture the correct contact between the primitives and will not preserve the model's topology under refinement. This problem could be overcome, at least in principle, by a classical conforming finite element mesh, but this would require an accurate generation of the as-manufactured geometry and topology and a suitable meshing algorithm - both are difficult unsolved problems.

In contrast, we propose a general framework to the thermal simulation of AM processes (driven by a moving heat source) that accurately models as-manufactured geometry and scales (linearly) in time and space with the complexity of the manufacturing process plan. The detailed description of the general framework is in the next chapter.
3 GENERAL FRAMEWORK

In this chapter, we describe the general framework to the thermal simulation of AM processes that is based on the novel contact-aware path-level (CAPL) discretization and scales (linearly) in time and space with the complexity of the manufacturing process plan.

Conceptually, the thermal simulation generated by the framework consists of two main stages: pre-processing and execution. In the pre-processing stage, the manufacturing primitive (e.g., roads in the FDM process) of the AM process is identified and discretized into elements based on the physical characteristics of the process. A contact graph - a spatial data structure that explicitly captures and represents all the contacts between discrete elements, is created and used to traverse the computing domain. In the execution stage, the contact graph is used to perform transient thermal simulation, updating thermal history for all elements that are significantly affected in each time step.

There are four steps (see Figure 1.1) to instantiate a thermal simulation for a specific AM process: 1) Path-level discretization, which discretizes the manufacturing primitives into elements based on tool path; 2) Construct and update the contact graph; 3) Build the lumped-capacitance model; 4) Design active body that can localize computing. The first two steps construct the CAPL discretization. We describe these steps in detail below in Chapter 3.1-3.3.

3.1 Contact-aware path-level (CAPL) discretization

The construction of CAPL discretization includes two steps: tool path discretization and contact graph construction. The discretization is directly applied to the manufacturing primitive and the contact graph is used to explicitly capture and represent all contact conditions.
3.1.1 Path level discretization

To be manufactured by an AM process, a designed solid model is converted into a process plan containing discrete commands that specify tool path together with printing process specifications (e.g., velocity of the printer head and deposition temperature in FDM process, laser power and scan speed in PBF process). Figure 3.1 shows an example of G-Code file used in the FDM process, which defines the path of moving printer’s head. As the heat source moves on the path, it transforms the raw material into solidified *as-manufactured* shape. It is intuitively clear that the as-manufactured shape is a sweep of solidified material over the path travelled by the heat source, but the cross-section of this sweep varies significantly depending on the AM process. This as-manufactured shape must be discretized for simulation purposes. There are at least two reasons why voxelization is inappropriate for this task. The as-manufactured geometry is determined by complex overlap and contact conditions between the swept roads, leading to (nearly) non-manifold shapes that are not likely to be resolved by any finite size voxelization. Secondly, the actual as-manufactured shape may not be known a priori. For example, in PBF process, the manufactured shape is determined by the melt pool, which in turn is determined by the thermal history of the process. Instead, we initialize the simulation by discretizing the manufacturing path into line segments corresponding to discrete elements; the actual shape of these elements will be determined by the process and (possibly) thermal history.

Thus, in the pre-processing stage, the tool path information is extracted from the process plan and is discretized into many short line segments based on segmentation parameters (i.e., maximum/minimum element length). Each line segment is called a sub-path and we define *element* as material created as the result of the first phase change from raw medium to solid material. For example, in the simulation of FDM process, an element is defined as material extruded through moving printer head along the corresponding sub-path, whereas in the simulation of PBF process, an element is defined as newly melt/sintered material by laser scan along the corresponding sub-path. Two user defined parameters, maximum element length
Figure 3.1: An example of G-code file used in FDM process.

(\(\Delta x_{\text{max}}\)) and minimum element length (\(\Delta x_{\text{min}}\)) determine the lengths of the sub-paths that are guaranteed to lie in the range \([\Delta x_{\text{min}}, \Delta x_{\text{max}}]\).

Figure 3.2 shows a discretization example of one layer in a octopus part [103]. In the figure, the tool path is represented by blue solid lines, and element centers are denoted by small red dots.

In our framework, each element in the discretization represents a small closed domain and two such elements can only overlap on boundaries. The whole as-manufactured domain (\(\Omega\)) equals to the union of all \(n\) elements.

\[
\Omega = \bigcup_{i=1}^{n} \omega_i = \sum_{i=1}^{n} \omega_i, \tag{3.1}
\]
where $\omega_i$ represents the domain of $i$-th element. The mapping from a sub-path (a space line segment) to the 3D geometry (i.e., size and shape) of an element is modeled by sweeping of the cross-section shape along the sub-path. Different processes will have different mappings because the notion of the element depends on the process as well. For instance, in the simulation of FDM process, the cross-section shape for an element is determined by the size and shape of the extrusion nozzle and is fixed. In other processes, such as PBF process, the cross-section size and shape for an element are also influenced by thermal history, which may change during the simulation. Also, since a realistic process plan is composed of a piecewise linear path with different lengths, the lengths of elements are non-uniformly distributed.

In addition to spatial discretization, the simulation is also discretized in time. It should be noted that elements and time steps do not need to have one-to-one
correspondence such that a new element is deposited in each time step. Multiple elements may be created in a single time step, and multiple time steps may be needed to resolve the shape of a single element. Due to the nature of additive manufacturing, it is safe to say the number of elements and number of time steps in the simulation have a positive correlation in the sense that an increased number of elements implies more time steps needed to simulate the manufacturing process. This implies $m = \Theta(n)$, where $m$ is the number of time steps, and $n$ is the number of elements.

### 3.1.2 Contact graph

We observe that the discretized domain is essentially a cell complex defined by the union of all elements. The boundary of each element is partitioned into areas of contact with other elements or free space. Each such contact area is a face of the element in the complex. We introduce a spatial data structure called *contact graph* representing connectivity (i.e., adjacency information between elements) of the cell complex. Formally, contact graph is the dual one-dimensional cell complex, where 0-cells correspond to the 3D elements, and 1-cells correspond to the faces in the primal cell complex representing the discretized as-manufactured domain. In the execution stage, the primal cell complex is used to perform transient thermal simulation, updating thermal history for all elements that are significantly affected in each time-step. Contact graph provides an elegant layer of abstraction that allows efficient traversal of the discretized domain.

The contact graph, $G = (\mathcal{V}, \mathcal{E})$ is comprised of a set $\mathcal{V}$ of vertices, where each vertex represents an element, together with a set $\mathcal{E}$ of edges corresponding to contacts between adjacent elements. Surface contact between elements $i$ and $j$ is represented by an edge $e_{i,j} \in \mathcal{E}$. The information (i.e., contact area, center distance) which is needed to compute heat transfer on the edge is stored in the weight of the edge. The maximum number of elements in contact with a given element is bounded by a constant. This stems from the requirement on minimum size (and hence finite possible contact area) of an individual element and the fact that every
element can be in contact only with elements in adjacent paths within the same layer or in adjacent layers (i.e., above and below its layer). Thus, the number of contacts of an element is not related (e.g., proportional) to the total number of elements in the domain. We claim that contact graph is a sparse graph which implies \( |E| = O(|V|) \). In other words, the average degree of a vertex is \( O(1) \), thus the size of contact graph is proportional to the number of elements \( O(|V|) \). From the above discussion, we can see contact graph is a sparse undirected graph. Thus adjacency list is usually used to represent the contact graph in implementation.

Contact graph is initialized by the manufacturing process plan in the pre-processing stage. The elements are ordered according to the process plan, detection of potential contact is conducted on every element. Space partitioning techniques could be used to localize the contact detection query such that initializing contact graph requires only \( O(|V|) \) time. In our simulation, a binning algorithm is applied to partition the domain into bins and each bin is indexed by a 3-tuple of integers \((x, y, z)\). The value \( z \) represents the layer number, \( x \) and \( y \) defines the coordinate of the bin within the layer. Each element keeps track of its bin number and each bin also stores a list of element indexes that are in the bin. As illustrated in Figure 3.3, element centers are represented by dots, the scan path is denoted by a blue solid line, bins are denoted by yellow dashed lines. Only elements (red) within the same or neighboring bins of the \( i \)-th element (green) are checked when computing \( i \)-th element’s contact within the same layer. Since element position is fixed during the simulation, the bin data structure is built once in linear time \( O(|V|) \) during pre-processing which has no influence on the whole time complexity.

A contact graph example of a part with two layers is shown in Figure 3.4. This part is printed by aligned raster paths. In the figure, the scan path is represented by blue solid lines; element centers are denoted by red dots; contacts between adjacent elements are represented by red solid lines. Apparently, it is a sparse graph.

During the execution stage described in Chapter 3.4, contact graph is used to traverse the domain and determine which elements are active in each time step as well as manage thermal interaction between contacting elements.
Figure 3.3: Illustration of binning algorithm.

Figure 3.4: Example of contact graph.
3.2 Lumped-capacitance model

The standard approach for formulating a transient thermal problem is based on the conservation of energy, which states that the volume’s energy variation over a given time period is equal to the summation of heat produced in the volume and the net in-flowing heat. The integral governing equation of heat conservation is written as

$$\int_\tau \int_\omega \rho c \dot{T}(t) dV dt = \int_\tau \int_{\partial \omega} Q \cdot n dA dt + \int_\tau \int_\omega \sigma dV dt,$$

where $\tau$ represents the time interval, $\omega$ represents the domain of interest, $\rho$ is density, $c$ is the constant pressure specific heat capacity of the material, $T$ is temperature, $\partial \omega$ denotes the boundary of $\omega$, $Q$ denotes the thermal flux on the domain’s boundary, $n$ is the normal vector of $dA$, and $\sigma$ is the heat production rate which accounts for the heat source term in the domain. In the above equation, the left-hand side (LHS) accounts for the energy variation in the domain, the first term on the right-hand side (RHS) accounts for energy exchange on the boundary, the second term on RHS accounts for energy generated inside the domain.

The correct temperature field history should satisfy the conservation of energy (Equation 3.2) on arbitrary time ($\forall \tau$) and space ($\forall \omega$) setting. Considering the complicated geometry of realistic parts as well as material nonlinearity, this transient heat transfer problem normally does not admit analytical solutions. Numerical methods are usually applied to solve the equation approximately.

In order to solve it numerically, the conservation of energy is enforced on a discrete time and space setting. Since the size of an element is relatively small, its Biot number is low, which means the element is determined to be “thermally simple”, which implies heat conduction within an element is much faster than heat transfer across the boundary of the element [32]. The “thermally simple” determination allows us to assume uniform temperature distribution in the element. In other words, it is assumed that the temperature within the element is completely uniform.
in space, although this spatially uniform temperature value changes over time. A
**lumped-capacitance** model [32] can be built to approximate the transient conduc-
tion. During pre-processing, the evolving fabrication domain is discretized into
many elements according to the process plan. Subsequently, the heat conservation
equation for i-th element can be obtained.

\[ m_i c_i \dot{T}_i = Q_i + \sigma_i, \quad (3.3) \]

where \( m_i \) is the mass of the i-th element; \( Q_i \) denotes the heat transfer term
on the boundary of i-th element, and \( \sigma_i \) represents the heat source term which
accounts for the energy absorbed from heat source. For example, in the PBF process,
\( \sigma_i \) represents the laser heating term which accounts for the energy absorbed from
the laser scan.

The heat transfer term on the boundary of i-th element can be divided into
several thermal effects: convection, radiation, and conduction.

\[ Q_i = Q_i^{\text{conv}} + Q_i^{\text{radi}} + Q_i^{\text{cond}}, \quad (3.4) \]

where \( Q_i^{\text{conv}}, Q_i^{\text{radi}}, Q_i^{\text{cond}} \) represent convective, radiative and conductive heat
transfer term respectively. Both Equation 3.3 and 3.4 are general equations that work
for all AM processes driven by a moving heat source. Assuming all the material
properties are fixed, the formulation of each heat transfer term is elaborated as
follows.

The convective term accounts for the energy dissipated \(^1\) from a free surface to
the ambient environment through heat convection, which can be expressed as

\[ Q_i^{\text{conv}} = h_{\text{conv}} A_i^{\text{free}} (T_\infty - T_i(t)), \quad (3.5) \]

\(^1\)positive \( Q_i^{\text{conv}} \) means heat absorbed through convection.
where $h_{\text{conv}}$ is the convective heat transfer coefficient; $A_i^{\text{free}}$ is the free surface area of $i$-th element; $T_\infty$ denotes the ambient temperature.

In addition to heat convection, there is energy emitted from the free surface through radiation. The radiative term is expressed by Stefan-Boltzmann law

$$Q_i^{\text{radi}} = \epsilon \sigma_{SB} A_i^{\text{free}} (T_\infty^4 - T_i(t)^4), \quad (3.6)$$

where $\epsilon$ is a dimensionless parameter called the emissivity coefficient. $\sigma_{SB}$ is the Stefan-Boltzmann constant ($\sigma_{SB} = 5.670 \times 10^{-8} \text{W/(m}^2 \cdot \text{K}^4)$). Kelvin scale should be used when calculating radiation.

The conductive term ($Q_i^{\text{cond}}$) is the most complicated term which includes:

1) conduction to neighboring elements (i.e., preceding element and succeeding element) along the tool path ($Q_i^{\text{path}}$); 2) conduction to build platform ($Q_i^{\text{plat}}$); 3) conduction to contacting elements in adjacent paths ($Q_i^{\text{contact}}$) and other process related terms. Next the general formulas of $Q_i^{\text{path}}$, $Q_i^{\text{plat}}$ and $Q_i^{\text{contact}}$ are described below.

The heat conduction is driven by temperature difference and temperature gradient along the path is approximated by finite difference.

$$Q_i^{\text{path}} = \lambda \left( A_{i,i-1} \frac{T_{i-1}(t) - T_i(t)}{0.5(L_i + L_{i-1})} + A_{i,i+1} \frac{T_{i+1}(t) - T_i(t)}{0.5(L_i + L_{i+1})} \right) \quad (3.7)$$

Where $\lambda$ is the thermal conductivity. $T_i(t)$ is the temperature of $i$-th element at time $t$; $A_{i,j}$ is the cross-sectional area between $i$-th element and $j$-th element; $L_i$ is the length of $i$-th element.

Furthermore, as elements on the first layer contact the cooler build platform, heat transfer by conduction develops between them. A temperature difference arises at the interface due to the thermal contact resistance. Since the mass of build platform is much higher than that of the element, the conduction heat transfer with the build platform can be considered in the form of convection [6], with the
thermal contact conductance \( h_{\text{plat}} \). In this case, the expression for heat conduction between the \( i \)-th element and the build platform becomes

\[
Q_{\text{plat}, i} = h_{\text{plat}} A_{\text{plat}, i} (T_{\text{plat}} - T_i(t)), \tag{3.8}
\]

where \( A_{\text{plat}, i} \) is the contact area of the \( i \)-th element with build platform; \( T_{\text{plat}} \) is the temperature of the build platform, which is usually a manufacturing parameter, and \( h_{\text{plat}} \) is the thermal contact conductance between the element and build platform.

The contact between two adjacent elements with different temperatures can also be formulated as thermal contact conductance. If \( S_i \) is a set of elements contacting \( i \)-th element, heat transfer to contacting elements can be written as

\[
Q_{\text{contact}, i} = \sum_{j \in S_i} h_c A_{i,j}^c (T_j(t) - T_i(t)), \tag{3.9}
\]

where \( A_{i,j}^c \) is the contact area between \( i \)-th element and \( j \)-th element, and \( h_c \) is the thermal contact conductance coefficient between two contacting elements.

For some AM processes, \( Q_{\text{cond}}^i \) may contain some process-related terms. For example, \( Q_{\text{cond}}^i \) includes heat conduction to neighboring powders in the simulation of the PBF process. Also, different AM processes may use different materials, which are modeled differently in the simulation.

### 3.3 Scalability and active body

For thermal simulation to be scalable, it must achieve linear complexity in time and in space, in order to support real-time response and control of the manufacturing processes for realistically complex components. As we already observed, the time complexity is determined by both the number of time steps and the cost of performing numerical simulation at every time step. Suppose there are \( n \) elements and
m time steps (with a positive correlation between m and n, m = Ω(n)). Naively, O(mn) space is needed to store the temperature history of all elements assuming each record takes constant space. Furthermore, the complete thermal simulation of an AM process requires O(mw) computational cost, where w is the total cost in any one time step and could be as high as O(n^3) for some numerical methods. This dramatically limits the size of feasible simulations both in time and in space. Such simulations are clearly not scalable.

In contrast, our proposed numerical scheme only requires constant time to update the temperature of an element in one time step, which implies w = Ω(n) and the total time complexity of simulation is O(mn). This is a great improvement compared with O(mn^3), but quadratic time complexity is also not likely to be practical for realistic simulation scenarios with hundreds of thousands or million elements/time steps. It is clear that linear time complexity (O(m)) is only possible if the update at every time step takes on average constant time (O(1)). Linear time complexity also requires that O(m) space is used to store the full temperature history.

We overcome the difficulty by observing that the majority of elements are not affected by the moving heat source at any given time, because they either already cooled down or are far away from the heat source. An element is formed a long time ago or far away from the current location of the heat source has reached thermal equilibrium with its surroundings. So updating the temperature of all the elements in every time-step is neither efficient nor necessary. The simulation is localized by introducing the concept of “active body”, which is a subset of contact graph. In each time step, only updating the temperature of elements in the active body. With the above observation, **active body should cover all the elements which are “close” to the current heat source either in time or space.**

Define P^t_0 to be the location of heat source at time t. Conceptually the active body (A^t ⊂ V) of elements at time t contains two parts A^t_temp and A^t_spat, and A^t is the union of the two parts (A^t = A^t_temp U A^t_spat). A^t_temp is the temporal locality part which includes elements close to P^t_0 in time. A^t_spat is the spatial locality part which includes elements close to P^t_0 in space. Time difference is easy to retrieve
since elements are ordered sequentially by their forming time. Also, space distance between two elements could be represented by the length of the shortest path between two elements on contact graph (assuming every edge weight equals 1). During the manufacturing process, a typical temperature history of an element involves initial heating and cooling and several weaker heating-cooling cycles afterwards. The functionality of $A_{\text{temp}}^t$ is to keep elements in active body for its initial heating-cooling cycle. The functionality of $A_{\text{spat}}^t$ is to capture the subsequent (re)heating-cooling cycles by pushing elements on the heat propagation path to active body. Different AM processes may lead to different designs of temporal active body and spatial active body.

During the simulation, the active body is updated in every time step. Since the size of contact graph grows linearly with respect to the number of elements in the domain ($|E| + |V| = \mathcal{O}(|V|)$), and active body only contains a small subset of elements in the contact graph. With proper design, the number of elements in the active body is bounded by a constant at any given time. Updating active body also takes constant time ($\mathcal{O}(1)$).

Elements that are inactive at time $t$ have relatively low temperature and the net heat transfer of an inactive element with other elements is low. Hence the heat exchanges between inactive elements may be replaced by simpler temperature update schemes, even using analytic solutions (e.g., Newton’s law of cooling), with little or no loss of accuracy. Furthermore, the temperatures of inactive elements can be updated on demand, in a lazy fashion, using similar simplified solutions. Such lazy update does not affect the running time of the simulation.

The size of active body is controlled by several parameters, selected based on physical tests and the desired tradeoff between simulation accuracy and efficiency.

3.4 Simulation

In the execution stage, a numerical scheme is picked to update element temperature. There are many options, but fundamentally two choices must be made in selecting a numerical scheme: implicit/explicit and low/high order.
Implicit schemes require solving a system of linear equations ($Ax = y$) in each time step, potentially leading to $O(n^3)$ complexity. Hence, an explicit method is more suitable than an implicit method for realistic process plans. Compared with an implicit scheme, an explicit scheme is simple to implement and does not require solving a system of linear equations in each time step. However, explicit schemes do not have unconditional stability which means it requires choosing small $\Delta x$ and $\Delta t$ to satisfy stability condition, which in turn could lead to a large number of time steps.

With high order methods, it is possible to use a larger time step to achieve the same level of accuracy as low order methods. However, in our specific problem (thermal simulation of AM process driven by a moving heat source), the element length is determined by both segmentation size and the tool path and in the simulation of some processes the time step is positively correlated with element length, it is impossible to use large time step, which means there are no benefits using higher order methods. Thus, we conclude that a low order explicit method is a good choice for the general simulation framework.

A good first-order explicit method is standard **Forward Euler** (FE) time marching scheme which is formulated as

$$T_i(t + \Delta t) = T_i(t) + \Delta t \cdot \dot{T}_i(t)$$

$$= T_i(t) + \Delta t \cdot \frac{Q_i(t) + \sigma_i(t)}{m_i c_i}$$  \(3.10\)

For other options, one can refer to **Runge-Kutta** methods which are a family of implicit and explicit iterative methods. Forward Euler method belongs to this family. Normally, for parts with complicated geometry, Forward Euler method is a good choice. If the tool path is composed of only long line segments, the second-order Runge-Kutta method may be a better choice.

It should be noted that the choice of the numerical method is a separate issue from CAPL discretization. For example, we could start with CAPL discretization and conceivably refine it into a classical finite element mesh in order to use standard
finite element analysis. This would require the geometric representation of all contacts, and will satisfy the requirements of accurate discretization, but is not likely to achieve desired computational efficiency. Instead, the proposed numerical method (lumped-capacitance method) is capable of achieving high computational efficiency when applied directly to CAPL discretization. This is achieved by a numerical scheme that propagates the temperature history through the contact structure of CAPL discretization which localizes computation based on empirical and simulation data.

An algorithm of general simulation is summarized in Algorithm 1. Initially, all the elements are inactive and elements are activated in order according to the process plan. Different AM processes have different printing procedures which may lead to different element activation methods. Elements can be activated one by one or layer by layer. For some processes (e.g., PBF process), element size may change after the heat transfer update, leading to a change in an element’s contacts and requiring update of the contact graph.

By localizing computation inside the active body which contains at most constant number of elements, the total execution achieves linear time complexity (\(O(m) = O(n)\)) and hence is scalable. This in turn implies that the constructed simulation not only scales to handle three-dimensional (3D) printed components of arbitrary
complexity but also can achieve real-time performance.

**Algorithm 1: Algorithm of General Simulation**

| input  | :geometry (i.e., tool path), material properties, manufacturing parameters |
| output | :temperature history of each element |

discretize manufacturing primitives into elements based on tool path;
initialize contact graph;

**foreach** time step **do**
  activate elements based on process plan;
  update active body;
  **foreach** element in active body **do**
    calculate heat transfer to it;
    calculate its new temperature;
  **end**
  **foreach** element in active body **do**
    update its temperature;
    update its size and contact graph when needed;
  **end**
**end**

3.5 Discussion

In this chapter, we describe a general framework to the thermal simulation of AM processes driven by a moving heat source that accurately models as-manufactured geometry and scales (linearly) in time and space with the complexity of manufacturing process plan. The salient features of the simulations include accurate discretization of as-manufactured geometry, whose topology is represented by the (dual) contact graph that captures the adjacency information between elements in the discretization. The scalability of the simulation is achieved by exploiting spatial and temporal locality, based on empirical process data, which allows updating thermal state of at most a constant number of elements at each time step.
This framework can be implemented as a general computing platform that contains several modules. The interface, which could be seen as a set of promises that a module can do, is well defined. And two modules can only communicate with each other through their interfaces. The implementation of each module is hidden from other modules, such that every pair of modules are decoupled and each module has plug in compatibility. In order to improve code reusability, some modules (e.g., discretization) and functions can be made general, which are applicable to multiple AM processes. In order to apply the framework to a new AM process, one needs to instantiate the four components (path-level discretization algorithm, contact graph data structure, lumped-capacitance heat-transfer model, and active body localization algorithm) of the framework and implement (or specify) the modules for this specific AM process. Through this, all the process related detail is buried in the implementation of modules. And the module interfaces are generic. Essentially, this framework provides a way to decompose the problem and manage complexity.

In order to demonstrate the efficacy and generality of the framework, we applied it to build thermal simulations of two different AM processes, fused deposition modeling (FDM) and powder bed fusion (PBF) in the following two chapters.
4 FUSED DEPOSITION MODELING

4.1 Introduction

Fused deposition modeling (FDM), also known as fused filament fabrication (FFF) is a 3D printing process that uses a continuous filament of a thermoplastic material. In order to be manufactured using FDM, a designed solid model (i.e., CAD model) is converted into a process plan that specifies the tool path of printer head together with the printing process specifications, such as build direction, nozzle diameter, and infill percentage [49]. Perhaps, the most common example of this scenario is a process plan generated by process planning software such as Slic3r [37] from stereolithography format describing the solid’s boundary representation. In the course of manufacturing, as the printer head moves along the tool path, the raw filament is fed into the heated liquefier by rollers, and the semi-molten filament is extruded through a nozzle which is mounted on the printer head. For each layer, the nozzle moves following a piecewise linear path horizontally. The material deposited along each line segment is commonly referred to as a road. A schematic of the FDM extrusion process is shown in Figure 4.1.

FDM process is driven by a moving heat source, and the temperature history of the deposited road plays an important role in determining the mechanical properties and geometry of the printed components. The formation of bonds between adjacent roads is induced by the thermal energy of semi-molten material [76], and the repeated heating and cooling of the materials can aggregate nonuniform thermal gradient and cause stress accumulated that consequently result in part distortion [104]. Thermal analysis not only is important to understand (and predict) different thermally induced manufacturing failures (such as delamination and warpage) but also provides the basis for the design of manufacturing process plan, like tool path design and optimization of manufacturing parameters. Broadly speaking, the previous research on thermal simulation of FDM process can be divided into two categories: 1) simplifying the problem such that it can be solved analytically; 2) continuum-based methods, which are based on voxelization of
design geometry (usually approximated by finite element methods). As discussed in Chapter 2.1, neither of the approaches is scalable to simulate the manufacturing process of realistically complex components. Due to the simplified assumptions, analytical models are limited to simple cases (e.g., cooling profile of a single road). Continuum-based models based on voxelized design geometry can not account for significant differences between as-designed and as-manufactured geometry and tend to be computationally prohibitive.

In this chapter, the proposed general framework (described in Chapter 3) is applied to build a thermal simulation of FDM process that accurately models as-manufactured geometry and scales (linearly) in time and space with the complexity of manufacturing process plan. In the model, the tool path is discretized into many sub-paths. An element is defined as **material extruded through moving printer head along the corresponding sub-path**. Contact graph is constructed in the pre-processing stage to capture adjacency information between elements. A lumped-capacitance model that accounts for the most relevant thermal effects including heat convection and radiation to the environment, heat conduction along the road,
heat conduction with build platform, heat conduction between adjacent roads as well as between adjacent layers, is built to perform the thermal simulation. The simulation achieves linear time complexity by localizing the computation inside “active body”, which allows updating at most a constant number of road elements at each time step. This implies that our simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance. The simulation is fully implemented, validated against known analytic solutions, and is tested on realistic complex shapes.

The rest of this chapter is organized as follows: The detailed formulation is described in Chapter 4.2. In it, CAPL discretization is introduced in Chapter 4.2.1. Chapter 4.2.2 and 4.2.3 describe lumped-capacitance model and active body. Some numerical experiments are covered in Chapter 4.3.

### 4.2 Formulation

Conceptually, the thermal simulation generated by the framework consists of two stages: pre-processing and execution. In the pre-process stage, the CAPL discretization is constructed in two steps. In the first step, the manufacturing primitives (deposited roads) are discretized into elements based on the tool path. Then, a contact graph - a spatial data structure representing adjacency between discrete elements, is initialized and used to traverse the computing domain. In the execution stage, a lumped-capacitance model is built and the contact graph is used to perform transient thermal simulation, updating thermal history for all elements that are significantly affected in each time step.

#### 4.2.1 CAPL discretization

##### 4.2.1.1 Tool path discretization

In FDM process, the printer is controlled by an input file (e.g., G-code file). The input file defines the tool path which is composed of many line segments. The G-code file and tool path of a single layer part from [55] is shown in Figure 4.2. In
pre-processing, each line segment in the tool path is discretized into one or more short line segments, called sub-paths. Each sub-path with material deposition corresponds to an element. Note that the printer head can also move without extruding any material. An element is defined as the material extruded through moving printer head along the corresponding sub-path. The detailed mapping from sub-path to element is described in the next section. The elements will be used to simulate the actual deposition process, with elements being deposited sequentially at distinct time steps. In each time step, if there is material deposited, activate the element associated with the material.

Figure 4.2: G-code file (left) and tool path (right) of a single layer part.

The detailed discretization process is described as follows. The tool path information (many line segments) is read into the program and all sequential collinear line segments are merged into one line segment. Two user defined segmentation parameters are used to discretize the line segments: maximum element length $\Delta x_{\text{max}}$ and minimum element length $\Delta x_{\text{min}}$. A long segment with length $L > \Delta x_{\text{max}}$ is divided into $\lceil L/\Delta x_{\text{max}} \rceil + 1$ elements. Since very short element may cause numerical prob-
lems, an element with length \( L < \Delta x_{\text{min}} \) is ignored. This happens rarely because typical \( \Delta x_{\text{min}} \) is very small relative to printer’s accuracy (e.g., \( \Delta x_{\text{min}} = 0.5 \text{ mm} \)) and such segments should not appear in the process plan. As a result, it is guaranteed that the length of each element lies in the range \([\Delta x_{\text{min}}, \Delta x_{\text{max}}]\).

![Figure 4.3: Discretization of a single layer part in Figure 4.2.](image)

The discretization of the above single layer part is shown in Figure 4.3, where the tool path is represented by solid lines and element centers are denoted by small red dots. Also, line segments on boundaries (i.e., the contour paths), where curvatures are large, tend to be shorter, as they control the accuracy of the part’s geometry set by the manufacturing processing software. So elements on the contour path tend to be shorter than elements on the raster path.

Initially, the elements are stored in an array sequentially according to the process plan. Each element record also includes its index, position in the global Cartesian coordinate, length, deposition time, and temperature history. Subsequently, as we
describe next, this array is transformed into a contact graph data structure that adds explicit adjacency information for each element.

4.2.1.2 Contact graph

The contact graph, which keeps track of all the elements and their adjacency information, is built in the pre-processing stage. Contact detection is conducted on each element in pre-processing stage. It should be noted that any given element can be in contact only with elements in the adjacent roads within the same layer or with elements in the adjacent (above or below) layers. Based on the discussion in Chapter 3.1.2, each element can only contact with at most constant ($O(1)$) number of neighboring elements. So the size of contact graph is linear in the number of elements ($O(|V|)$). Space partitioning method could be used to localize contact detection which avoids global searching. In this way, constructing the contact graph only requires $O(|V|)$ time. It is reasonable to assume that the position of an element and its contact information with neighboring elements remain unchanged during the simulation. Here we assume that bond grows instantaneously between two adjacent elements. Hence the contact graph is a static data structure that can be built once in the pre-processing stage for a given process plan. Since contact graph is a sparse graph, adjacency list is used to implement it. The $i$-th element in the constructed graph stores a list of contacting element indexes ($S_i$) and the corresponding contact areas ($\{A_{i,j} \mid j \in S_i\}$) computed with the method described below.

The mapping from a sub-path to an element is described as follows. An element is formed by sweeping the cross-section of the road along its sub-path as illustrated in Figure 4.4. In order to model the cross section of the road, consider the microphotograph of the cross section of an FDM part from [77] reproduced in Figure 4.5. It suggests that the contacting surfaces of the road can reasonably be approximated by the four planar regions: two vertical sides of the road where they contact other roads within the same layer, and two horizontal sides that can contact the roads in the layers immediately above and below the road. The precise manner
in which these sides are connected is not important, as long as we can estimate the contact and noncontact areas of the road. For concreteness, we approximate the road’s cross-section by a rectangle with four corners cut (Figure 4.6). The shape of the cut corner is assumed to be an isosceles triangle. More generally, the size of the cross-sectional shape depends on specific process parameters (e.g., speed of printer head and rate of deposition), though we will not deal with this dependence here.

The contact parameters $w$ and $h$ are easily determined from elementary geometry of the cross-section. $H$ and $W$ are preset manufacture parameters obtained from the process plan: $H$ is the layer height and $W$ is the road width. Another manufacturing parameter called extrusion factor $e$ ($0 < e < 1$), determines how full the rectangle $W \times H$ is filled by the road. Assuming the material is not compressible, the material conservation is expressed as

$$WH - \frac{1}{2}(W - w)(H - h) = eWH$$

(4.1)

where $h$ and $w$ represent the interface neck length between adjacent roads within the same layer and between adjacent layers, respectively (see Figure 4.6). As shown in Figure 4.5, $h = 2\gamma$ and the isosceles triangle implies

\[
\text{Cross section of road} \
\text{Road axis} \\
\text{Road direction}
\]
Figure 4.5: Microphotograph of the cross-sectional area of a FDM part, where $W$ is the road’s width, $H$ is the layer height, and $2y$ is the neck length between adjacent roads within the same layer [77].

\[ W - w = H - h \quad \text{(4.2)} \]

Combining Equation 4.1 and 4.2, we can obtain

\[ w = W - \sqrt{2(1 - e)WH} \quad \text{(4.3)} \]

\[ h = H - \sqrt{2(1 - e)WH} \quad \text{(4.4)} \]

With these parameters, the actual contact areas are determined by considering the contacts made by sweeps of these cross-sections along the road’s trajectories. We must distinguish contacts between roads within the same layer (Figure 4.7 and 4.8) and contacts between the roads in adjacent layers (Figure 4.9). In the first case, the contact area is approximated by a product of neck length $h$ and either contact
length $L'$ when the roads are parallel or road width $W$ when the road end contacts another road.

For contacting elements in adjacent layers, the contact area is determined by the overlap of the top view (build direction) projections of the elements, as illustrated in Figure 4.8. Computed area of two overlapping rectangles is a classroom problem in computational geometry (e.g., see [78] for reference). In our simulation, Sutherland-Hodgman algorithm is implemented to compute the overlapping rectangle.

![Figure 4.6: Cross-sectional shape of road.](image)

### 4.2.2 Lumped-capacitance model

Considering the thermal properties of the common material used in FDM, short element length, and small road cross-section, it is easy to check that Biot number of the element is less than 0.1, which implies that the heat conduction inside the element is much faster than the heat convection away from its surface, hence temperature gradient inside the element is negligible. So it is reasonable to assume the temperature within the element completely uniform in space, although this spatially uniform temperature value changes over time. Therefore a lumped-capacitance
The heat conservation equation for i-th element is written as
Figure 4.9: Top view of contact between two elements in vertically adjacent layers.

\[ m_i c_i \dot{T}_i = Q_i + \sigma_i \]  

(4.5)

In FDM process, the raw material is heated to a preset temperature before it is squeezed out of the nozzle, and there is no extra heat generated inside the element, so the heat source term on RHS of the above equation is zero (i.e., \( \sigma_i = 0 \)). Then Equation 4.5 is simplified to

\[ m_i c_i \dot{T}_i = Q_i, \]  

(4.6)

where \( Q_i \) denotes the heat transfer term on element boundary which could be divided into three different thermal effects.

\[ Q_i = Q_i^{\text{cond}} + Q_i^{\text{conv}} + Q_i^{\text{radi}} \]  

(4.7)
where $Q_{\text{cond}}^i, Q_{\text{conv}}^i, Q_{\text{radi}}^i$ represent heat conduction, convection and radiation on the boundary of $i$-th element. The conductive term in the above equation includes 1) heat conduction along (i.e., conduction to preceding contact element and succeeding contact element along the road) the road ($Q_{\text{path}}^i$); 2) heat conduction to build platform ($Q_{\text{plat}}^i$); 3) heat conduction to contact elements within the same layer and in adjacent layers ($Q_{\text{contact}}^i$). Substitute these terms into Equation 4.7, we can get the below formula (see Figure 4.10 for illustration of different thermal effects on the boundary of an element).

$$Q_i = Q_{\text{path}}^i + Q_{\text{conv}}^i + Q_{\text{radi}}^i + Q_{\text{plat}}^i + Q_{\text{contact}}^i$$

(4.8)

Figure 4.10: Illustration of different thermal effects on element boundary.

In the lumped-capacitance model, temperature is assumed to be uniformly distributed spatially within each element. The expressions for each heat transfer
term on RHS of Equation 4.8 are elaborated below.

First, consider the heat conduction along the road. Heat conduction is driven by temperature difference. Temperature gradient along the road is approximated by finite difference.

\[
Q_{i}^{\text{along}} = \lambda \left( A_{i,i-1} \frac{T_{i-1}(t) - T_{i}(t)}{0.5(L_{i} + L_{i-1})} + A_{i,i+1} \frac{T_{i+1}(t) - T_{i}(t)}{0.5(L_{i} + L_{i+1})} \right) \tag{4.9}
\]

Where in FDM simulation; \(T_{i}(t)\) is the temperature of \(i\)-th element at time \(t\); \(A_{i,j}\) is the cross-sectional area between \(i\)-th element and \(j\)-th element; \(L_{i}\) is the length of \(i\)-th element.

In addition to heat conduction along the road, there is energy dissipated from free surface to the ambient environment through heat convection, which plays a major role during the cooling process of FDM. And heat convection on the free surface can be expressed as

\[
Q_{i}^{\text{conv}} = h_{\text{conv}} A_{i}^{\text{free}} (T_{\infty} - T_{i}(t)) \tag{4.10}
\]

where \(h_{\text{conv}}\) is the convective heat transfer coefficient. In [68], Rodriguez et al. adopted the Churchill correlation for natural convection to estimate \(h_{\text{conv}} = 67 \text{ W}/(\text{m}^{2} \cdot \text{K})\) for an elliptical road deposited at 270\(^{\circ}\)C toward environment at 70\(^{\circ}\)C. \(A_{i}^{\text{free}}\) is the free surface area of \(i\)-th element which is computed in pre-processing by subtracting contacting area from the surface area. \(T_{\infty}\) denotes the ambient temperature which is set to be the chamber temperature of the printer (or the room temperature if the printer does not have a chamber).

Thermal radiation has usually been neglected in the previous work. However, we found in the numerical experiments that the ratio \(Q_{\text{radi}} / Q_{\text{conv}}\) could be as large as 0.25 (when the temperature is high). Hence radiation is covered in our simulation. The radiative term is given by Stefan-Boltzmann law.
\[ Q_{i}^{\text{radi}} = \varepsilon \sigma_{SB} A_{i}^{\text{free}} (T_{\infty}^4 - T_{i}(t)^4), \quad (4.11) \]

where \( \varepsilon \) is a dimensionless parameter which is called the emissivity coefficient. In our simulation, \( \varepsilon \) equals to 0.96 [13]; \( \sigma_{SB} \) is the Stefan-Boltzmann constant \( (\sigma_{SB} = 5.670 \times 10^{-8} W/(m^2 \cdot K^4)) \). It should be noted that Kelvin scale should be used when calculating radiation.

Furthermore, as a deposited road in the first layer contacts the cooler build platform, heat transfer by conduction develops between them. A temperature difference arises at the interface due to the thermal contact resistance. Since the mass of build platform is much higher than that of the road, the conduction heat transfer with the build platform can be considered in the form of convection [6], with the thermal contact conductance \( h_{c} \). In this case, the expression for heat conduction between \( i \)-th element and the build platform becomes

\[ Q_{i}^{\text{plat}} = h_{c} A_{i}^{\text{plat}} (T_{\text{plat}} - T_{i}(t)), \quad (4.12) \]

where \( A_{i}^{\text{plat}} \) is the contact area of the \( i \)-th element with the build platform; \( T_{\text{plat}} \) is the temperature of build platform, and \( h_{c} \) is the thermal contact conductance between road and build platform.

The contact between two adjacent roads with different temperatures can also be formulated as thermal contact conductance. If \( S_{i} \) is a set of elements contacting \( i \)-th element, heat transfer to contacting elements could be written as

\[ Q_{i}^{\text{contact}} = \sum_{j \in S_{i}} h_{r} A_{i,j}^{c} (T_{j}(t) - T_{i}(t)), \quad (4.13) \]

where \( A_{i,j}^{c} \) is the contact area between element \( i \) and \( j \), and \( h_{r} \) is the thermal contact conductance coefficient between two contacting roads.
4.2.3 Active body

Assuming that a single element is deposited at each time step, the complete thermal simulation of an FDM process with \( n \) elements will require \( O(n^2) \) temperature updates. Like we discussed in Chapter 3.3, quadratic time complexity is not scalable for realistic simulation scenarios with hundreds of thousands or million time steps.

In order to overcome this difficulty to build a scalable thermal simulation, the concept of active body is used to localize the temperature updates in a subset of all the deposited elements. As we discussed before, a good design of active body relies on a better understanding of heat transfer in the specific AM process. An important observation of FDM is the majority of elements are not affected by the deposition of a new element, because they either already cooled down or are far away from the newly deposited element. For example, empirical data suggests that for ABS P400 material, it takes about \( p = 8 \) s to cool down (close) to the envelop temperature \[6\] and the thermal influence of the newly deposited element does not propagate through more than \( q \) contacts with neighboring elements due to thermal contact resistance on the interface. Thus, an element that is deposited a long time ago or far away from the current deposition area has reached thermal equilibrium with its surroundings. So updating temperature of all the deposited elements in every time step is neither efficient nor necessary. Using the above observation, we design the active body which covers all the elements that are “closed” to the current deposition area either in time or space.

The detailed definition of active body is introduced as follows. Active body \((A_t \subset \mathbb{V})\) of elements at time \( t \) is the union of two sets \( A_t^{\text{temp}} \) and \( A_t^{\text{spat}} \).

\[
A_t = A_t^{\text{temp}} \cup A_t^{\text{spat}}
\]  

(4.14)

The temporal active body \( A_t^{\text{temp}} \) includes all elements deposited in the last \( p \) seconds immediately preceding time \( t \), i.e., during the time interval \([t - p, t]\). The parameter \( p \) is a predefined parameter based on physical observation of the cooling profile of a single road. The elements in the temporal active body \( A_t^{\text{temp}} \) are in
contact with their neighbors, which may or may not be in $A_{\text{temp}}^t$. As these elements get reheated, they conduct the heat to their neighbors and so on. Formally we define the contact $q$-neighborhood $N^j_q$ of element $j$ in the contact graph $V$ to be the set of elements $\{k \mid \text{dist}(k, j) \leq q\}$, where $\text{dist}(k, j)$ is the length $^1$ of the shortest path between the elements $k$ and $j$ in the contact graph. However, as the elements in $A_{\text{temp}}^t$ cools off, their effect on neighboring elements diminishes. Hence, we assume that the reheating effects are significant only in the contact $q$-neighborhoods of elements in the set $A_{\text{core}}^t \subset A_{\text{temp}}^t$, which is defined to be the set of elements deposited in the last $s$ time-steps prior to time $t$. In other words, we define the spatial active body at time $t$ as

$$A_{\text{spat}}^t = \bigcup_{j \in A_{\text{core}}^t} N^j_q$$

Spatial active body $A_{\text{spat}}^t$ could be computed through conducting breadth first search (BFS) from $A_{\text{core}}^t$ for $q$ depth which requires $O(|A_{\text{spat}}^t|)$ operations.

![Diagram](image)

Figure 4.11: Active body $A^t = A_{\text{temp}}^t \cup A_{\text{spat}}^t$.

---

$^1$The length of path between the two nodes in the graph is the number of edges in the path.
Figure 4.11 shows an example of a printing path, as well as the core, spatial, and temporal active bodies at a time instant \( t \) (element \( t \) is the newly deposited element which is in the third layer). From it, we can see \( A^t_{\text{core}} \) covers all the (most) recently deposited elements. The functionality of \( A^t_{\text{temp}} \) is to capture the initial cooling phase in the temperature history. The functionality of \( A^t_{\text{spat}} \) is to capture the reheating and cooling cycles afterwards.

Informally, since the core elements are deposited most recently, they have high temperature and thus interact actively with their neighbors. The elements of the active body that are not in the core are about to leave the active body and have relatively low temperatures, which means their thermal interaction with the neighboring elements outside of the active body is relatively insignificant. These interactions are still captured during the temperature update, but we assume that the thermal energy does not propagate any further.

During the simulation, the active body is updated in every time step. If no new elements are deposited, both \( A^t_{\text{temp}} \) and \( A^t_{\text{spat}} \) shrink in size as elements leave the active body. If a new element is deposited at time \( t \), it is added to the temporal active body \( A^t_{\text{temp}} \) and its \( q \)-neighborhoods is added to the spatial active body \( A^t_{\text{spat}} \). It should be clear that each update of active body takes at most constant time (\( O(1) \)). Furthermore, we claim that, for any given discretization and fixed parameter values (\( p \), \( q \) and \( s \)), the number of elements in the active body is bounded by a constant at any given time. It is fairly obvious that the size of temporal active body \( |A^t_{\text{temp}}| \) bounded by a constant determined by \( p \). To see that it is also true for spatial active body \( A^t_{\text{spat}} \), observe that the maximum number of elements in contact with a given element is bounded. This stems from the requirement on minimum size (and hence finite possible contact area) of an individual element. It follows that the maximum degree \( k \) of a vertex in the contact graph is also bounded. A loose upper bound on the spatial active body is then obtained immediately as \( |A^t_{\text{spat}}| \leq sk^{q+1} \). However, the bound is fairly conservative, elements are repeatedly counted in the formula. For example, in a typical simulation, with \( p = 8 \), \( q = 3 \) and \( s = 150 \), the average size of active body \( |A^t| \) is about 1,000. This active body design follows the guide described in Chapter 3.1.2 and temporal locality is defined by \( p \) and spatial locality.
is defined by \( q \) and \( s \).

In every time step of the simulation, the temperature of all elements that are active at time \( t \) (i.e., elements in active body \( A^t \)) are updated. Since the cost of such an update is bounded by a constant, the whole simulation takes \( O(m) \) time, where \( m \) is the total number of time steps. Since \( m \) is approximately equal to the number of elements \( n \), the whole simulation achieves linear time complexity \( (O(n)) \).

Elements that are inactive at time \( t \) (elements that are not in active body \( A^t \)) have relatively low temperatures, and the net heat transfer of an inactive element with other elements is low. We ignore the heat exchange of an inactive element with other elements. Due to low temperature, both radiation and heat conduction to the build platform are also ignored. Hence we only consider heat convection which means the rate of heat loss of an (inactive) element is proportional to the difference in temperatures between the element and the environment. So Newton’s law of cooling [32] could be used to update their temperatures. In other words, temperatures of inactive elements are updated on demand, when these elements are queried (e.g., when an element becomes active again). This mechanism is called lazy update. Lazy update does not affect the running time of the simulation.

The size of active body is determined by the parameters \( p, q, \) and \( s \). Their values are selected based on physical observations and the desired tradeoff between simulation accuracy and efficiency. A larger active body requires more temperature updates and hence longer running time, but it may improve the accuracy of the computed results.

### 4.2.4 Simulation

Forward Euler (FE) time marching is used to update the temperature.

\[
T_{i}^{n+1} = T_{i}^{n} + \Delta t^{n} \cdot \hat{T}_{i}^{n}
\]  

(4.16)

Where \( T_{i}^{n} \) denotes the temperature of \( i \)-th element at time \( t^{n} \) and \( \Delta t^{n} \) is the \( n \)-th time step, \( \Delta t^{n} = t^{n+1} - t^{n} \); The temperature derivative at time \( t^{n} \) (\( \hat{T}_{i}^{n} \)) is computed
with Equation 4.6 where \( Q_i \) is computed using Equation 4.9 to 4.13. The formula of \( \dot{T}_i^n \) is written as

\[
\dot{T}_i^n = \frac{Q_i(t^n)}{\rho c V_i} = \frac{1}{m_i c_i} \left( \lambda \left( A_{i,i-1} \frac{T_{i-1}(t^n) - T_i(t^n)}{0.5(L_i + L_{i-1})} \right) + A_{i,i+1} \frac{T_{i+1}(t^n) - T_i(t^n)}{0.5(L_i + L_{i+1})} \right) + 

h_{\text{conv}} A_i^{\text{free}} (T_\infty - T_i(t^n)) + \epsilon \sigma B A_i^{\text{free}} (T_\infty^4 - T_i(t^n)^4) + 

h_c A_i^{\text{plat}} (T_{\text{plat}} - T_i(t^n)) + \sum_{j \in S_i} h_r A_{i,j} (T_j(t) - T_i(t^n)) 
\]

(4.17)

where \( \rho \) is the density of the material which is assumed to be fixed in the simulation; \( c \) is the specific heat capacity which is also assumed to be fixed in the simulation; \( V_i \) represents the volume of \( i \)-th element (see Figure 4.12 for illustration). And this formula assumes \( i \)-th element has both preceding contact element and succeeding contact element. If not, the corresponding term is zero. Also, heat conduction to build platform only exists for elements on the first layer.

The thermal simulation is executed in a stepwise manner by activating the elements in the contact graph that are affected by heat exchange during the FDM process. Before the process begins, all the elements are inactive. The thermal analysis initiates with the current temperature distribution as the initial condition. In each time step, the printer head either moves without extruding any material or extrudes amount of material associated with some element in the graph. And the above numerical scheme (Equation 4.17) is used to update temperature of elements in the active body. The update assumes that material extrusion takes place instantly at the start of each time step. Since the scheme is explicit, it does not require solving a system of equations and can be performed simultaneously (and in parallel) for all elements. The initial (final) cross-section of the first (last) element on each sub-path is treated as free boundaries.

The full algorithm of the simulation is summarized in Algorithm 2. By localizing
computation inside the active body which contains at most constant number of elements, the total execution achieves linear time complexity and hence is scalable. This in turn implies that the constructed simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance.

In the simulation, the time step is proportional to element length. The element length is chosen such that the simulation remained stable and the results did not change when the element length was further reduced ($\Delta x_{\text{max}} = 1 \text{ mm}$). We want to show the stability and convergence of the proposed numerical scheme. As an explicit method, Forward Euler is conditional stable. From Equation 4.17, we can see that the most strict stability condition is enforced by using Forward-Time Central-Space (FTCS) method to approximate the diffusion term. FTCS can yield unstable solutions that oscillate and grow if $\Delta T$ is too large. The stability condition [67] for uniform grid is given by
Algorithm 2: Algorithm of FDM simulation

input: tool path, material properties and manufacturing parameters
output: temperature history of each element
discretize tool path into elements;
built contact graph;
foreach time step do
    if there is material deposited in this time step then
        activate the element associated to the material;
    end
    update active body;
    foreach element in active body do
        calculate heat transfer to it;
        calculate its new temperature;
    end
    foreach element in active body do
        update its temperature;
    end
end

\[ \Delta t < \frac{(\Delta x)^2}{2K}, \]  \hspace{1cm} (4.18)

where \( K = \frac{\lambda}{\rho c} \). Since in our simulation the element length lies in the range \([\Delta x_{\text{min}}, \Delta x_{\text{max}}]\), we show that the numerical scheme is stable for the more restrictive case of a uniform grid with \( \Delta x_{\text{min}} = 0.5 \) mm and acrylonitrile butadiene styrene (ABS) P400 material that is widely used in FDM process. The thermal properties of ABS P400 were reported by \[6\], which are listed below. Substituting the thermal properties and \( \Delta x_{\text{min}} \) into Equation 4.18, stability condition becomes \( \Delta t < 1.5424 \) s.

We now consider two separate cases. During the manufacturing process, when the nozzle travels without extruding any material, travel time \( t_{\text{travel}} = \frac{\text{Distance}}{v_{\text{travel}}} \), where \( v_{\text{travel}} \) is the speed of printer head during traveling without extruding any material. If \( t_{\text{travel}} > 0.1 \) s, it is discretized such that time step \( \Delta t < 0.1 \) s. And when there is an element extruded in this time step, time step \( \Delta t = \frac{\Delta x}{v_{\text{infill}}} \) where \( v_{\text{infill}} \) is the speed of
Table 4.1: Properties of ABS P400

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal conductivity, $\lambda$ (W/m · K)</td>
<td>0.177</td>
</tr>
<tr>
<td>Specific heat, $c$ (J/kg · K)</td>
<td>2080</td>
</tr>
<tr>
<td>Density, $\rho$ (kg/m$^3$)</td>
<td>1050</td>
</tr>
</tbody>
</table>

printer’s head while extruding amount of material. Substituting this into Equation 4.18, we conclude that the stability condition requires $v_{\text{infill}} > \frac{2K}{A_x} \geq 0.3242$ mm/s. Since the deposition speed of FDM is much greater than 1 mm/s, it is safe to claim that the proposed numerical scheme is stable during simulating the manufacturing process. The thermal simulation may also continue after the 3D printing process completes; in this case, we simply choose the constant time step of $\Delta t = 0.1$s, which obviously lies in the stability region. Thus, in both cases, the proposed numerical scheme is stable. Therefore, the Lax equivalence theorem [48] implies that the proposed numerical procedure converges.

4.3 Numerical experiments

The proposed simulation is fully implemented and tested in this Chapter. Chapter 4.3.1 validates the simulation by applying it to simulate the deposition of a single road. Chapter 4.3.2 discussed the influence of active body on computational efficiency and accuracy. The proposed simulation is applied to simulate a realistic part in Chapter 4.3.3.

4.3.1 Simple accuracy test

To test the accuracy of the proposed simulation, we applied it to a simplified problem - deposition of a single road - which has the analytical solution [6] and compared the computed cooling profile with that given by the analytical solution. A single deposition road is modeled as a one-dimensional elliptic sweep. The head moves at
a constant speed of \( v \) along the \( x \)-axis when extruding. The origin of the reference coordinate system is set at the extrusion nozzle (see Figure 4.13).

Consider only heat conduction and convection, the governing partial differential equation reduces to

\[
\rho c \lambda \frac{\partial T}{\partial t} = A \lambda \frac{\partial^2 T}{\partial x^2} - hP(T - T_\infty)
\] (4.19)

where \( A \) and \( P \) are the area and the perimeter of the elliptic cross-section, respectively. The boundary conditions are defined as

\[
T = \begin{cases} 
T_0 & x = 0 \text{ and } t \geq 0 \\
T_\infty & x = \infty \text{ and } t \geq 0,
\end{cases}
\] (4.20)

where \( T_0 \) is the deposition temperature and \( T_\infty \) is the environment temperature. The origin is moving at a velocity \( v \ (x = vt) \), and the time dependence term \( \frac{\partial T}{\partial t} \) is transformed to

\[
\frac{\partial T}{\partial t} = \frac{\partial T}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial T}{\partial x} v
\] (4.21)
As is shown in [6], Equation 4.19 is then reduced to an ordinary differential equation with analytic solution
\[ \rho c A v \frac{\partial T}{\partial x} = \Lambda \frac{\partial^2 T}{\partial x^2} - h P (T - T_\infty) \] (4.22)

The analytical solution of Equation (4.22) with the boundary conditions defined by Equation 4.20 is as follows:

\[ T = T_\infty + (T_0 - T_\infty) e^{-mx}, \] (4.23)

with \( m = \sqrt{1+4\alpha \beta} \frac{-1}{2\alpha} \) and \( x = vt \), where \( \alpha = \frac{\lambda}{\rho c v} \) and \( \beta = \frac{h P}{\rho c A v} \).

We applied the proposed numerical simulation to this problem to compute the cooling profile of the road. As shown in Figure 4.14, the results of the simulation using time step \( \Delta t = 0.1 \text{s} \) are in good agreement with the analytic solution of Equation 4.23.

Figure 4.14: Comparison of simulation (\( \Delta t = 0.1 \text{s} \) results with the analytic solution from [6])
4.3.2 Influence of active body

In this chapter, the influence of active body on computation results and efficiency are studied using a more realistic part (an octopus part) commonly used for testing FDM process. The tool path model of the octopus part is shown in Figure 4.15. In order to make it clear, different sub-paths are plotted in randomly picked colors. This part is sliced into 10 layers. After the path discretization, there are 59,595 elements in the whole part.

![Tool path model of the octopus part.](image)

We applied the thermal simulation to simulate the manufacturing process of the first $k$ layers of the octopus part ($k$ varies from 1 to 9) with and without active body. Parameters of active body are chosen as $p = 8s$, $q = 3$, $s = 150$. The running times of both simulations are compared in Figure 4.16 (the loglog scale plot is in Figure 4.17). It can be easily seen that, without active body, the running time grows much faster ($O(n^2)$) with regards to the number of time steps as opposed to linear time ($O(n)$) performance when using active body.
Figure 4.16: Influence of active body on simulation time of FDM process.
Next, the influence of active body on computation result is studied. In this part, we want to show proper parameters (i.e. $p$, $q$ and $s$) of active body could be chosen based on physical observations and using proper parameters to define active body, the simulation could obtain nearly the same result as that without active body. However active body can reduce the number of temperature updates per time step dramatically.

The value of $p$ is related to the initial cooling, during which temperature drops under transitional temperature after deposition. Hence $p$ should be large enough to cover the initial cooling. Roughly speaking, $q$ and $s$ together control the shape of active body, among them $q$ controls the depth of active body which is used to capture the reheating scenario, whereas $s$ controls the size of the horizontal cross-section of active body. The choice of proper parameters depends on the specific process plan (e.g. material properties, deposition temperature, and envelop temperature,
etc.). We can choose $p$ by observing the cooling profile of one road through either experiments or simulations. The value of $q$ could be set as the maximum penetration depth which could also be obtained by experiments or simulations. The choice of $s$ depends on desired trade-off between accuracy and efficiency. Based on our numerical experiments, $A_{\text{core}}$ should cover at least one fourth to one third of the hottest elements in $A_{\text{temp}}$ to capture the reheating phenomena well. So $A_{\text{core}}$ could also be set as a fixed ration of $A_{\text{temp}}$.

The proposed simulation is applied to simulate the manufacturing of the first 10 layers of the octopus part, which has 24,340 road elements and 24,965 time steps in the discretized tool path. Based on our numerical experiments, active body parameters are chosen as $p = 8 s$, $q = 3$, $s = 150$. A road element in the third layer is picked to study its temperature profile. The reference solution is given by simulation without active body (updating temperature of all deposited elements in each time step). The comparison of the computed temperature profile is shown in Figure 4.18.

It can be seen from Figure 4.18 that the blue dashed line (simulation with active body) agrees well with the red line (simulation without active body). The mean absolute error (MAE) is 0.3559 °C and mean absolute percentage error (MAPE) is 0.48%. Also, the average number of temperature updates per time step with active body is 1,149, whereas the average number goes up to 12,162 when active body is not used (all the deposited elements are updated in each element). From it we can see under the help of active body, the simulation can get nearly the same temperature result with much fewer temperature updates.

To confirm that the choice of parameters does not depend on geometry, we applied the same active body parameters (i.e. $p = 8 s$, $q = 3$, $s = 150$) to simulate two other parts with the same material and manufacture parameters as the octopus part. Figure 4.19 shows the result of simulating the manufacturing of first 10 layers of the mobius arm part whose 3D model is shown in Figure 4.20. An element in the second layer is chosen to study its temperature profile. It can be seen in Figure 4.19 that the computed temperature profile agrees well with the reference solution shown in red line (MAE = 0.3823 °C and MAPE = 0.49%). Figure 4.21 shows the
result of simulating the manufacturing of first 5 layers of the aerospace rocker part whose 3D model is shown in Figure 4.22. An element in the second layer is picked to compare its temperature profile. From Figure 4.21, we could see active body also performed well with MAE = 0.4676 °C and MAPE = 0.63%.

### 4.3.3 Realistic numerical simulation

The 3D model for a realistic part is shown in Figure 4.20 and the tool path of the mobius arm part is shown in Figure 4.23.

The tool paths for printing support material are shown in blue, whereas tool paths for printing the part’s shape are shown in mixed colors. This part was discretized into 69,635 road elements. The temperature fields predicted by the thermal simulation at different times are shown in Figure 4.24, where road elements are displayed as line segments with different colors according to their respective temperatures.
Complete simulation of 70,410 time steps in the printing process took 24.052 seconds to execute. The reference computer system configuration included 2.8 GHz quad-core Intel Core processor (Intel Core i7-4980HQ) with 16 gigabytes of RAM, running Java 1.8 atop Apple macOS sierra 10.12.3. In comparison, it takes about 821.913 seconds to 3D print the part using an FDM printer whose nozzle moves at 6 cm/s, which means the simulation achieves real-time performance.
Figure 4.20: 3D model of a “mobius arm” part (Courtesy of Stratasys, Ltd.)

Figure 4.21: Influence of active body on computation results of an aerospace rocker part.
Figure 4.22: 3D model of an aerospace rocker part (Courtesy of Stratasys, Ltd.).

Figure 4.23: Tool path of the "mobius arm" part.
Figure 4.24: Snapshots of thermal simulation for the manufacturing of mobius arm part, color indicates the temperature.
4.4 Discussion

In this chapter, we applied the proposed general framework to build a scalable thermal simulation of FDM process. The simulation is performed directly on the as-manufactured model, derived from the manufacturing process plan, and includes detailed geometry of the printed part as well as the deposition order in which this geometry is created. In the pre-processing stage, the tool path is discretized into many sub-paths and an element is defined as the material extruded through moving printer heat along the corresponding sub-path. The contact graph, which is a static data structure in FDM process capturing the adjacent information of elements is built once in pre-processing. A lumped-capacitance model, which covers all the important thermal effects developed during the manufacturing process, including heat convection and radiation to the environment, heat conduction with build platform, along the road and between adjacent roads (and adjacent layers) is built to conduct the thermal simulation. The computation is localized by active body, which covers elements close to the heat source spatially and temporally. Comparing with the algorithm of general simulation (Algorithm 1), the algorithm of FDM simulation (Algorithm 2) is simpler. The simulation has been fully implemented and tested. As we showed both theoretically and experimentally, the simulation is numerically stable, converges, and achieves linear time complexity. This also implies that the simulation not only scales to handle 3D printed components of arbitrary complexity but also can achieve real-time performance. The computed temperature field history by the simulation could be used to predict properties (e.g., bond strength between adjacent roads) of the printed components and detect manufacturing failures (e.g., warping). Another possible extension is to use the insights gained from simulation to design tool path plans to achieve desired temperature field(s). One challenging aspect of this work is that we are not aware of any other reference implementations or experimental measurements that could be used to validate the correctness of computed results. Finally, the proposed simulation has real-time performance, which could be used to build a feedback control of manufacturing parameters.
5 POWDER BED FUSION

5.1 Introduction

In the last few years, there has been a significant increase in the use of industrial metal AM processes to fabricate functional (e.g., structural and load-bearing) parts due to their high design flexibility and significant weight savings [8]. The powder bed fusion (PBF) process is a widely used AM technology to produce metallic parts. Figure 5.1 depicts a schematic of a typical PBF machine. PBF process is driven by a moving heat source. During the manufacturing process, a laser (or electron beam) locally melts and consolidates metal powders along a predefined tool path. Upon completion of each layer, additional powder is spread onto the bed from an adjacent reservoir and evenly distributed by a roller or a blade. This process continues until the whole part is finished (or the build fails).

![Schematic of a typical powder bed fusion set up.]

Figure 5.1: Schematic of a typical powder bed fusion set up [2].

Microstructure, distortion, and mechanical properties of parts fabricated with the PBF process depend heavily on the temperature field evolution during the
manufacturing process. Thermal field evolution during the manufacturing process plays an important role in determining both geometric and mechanical properties of the fabricated parts. To improve quality control and optimize manufacturing parameters in PBF process, it is critical to have a quantitative understanding of the relationship between process inputs (i.e., manufacturing parameters and scan path) and thermal field evolution. Computational models and simulations help to quantify and understand heat transfer during the manufacturing process.

Thermal simulation of the PBF process is computationally challenging due to the geometric complexities offered by the manufacturing process and the inherent computational complexity that requires a numerical solution at every time increment of the process. Powder-based high fidelity simulation models can be overly computationally expensive, to the extent that they are not capable of simulating realistically complex parts. (Semi-)analytical thermal models relying on over-simplified assumptions are not suitable for simulating realistic process plans. Common continuum-based thermal models relying on voxelization of the design geometry ignore the obvious fact that design geometry differs significantly from the as-manufactured shape [49]. Since traditional numerical methods (e.g., finite element method) require temperature updates of all the elements at every time step, the complete thermal simulation of the PBF process with \( n \) elements and \( m \) time steps requires \( O(mw) \) computational cost, where \( w \) is the total cost in any one time step and could be as high as \( O(n^3) \). This cost dramatically limits the size of feasible simulations both in time and in space. Such simulations are clearly not scalable. To be scalable, simulations should grow linearly with increased complexity in geometry (space) and time (number of time steps).

In this chapter, the proposed general framework is applied to build a thermal simulation of PBF process that accurately models as-manufactured geometry and scales (linearly) in time and space with the complexity of the manufacturing process plan. Our approach is unique in that it does not restrict itself to simulations on the part design geometry, but instead simulates the formation of the geometry based on the process plan of a part. The implication of this distinction is that the simulations are in tune with the as-manufactured geometry. This means that
localized calculations are more aligned with the process than the design, which is a more realistic abstraction of real-world behavior.

In our model, the scan path is discretized into many sub-paths. **An element is defined as the material that has been newly melted/sintered by a laser scan along the corresponding sub-path.** An element growth mechanism is introduced to simulate the evolution of a melt pool during the manufacturing process. A spatial data structure, called contact graph, is built to capture adjacency information between elements. A lumped-capacitance model that accounts for the most relevant thermal effects including laser heat absorption, convection and radiation to environment, conduction along the scan path, conduction to the build platform, conduction to the powder bed, and conduction to contacting elements, is built to perform the thermal simulation. The simulation achieves linear time complexity by localizing the computation inside a subset of elements called “active body” through exploiting spatial and temporal locality. Active body limits the need to update to at most a constant number of elements at each time step. This strategy implies our simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance. The output of the simulation is the complete temperature history of each element over the course of a manufacturing process. The simulation is fully implemented and validated against experimental data and other simulation results.

The rest of this chapter is organized as follows: The detailed formulation is described in Chapter 5.2. In it, CAPL discretization is introduced in Chapter 5.2.1. Chapter 5.2.2, 5.2.4, 5.2.3 and 5.2.5 describe lumped-capacitance model, element growth mechanism, modeling of material properties and active body respectively. Numerical experiments and validation are included in Chapter 5.3.

### 5.2 Formulation

Conceptually, the thermal simulation generated by the framework consists of two stages: pre-processing and execution. In the pre-processing stage, the manufacturing primitive (material melted/sintered by the laser scan) is discretized into
elements based on the scan path. A contact graph - a spatial data structure representing adjacency between discrete elements, is initialized and used to traverse the computing domain. An element growth mechanism is introduced to capture the element’s cross-section change during the simulation. In the execution stage, a lumped-capacitance model is built and the contact graph is used to perform transient thermal simulation, updating thermal history for all elements that are significantly affected in each time step.

5.2.1 CAPL discretization

5.2.1.1 Scan path discretization

In PBF process, the machine is controlled by an input file that defines the scan path (composed of many line segments) as well as the laser power corresponding to the path. Different machines may have different input file formats. Figure 5.2 plots the scan path of a single layer part. In it, the scan path with laser on is plotted in red and the scan path with laser off is plotted in blue. During the pre-processing stage, the scan path and laser power information are extracted from the input file.

Then each line segment in the process plan is discretized into one or more short line segments, called sub-paths using the same approach described in Chapter 3.1.1. Figure 5.3 plots the discretization of the part in Figure 5.2. In it, the element centers are represented by red dots. Each sub-path with positive laser power corresponds to an element. Since laser power may not be constant, each element should also keep track of the laser power value. Element is defined as newly melt/sintered material by laser scan along the corresponding sub-path. The mapping from sub-path to the corresponding element’s shape and size is described in the next section.

Unlike FDM simulation, in which element is deposited sequentially to simulate the deposition process, in PBF simulation, as a new layer begins, all the elements on this layer are activated and can participate in the simulation immediately because the laser beam can preheat the powders in its neighborhood. Initially, elements are in powder state. During the simulation, the powder material is heated and melted by a laser scan. The actual laser scan process is simulated by moving the
laser beam on the path (elements) according to the process plan. The simulation is also discretized in time. Temperature changes very quickly in PBF process, in order to capture the behavior, a very short time step is used and multiple time steps are needed to resolve the shape of a single element.

The elements are stored in an array sequentially according to the process plan. Each element keeps track of its size, position in global coordinate, scan time, layer number as well as the temperature history. Also at a given time, the position of the laser beam center is easily obtained from this information. Subsequently, as we described next, this array is transformed into a contact graph data structure that adds explicit adjacency information for each element.
5.2.1.2 Contact graph

The simulation is conducted on a contact graph comprising a set $V$ of vertices (elements) together with a set $E$ of edges corresponding to contacts between adjacent elements. In order to compute contact information between adjacent elements, we need to know the size and shape of each element. An element is modeled as a sweep of the melt pool cross-section along its sub-path, which may not be laterally symmetric due to asymmetric thermal conditions on both sides. Each element keeps track of its left width ($W_{\text{left}}$) and right width ($W_{\text{right}}$). The cross-sectional shape is approximated by a rectangle $W \times H$, where $W$ is the width of this element ($W = W_{\text{left}} + W_{\text{right}}$) and $H$ is the layer height. The initial width of an element is $W_0$, which is a predefined parameter. If the center distance $D$ between the element $i$ and its left closest parallel element $j$ is less than (or equal
to) $W_0$, then its initial left width is $D/2$ and it is left contact with element $j$. If $D$ is greater than $W_0$, then its initial left width is $W_0/2$. The value of $W_0$ could be determined by empirical data (e.g., laser spot size), high-fidelity simulations, or data-driven models. Element size is also influenced by the thermal history. During the simulation, an element may grow its width to left and right independently. The generation and growth of elements during the simulation capture the laser scan and powder melting phenomenon during the manufacturing process. A more detailed description of element growth mechanism is in Chapter 5.2.4.

Since the size of elements can change during the simulation, contact graph is no longer a static data structure that can be built once in the pre-processing stage. Contact graph is initialized by the manufacturing process plan in pre-processing stage and updated during the simulation. And it is still a sparse graph due to the fact that an element can only contact with elements in its (close) neighborhood within the same layer or adjacent layers. The elements are ordered according to the process plan, contact detection is conducted on every element. The actual contact areas are determined by considering the contacts made by sweeps of these cross-sections along the element’s trajectories. Space partitioning techniques could be used to localize the contact detection query such that initializing contact graph requires only $O(|V|)$ time. In our simulation, a binning algorithm is applied to partition the domain into bins and each bin is indexed by a tuple of three integers $(x, y, z)$. The value $z$ represents the layer number, $x$ and $y$ defines the coordinate of the bin within the layer. Each element keeps track of its bin number and each bin also stores a list of element indexes that are in the bin. As illustrated in Figure 5.4, element centers are represented by dots, scan path is denoted by blue solid lines, bins are denoted by yellow dashed lines. Only elements (red) within the same or neighboring bins of the $i$-th element (green) are checked when computing $i$-th element’s contact within the same layer. Since element position is fixed during the simulation, the bin data structure is built once in $O(|V|)$ time during pre-processing which has no influence on the whole time complexity.

A contact graph of a three-layer part scanned using aligned raster path is shown in Figure 5.5. In the figure, scan path is represented by blue solid lines; element
5.2.2 Lumped-capacitance model

Due to its short length and thin layer height, the Biot number of an element is low, which means the element is determined to be “thermally simple” [32]. The “thermally simple” determination allows us to assume a uniform temperature at the interior of the element. A lumped-capacitance model is built upon the elements to conduct the thermal simulation. In the pre-processing stage, the evolving fabrication domain is discretized into many elements based on the laser scan path. The thermal system is reduced to a number of discrete elements (lumps) and temperature distribution within the element is completely uniform in space, although this spatially uniform temperature value changes over time. Subsequently, the heat
conservation equation for the i-th element could be written as

$$m_i c_i \dot{T}_i = Q_i + \sigma_i,$$

(5.1)

where $m_i$ is the mass of the i-th element, and $m_i = \rho_i (1 - \gamma)V_i$, $\rho$ is the density of bulk material and $\gamma$ is the porosity. $V_i$ is the volume of i-th element; and $c$ is the constant pressure specific heat capacity of the material. The modeling of material properties is discussed in Chapter 5.2.3. $T$ is temperature, $Q_i$ denotes the heat transfer term on the boundary of i-th element, $\sigma_i$ is the heat source term which accounts for energy generated inside of the element. In PBF process, this term exists and it represents energy absorbed from the laser scan. Conventionally, $H$ is used to represent the laser heating term, so Equation 5.1 is rewritten as
\[ m_i c_i \dot{T}_i = Q_i + H_i, \]  

(5.2)

The heat transfer on i-th element’s boundary \( Q_i \) could be divided into three thermal effects: conduction, convection, and radiation.

\[ Q_i = Q_{i \text{cond}} + Q_{i \text{conv}} + Q_{i \text{radi}}, \]  

(5.3)

where \( Q_{i \text{cond}}, Q_{i \text{conv}} \) and \( Q_{i \text{radi}} \) represent conductive, convective and radiative term on the boundary of i-th element. The conductive term \( Q_{i \text{cond}} \) includes: 1) heat conduction to elements (i.e., preceding contact element and succeeding contact element) along the laser scan path \( Q_{i \text{path}} \); 2) heat conduction to neighboring powders in the powder bed \( Q_{i \text{powder}} \); 3) heat conduction to the build platform \( Q_{i \text{plat}} \); 4) heat conduction to contacting elements in adjacent scans within the same layer or in above/below layers \( Q_{i \text{contact}} \); Substitute these into Equation 5.3, we can get the below formula.

\[ Q_i = Q_{i \text{path}} + Q_{i \text{powder}} + Q_{i \text{plat}} + Q_{i \text{contact}} + Q_{i \text{conv}} + Q_{i \text{radi}} \]  

(5.4)

In the lumped-capacitance model, temperature is assumed to be uniformly distributed spatially within each element. The numerical formulation of each heat transfer term on RHS of Equation 5.4 is described in detail as follows.

First, consider the heat conduction along the scan path. Heat conduction is driven by temperature difference. Temperature gradient along the path is approximated by finite difference.

\[ Q_{i \text{path}} = \lambda_{i,i-1} A_{i,i-1} \frac{T_{i-1}(t) - T_i(t)}{0.5(L_i + L_{i-1})} + \lambda_{i,i+1} A_{i,i+1} \frac{T_{i+1}(t) - T_i(t)}{0.5(L_i + L_{i+1})}, \]  

(5.5)
where $\lambda$ is the thermal conductivity; Due to the large temperature variation during heat transfer, $\lambda$ is a parameter depends on temperature. $\lambda_{i,i-1} = \lambda_{T=T_{i-1}(t)}$ where $T_{i-1}(t)$ is the average temperature of $i$-th element and $(i-1)$-th element; $T_i$ denotes the temperature of $i$-th element; $A_{i,j}$ represents the cross-sectional area between element $i$ and $j$; $L_i$ is the length of $i$-th element.

Some elements are surrounded by powders (on both sides), and energy may be transferred between such elements and neighboring powders in the powder bed when there is a temperature difference. In order to estimate the temperature gradient, the contact is modeled as two powder particles touching with each other, and center distance is approximated by the mean powder diameter $\mu_D$, the formula for $Q_i^{\text{powder}}$ is

$$Q_i^{\text{powder}} = \lambda_{\text{powder},i} \epsilon_{\text{powder}} A_{i}^{\text{side}} \frac{T_{\text{powder}} - T_{i}(t)}{\mu_D}, \quad (5.6)$$

where $T_{\text{powder}}$ is the temperature of the powder bed, which is a manufacturing parameter. $\lambda_{\text{powder},i}$ denotes thermal conductivity at temperature $0.5 \cdot (T_{\text{powder}} + T_{i}(t))$; $A_{i}^{\text{side}}$ is the free side area of the element, which is computed by subtracting contacting surface (with elements in the same layer) from side area. In our simulation, the heat conduction from left side to powder bed is computed separately from heat conduction from right side to powder bed. And $A_{i}^{\text{side}} = A_{i}^{\text{left side}} + A_{i}^{\text{right side}}$. $A_{i}^{\text{left side}}$ does not necessarily equal to $A_{i}^{\text{right side}}$ due to different contact information. In order to compute heat conduction from left side to powder bed, substitute $A_{i}^{\text{left side}}$ into Equation 5.6. Since element can not contact perfectly with powder, a parameter called “contact ratio with powder bed” ($0 < \epsilon_{\text{powder}} < 1$) is introduced to compute the effective contact area ($\epsilon_{\text{powder}} A_{i}^{\text{side}}$). In our simulation, $\epsilon_{\text{powder}} = 0.5$.

Furthermore, as an element in the first layer contacts the cooler build platform, heat transfer by conduction develops between them. A temperature difference arises at the interface due to thermal contact resistance. Since the mass of the build platform is much higher than that of the element, the conduction heat transfer with
the build platform can be considered in the form of convection, with the thermal contact conductance $h_c$. In this case, the formula for heat conduction between $i$-th element and build platform becomes

$$Q_i^{\text{plat}} = h_c A_i^{\text{bottom}}(T_{\text{plat}} - T_i(t)),$$

(5.7)

where $A_i^{\text{bottom}}$ is the bottom area of the element; $T_{\text{plat}}$ is the temperature of the substrate which is a fixed manufacturing parameter. In our simulation, $h_c$ depends on temperature and $\frac{\lambda_{\text{plat}}}{\mu_D}$ is used to approximate $h_c$.

There exists heat exchange between two contacting elements when they have different temperatures. If $S_i$ is a set of elements contacting $i$-th element, heat transfer to contacting element could be written as

$$Q_i^{\text{contact}} = \sum_{j \in S_i} \lambda_{i,j} A_{i,j}^c \frac{(T_j(t) - T_i(t))}{d_{i,j}},$$

(5.8)

where $A_{i,j}^c$ is the contact area between element $i$ and $j$, and $\lambda_{i,j} = \lambda|_{T= T_{ij}(t)}$, $d_{i,j}$ is the center (Euclidean) distance between element $i$ and $j$, $d_{i,j} = ||x_j - x_i||$.

In addition to heat conduction, there is energy dissipated from top free surface to the ambient environment through heat convection. And heat convection on the free surface can be expressed as

$$Q_i^{\text{conv}} = h_{\text{conv}} A_i^{\text{free}}(T_\infty - T_i(t)),$$

(5.9)

where $h_{\text{conv}}$ is the convective heat transfer coefficient and $A_i^{\text{free}}$ is the free surface area facing the ambient environment. In our simulation, $A_i^{\text{free}}$ is the top surface area of the element.

In addition to heat convection, there is energy emitted from the free surface through radiation. The radiative term is given by Stefan-Boltzmann law.
\[ Q_i^{\text{rad}} = \epsilon \sigma_{SB} A_i^{\text{free}} (T_\infty^4 - T_i(t)^4), \tag{5.10} \]

where \( \epsilon \) is the material emissivity and \( \sigma_{SB} \) is the Stefan-Boltzmann constant \( (\sigma_{SB} = 5.670 \times 10^{-8} \text{W/(m}^2 \cdot \text{K}^4)) \). Kelvin scale should be used when calculating the radiation term.

The laser heating term \( H \) only exists for elements in the top layer. In order to compute \( H \), we need to model the laser scan and laser energy absorption. In powder-based simulations (such as discrete element method) each element is modeled as a single spherical particle, hence laser absorption can be modeled in a relatively simple manner. However, the proposed simulation is based on path level, each element is defined as the newly sintered/melt material through scanning laser on the corresponding sub-path, which means an element may be composed of multiple powder particles. The interaction between laser and element is the collection of interaction between laser and powder particles. The laser heating term \( H(x, y) \) for an infinitesimal (centered at \((x, y)\)) is given as

\[ H(x, y) = \alpha I(r, z) \mathrm{d}x \mathrm{d}y, \tag{5.11} \]

where \( \alpha \) is the absorptivity of the material at the wavelength of the laser \((0 \leq \alpha \leq 1) \) \[25\]; \( I(r, z) \) denotes the laser intensity \((\text{W/m}^2)\) which is the function of radial distance \( (r) \) between the laser beam center and point \((x, y)\) and penetration depth \( z \). And \( \mathrm{d}x \mathrm{d}y \) is the projected area of the infinitesimal receiving direct, normal radiation from the laser. In order to compute the laser heating term for an element, \( H(x, y) \) is integrated on the top surface of the element.

In general, laser-beam propagation in PBF can be approximated by assuming that the laser beam has an ideal Gaussian intensity profile, which means the laser intensity decreases exponentially as the radial distance from laser beam’s center increases (Figure 5.6). For laser penetration, a Beer-Lambert type model is used
where laser intensity decreases exponentially with respect to the penetration depth [25]. Then the formula for laser intensity is expressed as

\begin{equation}
I(r, z) = I_0 e^{-\beta z} e^{-2r^2/\omega^2},
\end{equation}

where \( I_0 = \frac{2P}{\pi \omega^2} \) is the peak intensity; \( P \) is the laser power; \( \omega \) is the beam spot size measured to where the intensity falls to \( 1/e^2 \) of the peak intensity; And \( \beta \) is the optical extinction coefficient. Due to multiple reflections that take place off the powder particles, laser energy can penetrate much deeper in than powder bed than a solid material. The theory in [28, 30] is used to compute optical extinction coefficient \( \beta \) as a function of particle diameter \( D \) and powder bed porosity \( \gamma \).

\begin{equation}
\beta = \frac{3(1 - \gamma)}{2\gamma D}
\end{equation}

The metal powders are modeled as sphere particles. And truncated Gaussian distribution is used to model the distribution of particle diameter. This means particle diameter \( D \) follows the normal distribution \( N(\mu_D, \sigma^2) \) and \( D \) lies within
the interval \([D_{\text{min}}, D_{\text{max}}]\). In our simulation, the mean diameter is used to estimate the average optical extinction coefficient \(\beta_{\text{avg}}\) for elements and powder bed porosity \((\gamma)\) is a fixed parameter. Therefore the formula for \(\beta_{\text{avg}}\) is written as

\[
\beta_{\text{avg}} = \frac{3(1 - \gamma)}{2\gamma \mu_D}
\] (5.14)

The top surface of the element is not assumed to be perfectly flat due to the fact that an element is formed by melting powder particles, which means the penetration depth \((z)\) is not perfectly zero. The mean absolute deviation (around the mean) of the powder diameter is used to estimate the average penetration depth \(z_{\text{avg}}\) from the laser beam center. For a normal distributed random variable \(X\) with 0 mean and variance \(\sigma^2\) (i.e., \(X \sim N(0, \sigma^2)\)), the ratio of mean absolute deviation to standard deviation is \(\sqrt{2/\pi}\).

\[
\frac{E|x|}{\sqrt{E(x^2)}} = \sqrt{\frac{2}{\pi}}
\] (5.15)

In other words, the mean absolute deviation is about 0.8 times the standard deviation. The estimated average penetration depth \(z\) of the element could be computed as

\[
z_{\text{avg}} = \sigma_D \sqrt{\frac{2}{\pi}} + 0.5\mu_D
\] (5.16)

The top surface of the element is a rectangle. Suppose the rectangle is axis-aligned (i.e., \([x_1, x_2] \times [y_1, y_2]\)), and laser center is located at point \((x_0, y_0)\), the laser heating term of this element could be computed through integrating \(H(x, y)\) on the top surface of the element, which is written as
\[ H = \int_{x_1}^{x_2} \int_{y_1}^{y_2} H(x, y) \, dx \, dy \]
\[ = \alpha I_0 e^{-\beta \overline{z}_{avg}} \int_{x_1}^{x_2} \int_{y_1}^{y_2} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{\omega^2}} \, dx \, dy \]  
(5.17)

The above integration could be simplified by using the below integration result.

\[ \int e^{-\frac{(x-a)^2}{b^2}} \, dx = -\frac{1}{2} \sqrt{\frac{\pi}{2}} b \times \text{erf} \left( \frac{a-x}{b} \right) + \text{constant}, \]

(5.18)

where \( \text{erf}(x) \) is the error function which is plotted in Figure 5.7.

Figure 5.7: Error function.

Then the integration on RHS of Equation (5.17) could be computed as
\[
\int_{x_1}^{x_2} \int_{y_1}^{y_2} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{\omega^2}} \, dx \, dy = \frac{1}{8} \pi \omega^2 \left( f(x_0, x_2) - f(x_0, x_1) \right) \times \left( f(y_0, y_2) - f(y_0, y_1) \right),
\]

where \( f(a, b) = \text{erf} \left( \frac{\sqrt{2}(a-b)}{\omega} \right) \). Denote \( W = \left( f(x_0, x_2) - f(x_0, x_1) \right) \times \left( f(y_0, y_2) - f(y_0, y_1) \right) \). To obtain the simplified formula of the laser heating term \( H \) for (axis-aligned) elements, plug Equation 5.12 and Equation 5.19 into Equation 5.17.

\[
H = \alpha^2 \frac{2p}{\pi \omega^2} e^{-\beta_{avg} z_{avg}} \frac{1}{8} \pi \omega^2 \times W = \frac{1}{4} \alpha e^{-\beta_{avg} z_{avg}} \cdot P \cdot W
\]

From the above formula, we can see that \( H \) is proportional to laser power \( P \) and material absorptivity \( \alpha \). The value of \( \frac{1}{4} \alpha e^{-\beta_{avg} z_{avg}} \) is fixed and could be pre-computed in the simulation. The main work in computing laser heating term \( H \) is to compute \( W \) which requires evaluating the error function four times. The value of \( W \) is determined by the relative position of laser beam center and the element.

In order to compute \( H \) for an element with arbitrary orientation, the element is rotated to be axis-aligned before applying Equation 5.20. The same transformation is also applied to the laser center, which keeps its relative position to the element unchanged. Essentially this is a transformation of coordinates and the element is axis-aligned in the new coordinate system.

As illustrated in Figure 5.8, the position of a path element and the laser center are shown in green. In order to compute the laser heating term \( H \) for this element at time \( t \), we first rotate the element and laser center (w.r.t element’s center) to be axis-aligned (shown in blue dashed line), then Equation 5.20 is applied to compute \( H \). The rotation angle (i.e., angle between an element and global x-axis) is fixed and the rotation matrix is pre-computed in the simulation. The rotation angles of elements are also “cached” during the simulation. It should be noted that, as an
element’s width changes, the rotation of the element needs to get updated.

![Figure 5.8: Illustration of coordinate transformation when computing H.](image)

5.2.3 Material properties

In our thermal simulation of FDM [103], fixed material properties (i.e., specific heat capacity \(c\), thermal conductivity \(\lambda\), density \(\rho\)) are assumed. This assumption is no longer applicable to the thermal simulation of PBF process. In PBF process, a moving heat source is used to consolidate material (e.g. metal) in powder form to form three-dimensional objects. The powder is melt or even evaporated during the manufacturing process, which leads to a much higher temperature variation than that in FDM process. And according to numerical experiments, using different material properties can have a big influence on the computed temperatures. Therefore material properties are dependent on temperatures in the proposed simulation.

Use 316L stainless steel (316L SS) as an example, the specific heat capacity \(c\), thermal conductivity \(\lambda\) and density \(\rho\) values as a function of temperature are given in Table 5.1 [25]. In our simulation, we applied the same technique used in the literature, that properties are linear interpolated at temperature values between the specified data points. The data is extrapolated up to 1400 K (the final measured values are at 1255 K), after which the property values are fixed until phase change at 1700 K.
Table 5.1: Material properties for 316L stainless steel as a function of temperature [25]

<table>
<thead>
<tr>
<th>Temperature (K)</th>
<th>Specific heat (J/kg K)</th>
<th>Thermal conductivity (W/mK)</th>
<th>Density (kg/m³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>293</td>
<td>452</td>
<td>13.3</td>
<td>7952</td>
</tr>
<tr>
<td>366</td>
<td>485</td>
<td>14.3</td>
<td>7919</td>
</tr>
<tr>
<td>478</td>
<td>527</td>
<td>15.9</td>
<td>7877</td>
</tr>
<tr>
<td>589</td>
<td>548</td>
<td>17.5</td>
<td>7831</td>
</tr>
<tr>
<td>700</td>
<td>565</td>
<td>19.0</td>
<td>7786</td>
</tr>
<tr>
<td>811</td>
<td>573</td>
<td>19.8</td>
<td>7739</td>
</tr>
<tr>
<td>922</td>
<td>586</td>
<td>21.9</td>
<td>7692</td>
</tr>
<tr>
<td>1033</td>
<td>615</td>
<td>23.2</td>
<td>7640</td>
</tr>
<tr>
<td>1144</td>
<td>649</td>
<td>24.6</td>
<td>7587</td>
</tr>
<tr>
<td>1255</td>
<td>690</td>
<td>26.2</td>
<td>7537</td>
</tr>
<tr>
<td>1700 (liquid)</td>
<td>815</td>
<td>32.4</td>
<td>7300</td>
</tr>
</tbody>
</table>

Using the data listed in Table 1, we can get the below conductivity (Figure 5.9) and density curves (Figure 5.10). The red point denotes the melting point. The other data points are represented by green points. The blue point is the extrapolated point at 1400 K.

Specific heat capacity is more complicated than thermal conductivity and density due to the latent heat. Latent heat is thermal energy released or absorbed, during a constant-temperature process. To account for absorption and release of latent heat in phase change as the material heats up and subsequently melts, the apparent heat capacity method is used, similar to the method outlined by Bonocina et al. [9] and more recently by Muhieddine et al. [59]. In this technique, the energy needed for phase change to happen is taken into account by computationally raising the specific heat capacity of the material in a small range ($\Delta T$) around the melting or vaporization temperature. Specific heat capacity is represented by a staircase function. The algorithm is outlined as follows.
Figure 5.9: Conductivity of 316L stainless steel.

\[
\alpha(x) = \begin{cases} 
\frac{c_{\text{solid}}}{2} + \frac{L_m}{s \Delta T}, & T < T_m - \frac{\Delta T}{2} \\
\frac{c_{\text{solid}} + c_{\text{liquid}}}{2}, & T_m - \frac{\Delta T}{2} \leq T \leq T_m + \frac{\Delta T}{2} \\
\frac{c_{\text{liquid}}}{2} + \frac{L_v}{c \Delta T}, & T_m + \frac{\Delta T}{2} < T < T_v - \frac{\Delta T}{2} \\
\frac{c_{\text{gas}}}{2}, & T > T_v + \frac{\Delta T}{2}
\end{cases}
\] (5.21)

where \( T_m \) is the melting temperature, \( T_v \) is the vaporization (or boiling) temperature, \( L_m/s \) is latent heat of melting or solidification, and \( L_v/c \) is the latent heat of vaporization or condensation.

With the above algorithm, we can obtain the capacity curve shown in Figure 5.11. In order to make it clear, the figure does not cover the vaporization point. The curve shows that capacity jumps when temperature is around melting temperature which takes account of the melting latent heat.
Figure 5.10: Density of 315L stainless steel.

Figure 5.11: Specific heat capacity of 316L stainless steel.
5.2.4 Element growth

The proposed thermal simulation is based on the process plan which is the composition of manufacturing primitives (i.e., laser scan) and the computing domain is composed of a sequence of \( n \) elements. An element is modeled by sweeping of the cross-section shape along the corresponding sub-path. In this case, formulating and solving the associated heat-transfer equation is difficult because of the co-dependent relationship: the heat associated with an element influences the heat transfer, while the cross-section shape of an element is influenced by the thermal history. This co-dependency exists due to the nature of our simulation, which tries to approximate the process based on the tool path level instead of simulating the whole powder bed.

Recall that, each element is defined as newly melted/sintered material by laser scan along the corresponding sub-path. An element growth mechanism is introduced in the simulation to mimic the actual melting process. Here element growth refers specifically to the width of the element. Each element is initialized with an initial width \( (W_0) \) that typically is determined by empirical data, high-fidelity simulations, or data-driven models. In Chapter 6, a data-driven model is built which can predict melt pool width based on laser power, scan speed, and scan path. The trained surrogate model could be integrated into our simulation to guide element width initialization. For example, the initial element width can be set to laser spot size, which relates to where the material will absorb the most laser energy. Since there is no element shrinking mechanism in the simulation, the initial element width is intended to be less than the final width – again corresponding to physical reality. In each time step during the simulation, element width will grow according to its thermal conditions via iterative correction procedure. For a single laser scan, the final element width approximates the width of the melt pool as it passes completely through an element.

To describe the element growth mechanism, we address two questions. First, at what time in the simulation should an element grow its width? An element grows when it absorbs heat and a phase change begins; so, it is reasonable to trigger the
width growth when an element’s temperature is higher than some preset threshold. Since the apparent heat capacity method is used in the simulation, the threshold is set to be $T_m + \frac{\Delta T}{2}$ ($T_m$ is melting temperature), at which the element is fully melted. In the simulation, if the newly computed temperature of an element is greater than $T_m + \frac{\Delta T}{2}$, the element may expand its width during that time step.

Second, when the element does grow, how much should it grow? The simulation calculates the growth quantity based on energy conservation. An Element can grow the width asymmetrically to left or right independently. And each element keeps track of its left/right width and free surface. $A_{left side}^i$ and $A_{right side}^i$ denotes the left and right free surface of $i$-th element respectively. The energy transferred from the melted element to the powder bed in this time step is used to compute the width growth value $W_{left grow}^i$ and $W_{right grow}^i$. Since the formula for left growth and right growth are similar, we concentrate on the formula for left growth. The energy conservation equation for left growth is written as:

$$A_{left side}^i h_p c(T_i(t) - T_p) \cdot \Delta t = A_{left side}^i W_{left grow}^i \int_{T_p}^{T_i(t+\Delta t)} (\rho(T)c(T))dT$$

where $\rho$ and $c$ are the density and specific heat capacity of the material respectively. Both $\rho$ and $c$ depend on temperature. $h_p = \frac{\lambda_{powder}}{\mu D}$. $T$ is the element temperature. $T_p$ is the temperature of the powder bed. The LHS of the above equation represents the energy transferred from the left side of the element to the powder bed in this time step. The RHS represents the energy needed to heat up the newly grown material from $T_p$ to $T_i(t + \Delta t)$.

The formula for left width growth of element $i$ is

$$W_{left grow}^i = \frac{h_p(T_i(t) - T_p) \cdot \Delta t}{\int_{T_p}^{T_i(t+\Delta t)} (\rho(T)c(T))dT}$$ (5.22)

The integration on the denominator could be estimated using numerical integration methods. Since the denominator is only a function of $T_i$, it can be pre-computed in the pre-processing stage. Equation 5.22 assumes the element can grow freely
into the powder bed on its left side. This is true for a single path laser scan. For
general cases, the element’s growth may be restricted by other elements. In our
simulation, each element keeps track of its closest neighboring elements (within
the same layer) on both sides. So each element knows the maximum value it can
grow on its left/right side (i.e., \( \text{gap}_i^\text{left}/\text{gap}_i^\text{right} \)). If \( W_i^\text{grow} < \text{gap}_i^\text{left} \), update width of element \( i \) from \( W_i \) to \( W_i + W_i^\text{grow} \) and subtract \( W_i^\text{grow} \) from \( \text{gap}_i^\text{left} \). Otherwise update width of element \( i \) from \( W_i \) to \( W_i + \text{gap}_i^\text{left} \) and update the contact information of element \( i \) and its neighbors.

5.2.5 Active body

The PBF process is driven by a moving heat source (i.e., a laser/electron beam).
At a given time, the position and power of the laser beam are determined by the
process plan. An important observation is that the majority of the elements are
not affected by the moving laser beam. An element that is far away from the heat
source has reached thermal equilibrium with its surroundings. So updating the
temperature of all the elements in every time step is neither efficient nor necessary.
The concept of active body is introduced to localize computation. In each time
step, we only update temperature of elements in the active body. In our simulation,
a binning algorithm is applied to partition the domain into bins (see Figure 5.4)
and the bin where the latest laser beam locates, is used to indicate the heat source.
Active body covers all the elements that are close to the heat source temporally or
spatially. Active body (\( A^t \subset V \)) at time \( t \) contains temporal locality part \( A^t_{\text{temp}} \) and
spatial locality part \( A^t_{\text{spat}} \) and \( A^t = A^t_{\text{temp}} \cup A^t_{\text{spat}} \). Suppose the current printing
layer number is \( z^t \). We define \( S^t \) to be the set of bins (on layer \( z^t \)) that the laser
beam visits in the last \( p \) seconds immediately preceding time \( t \), i.e., during the time
interval \([t - p, t]\). The temporal locality part \( A^t_{\text{temp}} \) is defined as

\[
A^t_{\text{temp}} = \bigcup_{(x,y,z) \in S^t} B(x,y,z) \tag{5.23}
\]

where \( B(x,y,z) \) denotes the set of elements in bin \((x,y,z)\). Given two parameters
radius $r$ and depth $d$ ($d \leq r$), which are both non-negative integers, we can define the neighborhood of the bin $(x_0, y_0, z_0)$ to be a set of bins using L-1 norm on the grid of bins.

$$N(x_0, y_0, z_0) = \{(x, y, z) | \| (x - x_0, y - y_0, z - z_0)^T \|_1 \leq r, \ max(1, z_0 - d) \leq z \leq z_0\}$$

The spatial locality part $A_{spat}^t$ is defined as

$$A_{spat}^t = \{k \mid k \in B(x, y, z), (x, y, z) \in \bigcup_{(x_0, y_0, z_0) \in S^t} N(x_0, y_0, z_0)\} \quad (5.24)$$

From the above formula, we can see $A_{temp}^t \subset A_{spat}^t$, so $A^t = A_{temp}^t \cup A_{spat}^t = A_{spat}^t$. Radius $r$ and parameter $p$ control the horizontal dimension of the active body, depth $d$ controls the vertical dimension of the active body. It is obvious that the number of elements in the active body is finite, and the number is not related to the size of the problem (i.e., total number of elements). For any given discretization and fixed parameter values ($p$, $r$ and $d$), the number of elements in the active body is bounded by a constant at any given time.

An active body example is shown in Figure 5.12. In the figure, elements not in the active body are colored in blue. Elements in temporal locality part are colored in green. Elements in spatial locality part (and not in temporal locality part) are colored in red. From it, we can see active body only includes a small subset of elements.

The size of active body is determined by the parameters $p$, $r$ and $d$. Their values are selected based on physical observations and the desired tradeoff between simulation accuracy and efficiency. To accommodate different scan patterns (e.g., checkerboard pattern), $S^t$ is designed to cover all the bins visited by the laser scan in the last $p$ seconds instead of only the current bin. The number of bins in $S^t$ is very small (1 or 2). The value of $r$ and $d$ should be big enough such that, if the laser beam follows a continuous path, it is enough to cover only the current bin in $S^t$. So
the value of $p$ should be roughly the time for the laser beam to scan through a bin. The value of $r$ is related to laser’s range of influence. The value of $d$ is related to the penetration depth. The proper choice of $p$, $r$, $d$ could be determined by experiments or simulations. In the implementation, a one item cache is used to further improve the computational efficiency.

5.2.6 Simulation

In each time step, forward Euler (FE) time marching scheme is used to update temperature of elements in active body.

$$T_i(t + \Delta t) = T_i(t) + \Delta t \cdot \dot{T}_i(t) \quad (5.25)$$

$\dot{T}_i(t)$ is computed using Equation $5.2$, $5.5$, $5.6$, $5.7$, $5.8$, $5.9$, $5.10$, $5.20$.

In our simulation, the time step is chosen such that the simulation remained stable and the results did not change when the time step was further reduced. And a time step $\Delta t = 1 \mu s$ is used to solve the transient heat transfer problem.
The full algorithm of the simulation is summarized in Algorithm 3. By localizing computation inside the active body which contains at most constant number of elements, the total execution achieves linear time complexity \( O(m) = O(n) \) and hence is scalable. This in turn implies that the constructed simulation not only scales to handle three-dimensional (3D) printed components of arbitrary complexity but also can achieve real-time performance.

**Algorithm 3: Algorithm of PBF simulation**

<table>
<thead>
<tr>
<th>input</th>
<th>scan path, material properties and manufacturing parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>temperature history of each element</td>
</tr>
<tr>
<td></td>
<td>discretize laser scan path into elements;</td>
</tr>
<tr>
<td></td>
<td>initialize contact graph;</td>
</tr>
<tr>
<td>foreach</td>
<td>time step do</td>
</tr>
<tr>
<td></td>
<td>if a new layer starts then</td>
</tr>
<tr>
<td></td>
<td>activate all the elements on the layer;</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>update active body;</td>
</tr>
<tr>
<td>foreach</td>
<td>element in active body do</td>
</tr>
<tr>
<td></td>
<td>calculate heat transfer to it;</td>
</tr>
<tr>
<td></td>
<td>calculate its new temperature;</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td>foreach</td>
<td>element in active body do</td>
</tr>
<tr>
<td></td>
<td>update its temperature;</td>
</tr>
<tr>
<td></td>
<td>update its left/right width and contact information when needed;</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
</tbody>
</table>

### 5.3 Numerical experiments

In order to validate and demonstrate the efficacy of the proposed simulation, four numerical simulations are conducted. Firstly, the fact that the element’s final width approximates melt pool width on a single path laser scan is used to validate the element growth mechanism. Secondly, the active body design is validated through
studying the influence of active body on computational efficiency and accuracy. Then the proposed simulation is further validated through replicating the “residual heat effect” observed by the experiments in National Institute of Standards and Technology (NIST). Finally, the proposed simulation is applied to simulate the manufacturing process of a realistic part.

5.3.1 Single path

In order to validate the element width growth mechanism as well as the simulation, two numerical experiments using different materials are conducted. In the first experiment, the melt pool width predictions from our model were compared against measured melt pool width in experiments [36] and numerical results from Zohdi et al. [25]. These comparisons were made for a single pass of a Gaussian laser over a single layer of powder particles resting on a substrate. In the second experiment, we compared our results from simulating a single straight laser scan of PBF process against experimental and high-fidelity numerical results published by NIST [23, 57].

In the first experiment, we applied the same experimental settings (manufacturing parameters, material properties, domain size, etc.) as those described in [36]. Khairallah and Anderson observed an experimental melt pool width of 75 µm (Figure 5.13), while their simulation predicted a melt pool width of 72 µm. The melt pool width obtained by our mesoscale simulation is 78 µm. For comparison, the coupled DE-FD approach described in [25] predicts a melt pool width 85 µm. In conclusion, melt pool prediction from our simulation agrees well with the experiment and simulation results in the literature.

In support of our claim of reduced computational costs, our proposed mesoscale simulation only required less than 5 CPU seconds to run. The reference computer system configuration includes a 3.7 GHz 6-core Intel Core processor with 16 gigabytes of RAM, running C++11 atop Ubuntu 18.04.3 LTS. In comparison, the coupled DE-FD framework takes several CPU hours to run and the ALE framework used in the simulations from [36] requires on the order of 100,000 CPU hours to run. Thus, the proposed mesoscale simulation has the capability to be used as a
quick tool to optimize process parameters.

Figure 5.13: Experimental micrograph conducted by Khairallah and Anderson [36].

In the second experiment, we used the proposed simulation to predict the melt pool width of a single straight laser scan of the PBF process using nickel alloy 625. The dimensions of the domain is $L_x = 4 \text{ mm} \times L_y = 4 \text{ mm}$. The laser starts with its center point on one side of the domain and ends once the laser’s center has moved to the other end of the domain. The laser beam is assumed to be orthogonal to the plane of the powder bed. To avoid boundary condition (BC) effects, the laser is centered on the plane $y = \frac{L_y}{2} = 2 \text{ mm}$ and the melt pool size is taken in the middle of the domain at point $(\frac{L_x}{2}, \frac{L_y}{2})$. The predicted melt pool width was compared against experimental results as well as the numerical results obtained from high fidelity simulations conducted by NIST. The comparison is listed in the table below (Table 2). In each record, the first column shows NIST’s high fidelity result, the second column contains NIST’s experimental data [57] (NA means not available), and our simulation result is in the last column. From the table, we can see there are three levels of scan speed and two levels of laser power. In order to measure the accuracy of the proposed simulation, the mean absolute percentage error (MAPE) is computed. The MAPE of the simulation results with respect to NIST’s experimental data is 6.5%. The MAPE of the simulation results with respect to NIST’s high fidelity result is 4.1%. Therefore the results obtained from the
proposed simulation agree well with the results from NIST.

In order to demonstrate how element width growth works in the simulation, we plot temperature evolution and width evolution of the element together (see Fig. 5.14). The laser power is 195 W and the scan speed is 0.8 m/s. The initial element width $W_0 = 75 \, \mu m$. Both temperature and width data are normalized to the range $[0, 1]$. Temperature evolution is denoted by the blue line, width evolution is denoted by the red line. The horizontal red dashed line represents the melting temperature. It can be seen from the figure that the slope of the temperature curve decreases around melting temperature due to latent heat effect. Also, element width grows gradually from $W_0$ to its final value during the simulation, which agrees with the physical reality.

The proposed simulation is very efficient. It takes a couple of seconds to run all the 6 simulations to generate the Table 5.2. Also, further parallelization on multiple cores would further decrease the computing time. For example, through parallelizing the temperature updates in each time step of the simulation using OpenMP, we observed 3-4 times speed up.

The comparisons discussed in this section shows that the element growth mechanism, which is the heart of our simulation, can capture the element’s width very well. We believe that these favorable comparisons validate our mesoscale, thermal-simulation for a single path.

Table 5.2: Comparison of melt pool width results expanded from [57].

<table>
<thead>
<tr>
<th></th>
<th>195 W</th>
<th>122 W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2 m/s</td>
<td>248/227/238 (\mu m)</td>
<td>184/158/177 (\mu m)</td>
</tr>
<tr>
<td>0.5 m/s</td>
<td>160/150/164 (\mu m)</td>
<td>135/127/131 (\mu m)</td>
</tr>
<tr>
<td>0.8 m/s</td>
<td>127/132/136 (\mu m)</td>
<td>113/NA/105 (\mu m)</td>
</tr>
</tbody>
</table>

**5.3.2 Influence of active body**

In order to study active body’s influence on computational efficiency, we compared the running time of a simulation with active body and a simulation without active
body (updating temperature of all the elements in each time step). Both simulations are applied to simulate the manufacturing process of parts with different sizes (the scan path of the first 6 parts are plotted in Figure 5.15). The number of elements ranges from 2,226 to 116,814. The configuration of the reference computer system is the same as that in the last section. Figure 5.16 compares the running time of both simulations (Figure 5.17 plots the running time results in loglog scale). It can be easily seen that, without active body, the running time grows quadratically with regard to the number of elements as opposed to a linear time $O(n)$ performance when using active body. Even with active body, the running time of this simulation is longer than the actual manufacturing time due to the high scan speed. Linear time complexity implies that real-time performance can be achieved by using a suitable constant time speedup, e.g., using a high-performance platform, specialized hardware, or parallelization. For example, through parallelizing the temperature
updates in each time step with OpenMP, we observed a 3-4 times speed up.

![Tool path of parts with different size.](image)

Next, the influence of an active body on accuracy is studied. The simulation without active body is used as the reference. Ideally, a simulation with active body will obtain nearly the same result as that without active body. The proposed simulation is applied to simulate a four-layer part (see Figure 5.18), which has 5,276 elements. The active body parameters are chosen as \( r = 5 \) and \( d = 4 \). An element in the first layer (denoted by the red dot in Figure 5.18) is picked to study the temperature evolution. The comparison of the computed temperature evolution is shown in Figure 5.19.

It can be seen from Figure 5.19 that the blue dashed line (simulation with active body) agrees well with the red line (simulation without active body). Simulation with active body can capture the heating and cooling cycles very well. The running time of simulation without active body is 98.25 s, and the running time of simulation with active body is 15.32 s. We therefore observe that using active body can improve computational efficiency dramatically without sacrificing too much accuracy.
In order to demonstrate the simulation results, we applied the simulation to another part which has 4 layers. This part is discretized into 19,356 elements. The temperature fields computed by the thermal simulation with active body at different times are shown in Figure 5.20, where elements are displayed as line segments with different colors according to their respective temperatures. This figure illustrates the ability of our method to run localized simulations while creating and storing a history of previous layers. Note that previous layers have been activated around the area of the active path on the most recent layer, demonstrating how contact graph and active body work together to realize a scalable, multi-layer simulation.
Figure 5.17: Influence of active body on simulation time of PBF process (loglog scale).

Figure 5.18: Element position.
Figure 5.19: Influence of active body on accuracy

Figure 5.20: Snapshots of thermal simulation of a part with four layers.
5.3.3 Further validation

In order to further validate the simulation, we attempted to use the simulation to replicate the behavior observed in experiments. The experiment is conducted on the additive manufacturing metrology testbed (AMMT), which is a fully custom, open-platform laser powder bed fusion (L-PBF) system, at National Institute of Standards and Technology (NIST) [46]. AMMT can import G-code and transform it into a sequence of position and laser power pairs, which are not fully available from any commercial AM machines. The schematic of the AMMT system is shown in Figure 5.21. In order to monitor the process, two cameras are installed, including a high resolution camera that captures layer-wise images of the entire part and a high speed melt pool monitoring camera. The Galvo mirror system and the beam splitter allow the high-speed camera to focus on the current laser melting spot. The melt pool was monitored by the high-speed camera which is optically aligned with the laser beam, such that the image of the melt pool is maintained stationary within the camera’s field of view. The sampling rate of the camera is 2,000 Hz (i.e., 2,000 frameworks per second), which means the time step between two adjacent data points is 500 µs. The raw melt pool image is in grayscale and the pixel value ranges from 0 (black) to 255 (white). Figure 5.22 shows two examples of raw melt pool image. A threshold, which is used to distinguish pixels inside and outside of the melt pool, and real pixel size are determined by experiments. Hence the experimental melt pool area could be obtained.

An in-house developed AM software Simple AM (SAM), which is capable of slicing, scan path planning, AM G-code generation, and interpretation [93], was used to program the different scan strategies used in the experiment. Both the metal powder and the build plate are Inconel 625. The experimental data were collected from a single layer printed using raster scan strategy. The scan path is plotted in Figure 5.2. In it, the scan path with laser power on is plotted in red, and the scan path with laser power off is plotted in blue. The raster path overshot out of the interior rectangle to keep laser speed constant (1 m/s) when laser power is on. And constant laser power (195 W) is used. The interior rectangle is filled from the
bottom right corner to the top left corner. The normalized measured melt pool field is plotted in Figure 5.23. Even though constant laser power and laser speed are used, the melt pool size is not constant due to different residual heat effect caused by scan path. From Figure 5.23, we can see the overheating phenomenon (oversize melt pool) happens every other pass (scan).

Since our simulation is based on the scan path level, it is difficult to retrieve the melt pool width from it (single scan path is a special case in which element’s final width approximates melt pool width). Melt pool length is approximated by summing the length of continuous elements, which have temperature larger than the melting temperature. So we use computed melt pool lengths to study the variation of melt pool size. This corresponds to physical reality. From Figure 5.22, we can see that long length is the main reason for oversize melt pools. Our simulation is applied to simulate the same process plan and the computed normalized melt pool length field is plotted in Figure 5.24. From it, we can see the resultant melt pool length field shows similar behavior as the experiments - that the overheating phenomenon happens every other pass.

Figure 5.21: Schematic of AMMT system.
Figure 5.22: Raw melt pool images.

Figure 5.23: Experimental melt pool field.
Figure 5.24: Simulation melt pool field.
5.3.4 Realistic numerical simulation

The 3D model for a realistic part (Stanford Bunny) is shown in Figure 5.25. Figure 5.26 plots the scan path model of the part, which is sliced into 166 layers. In order to make it clear, sub-paths are plotted in randomly picked colors. In the simulation, the process plan is discretized into 237,817 elements. Complete simulation of 8,486,379 time steps in the printing process took 9008.6 $s$ to execute. The temperature fields computed by the thermal simulation with active body at different times are shown in Figure 5.27, where elements are displayed as line segments with different colors according to their respective temperatures.

Figure 5.25: 3D model of Stanford Bunny [83].
Figure 5.26: Scan path model of the Stanford Bunny part.

Figure 5.27: Snapshots of thermal simulation of Stanford bunny.
5.4 Discussion

In this chapter, we applied the proposed general framework to build a scalable thermal simulation of PBF process. The simulation is performed directly on the as-manufactured model, derived from the manufacturing process plan. In the pre-processing stage, the laser scan path is discretized into many sub-paths and an element is defined as the newly sintered/melt material by laser scan along the corresponding sub-path. The contact graph, which captures the thermal interactions between elements is initialized in pre-processing and updated during the simulation. An element growth mechanism is introduced to simulate the evolution of the melt pool. A lumped-capacitance model, which covers all important thermal effects developed during the manufacturing process, including laser heating, heat convection and radiation to the ambient environment, heat conduction with build platform, along the laser scan, between adjacent elements and with powder bed, is built to conduct the transient thermal simulation. A bin-based active body design is proposed and implemented to localize the computation, which covers elements close to the moving laser beam spatially and temporally. The simulation has been fully implemented and tested on both single laser scan and 3D parts. As we showed experimentally, the simulation achieves linear time complexity with the help of active body, which implies that the simulation scales to handle 3D printed components of arbitrary complexity. Simulation of single laser path is fully validated by comparing the simulation results with both experimental data and finer-scale simulation results.

The computed thermal history by the simulation could be used to predict part properties and detect manufacturing failures (e.g., underheat and overheat). We also note that the ability to efficiently simulate the thermal field evolution of the PBF process opens many possibilities for solving the inverse problem. The inverse problem includes optimizing manufacturing parameters for a given geometry and building feedback-control mechanisms to guide against failures during the manufacturing process.
6 DATA-DRIVEN MODEL - MELT POOL PREDICTION AND CONTROL

6.1 Introduction

In the last chapter, we applied the proposed general framework to build a scalable thermal simulation of PBF process on scan path level. One salient feature of the simulation is the size of an element is influenced by the thermal history. In the pre-processing stage, every element is initialized with an initial width. An element growth mechanism is introduced to update the element’s width in an iterative correction fashion during the simulation. In this chapter, we apply the CAPL discretization and the idea of locality to build a data-driven melt pool prediction model using experimental data. One application of this model is to guide the element width initialization in our mesoscale thermal simulation of PBF process.

As PBF process is driven by a moving heat source, the material undergoes three sub-processes to complete the powder-bulk material transformation: laser heat absorption, melt pool formation, and solidification [85]. The as-manufactured geometry is formed by the sweeping of a melt pool, where the microstructure and mechanical properties of the printed parts are significantly influenced by the shape and size of the melt pool [50]. Therefore, the formation and moving of the melt pool play an important role in determining the geometric and mechanical properties of the printed components. Studied defects such as pores and keyholes have been observed to be closely related to abnormal melt pools [15, 79, 35, 65]. Towards improved quality control and optimized manufacturing parameters, it is critical to have a quantitative understanding of the relationship between process inputs and the size of the melt pool.

It is widely accepted that both laser power and scan speed have a significant impact on the size of the melt pool [85, 52]. Laser power has a positive correlation with melt pool size and scan speed has a negative correlation with melt pool size [53, 80, 51]. The formation of a melt pool can also be affected by environmental
conditions such as the preheating temperature (when applicable) [101]. Many models to predict melt pool behavior have been developed based on different assumptions and purposes. These models can be divided into two categories: physics-based simulations and data-driven models.

Chapter 2.2 reviewed past work on physical-based simulations of PBF process. And, none of the physical-based simulations are ideal for predicting melt pool size for realistically complex process plans across various scales.

In comparison, data-driven models build relationships between process inputs (i.e., manufacturing parameters and scan path) and melt pool characteristics directly from data obtained from experiments or simulations [53, 89]. Recently, many researchers [58, 34, 47, 10, 91, 66, 90] have studied data-driven models in the additive manufacturing community. These models are most often developed independently of the physics. Thus, the assumptions and idealizations offered by data-driven models allow them to be much more efficient to evaluate than simulations. These assumptions and idealizations can sometimes lead to the oversimplification of complex problems. For instance, existing data-driven melt pool prediction models usually assume over-simplified process plans (e.g., constant laser power, scan speed, simple scan patterns). Such simplifications may not be suitable for realistic process plans with complicated geometry.

We formulate the melt pool prediction problem as a supervised learning problem. A multiple regression model, which takes process parameters and the scan path as input and melt pool size (i.e., length and width) as output, is built with a neural network and trained using experimental data from NIST. Our CAPL discretization and the idea of locality is used to encode the scan path information into two novel neighborhood features: a temporal neighborhood feature and a spatial neighborhood feature. The trained surrogate model achieves 8.3% mean absolute percentage error (MAPE) on the test set. The performance of the surrogate model is boosted further by ensemble methods (e.g., bagging, boosting, and stacking). The surrogate model is interpreted by studying the influence of individual features on the melt pool size. Two applications of the surrogate model are introduced. A laser power design algorithm, designed to keep the melt pool as constant as possible, is
proposed and the trained surrogate model is introduced into the algorithm as the forward solver. The resulting laser power plan is designed to keep melt pool size as constant as possible for any given scan pattern. Also, the surrogate model could be integrated into our mesoscale thermal simulation of PBF process to guide the element width initialization.

The rest of this chapter is organized as follows. The formulation of the neighborhood-based neural network is described in Chapter 6.2. In it, the experiment set up and data are introduced in Chapter 6.2.1. Data exploration is described in Chapter 6.2.2. Chapter 6.2.3 describe the process of feature engineering and the structure of the multiple-regression model. Chapter 6.2.5 applied different ensemble methods to improve the performance of the trained surrogate model. Chapter 6.2.6 interprets the surrogate model. Chapter 6.3 describes the two applications of the surrogate model.

6.2 Data driven model

6.2.1 Experimental Data

The experiment is conducted on the additive manufacturing metrology testbed (AMMT) at NIST. A detailed description of AMMT is in Chapter 5.3.3. In this chapter, the referenced experimental data was collected from a single layer printed using an “island” spiral concentrating scan strategy. Both the metal powder and the build plate are Inconel Alloy 625. A similar study was previously conducted by Yang et al., who investigated the same set of experimental data using polynomial regression in [91]. Our approach aims to improve on the earlier results by adopting a different training model with additional ensemble methods for improved performance. Also, a laser power design algorithm, which uses the trained surrogate model as a forward solver, is proposed.

The laser scan path is plotted in Figure 6.1. From the figure, we can see the layer is composed of four islands and each island is filled with a spiral path. The darker the path is, the later it is scanned (e.g., the upper right island is first scanned,
the lower left island is last scanned). The sampling rate of the camera is 2,000 Hz (i.e., 2,000 frames per second), which means the time step between two adjacent data points is 500 µs. The raw melt pool image is in grayscale and the pixel value ranges from 0 (black) to 255 (white). Figure 6.2 shows an example of melt pool image. A threshold, which is used to distinguish pixels inside and outside of the melt pool, and real pixel size are determined by experiments (i.e., each pixel stands for $8\mu m \times 8\mu m$ area). In this study, the boundary of the melt pool is assumed to be an ellipse and least square fitting is used to approximate the boundary. Figure 6.3 shows the pipeline of pre-processing melt pool images including sharpening the image, edge detection, least squares fitting with an ellipse.

![Figure 6.1: laser scan path.](image)

After pre-processing the raw images, melt pool sizes (i.e., length and width) are obtained. There are 4,700 raw data points as well as the corresponding laser position, laser power, and scan speed. Figure 6.4 lists the first ten experimental data points. Essentially, the data points discretize the scan path. Figure 6.5 plots the melt pool area distribution of the experimental data with a color map. From it,
we can see melt pools at the center of spirals tend to be larger than those on the boundaries.
Figure 6.4: Experimental data.

<table>
<thead>
<tr>
<th>time</th>
<th>x</th>
<th>y</th>
<th>power</th>
<th>speed</th>
<th>width</th>
<th>length</th>
<th>area</th>
<th>angle</th>
<th>intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>501.0</td>
<td>14.483</td>
<td>-20.201</td>
<td>0.000</td>
<td>495.54</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000</td>
<td>0.0000</td>
</tr>
<tr>
<td>1</td>
<td>1001.0</td>
<td>14.303</td>
<td>-20.368</td>
<td>0.000</td>
<td>448.45</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>1501.0</td>
<td>14.176</td>
<td>-20.387</td>
<td>38.552</td>
<td>359.01</td>
<td>0.068725</td>
<td>0.099146</td>
<td>0.005248</td>
<td>-65.318</td>
</tr>
<tr>
<td>3</td>
<td>2001.0</td>
<td>14.190</td>
<td>-20.138</td>
<td>93.542</td>
<td>634.21</td>
<td>0.080868</td>
<td>0.115790</td>
<td>0.007232</td>
<td>-59.637</td>
</tr>
<tr>
<td>4</td>
<td>2501.0</td>
<td>14.219</td>
<td>-19.718</td>
<td>105.020</td>
<td>902.47</td>
<td>0.083984</td>
<td>0.110180</td>
<td>0.007104</td>
<td>-55.568</td>
</tr>
<tr>
<td>5</td>
<td>3001.0</td>
<td>14.251</td>
<td>-19.269</td>
<td>105.020</td>
<td>902.47</td>
<td>0.076381</td>
<td>0.099066</td>
<td>0.005760</td>
<td>-40.088</td>
</tr>
<tr>
<td>6</td>
<td>3501.0</td>
<td>14.282</td>
<td>-18.820</td>
<td>105.020</td>
<td>902.47</td>
<td>0.078625</td>
<td>0.099475</td>
<td>0.005992</td>
<td>-50.743</td>
</tr>
<tr>
<td>7</td>
<td>4001.0</td>
<td>14.313</td>
<td>-18.371</td>
<td>105.020</td>
<td>900.62</td>
<td>0.086811</td>
<td>0.115970</td>
<td>0.007552</td>
<td>-36.774</td>
</tr>
<tr>
<td>8</td>
<td>4501.0</td>
<td>14.345</td>
<td>-17.922</td>
<td>105.020</td>
<td>902.47</td>
<td>0.076049</td>
<td>0.087773</td>
<td>0.005120</td>
<td>-55.342</td>
</tr>
<tr>
<td>9</td>
<td>5001.0</td>
<td>14.376</td>
<td>-17.473</td>
<td>105.020</td>
<td>902.47</td>
<td>0.077192</td>
<td>0.106130</td>
<td>0.006336</td>
<td>-23.780</td>
</tr>
</tbody>
</table>

Figure 6.5: Melt pool area distribution.
6.2.2 Data exploration

Before building the data-driven model, we explored the experimental data. Figure 6.6 plots the relationship between the laser-power/scan-speed/energy-density and the melt pool length/width/area. From it, we can see the distribution of the variables and the relation between them. Both the laser power and the energy density have a positive correlation with the melt pool size. The relationship between the scan speed and the melt pool size is not clearly shown by the data.

Figure 6.6: Pairplot of laser-power/scan-speed/energy-density and melt pool length/width/area.

6.2.3 Feature engineering and neural network

In order to build a data-driven model, we formulate the problem as a supervised learning problem (see figure 6.7). Both the process input (including laser power, scan speed, and laser scan path) and measured melt pool sizes are the input of the
algorithm, the output of the algorithm is a surrogate model. The feature vector \((X)\) contains information from the process plan, which includes laser power, scan speed, scan path, etc. The output feature \((Y)\) contains the measured melt pool size (i.e., melt pool length and width). The mapping from \(X\) to \(Y\) is denoted by \(f\) and \(Y = f(X)\). As the real \(f\) is unknown, various computational models (e.g., simulations) are developed to approximate it. Our goal is to train a hypothesis \(f'\) so that it will accurately approximate the mapping \(f\) by developing a neural network.

![Image](image.png)

Figure 6.7: The paradigm of supervised learning.

The PBF process is driven by a moving heat source. Heat transfer influences the formation and evolution of a melt pool, so both laser power and scan speed are important features. They are often primary considerations incorporated into any data-driven melt pool prediction models. However, using only laser power and scan speed is not enough. A melt pool is also affected by the scan path, and our approach looks to uniquely incorporated the effects of the scan path into the prediction of melt pool. For a given scan path, constant laser power and constant scan speed do not necessarily lead to a constant melt pool along the path. Variations will come from so-called neighboring effects that are related to the scan path. For example, melt pools tend to be larger at the center of a spiral path due to the high residual heat of these neighborhoods. The size of the melt pool can be predicted based
on manufacturing parameters and the local information of the scan path, which means that the melt pool at a position \((C_0)\) on the path is influenced locally by the laser scan on its neighborhood. The experimental data points can be interpreted as a discretization of the scan path. In order to quantify the neighborhood effect of a position on the scan path, we introduce two novel neighborhood features: a temporal neighborhood feature and a spatial neighborhood feature. The temporal neighborhood feature of the data point at \(C_0\) addresses the influence of laser scans that are close to \(C_0\) in time. The spatial neighborhood feature of the data point at \(C_0\) addresses the influence of laser scans that are close to \(C_0\) in space. In order to compute the neighborhood features of the data point at \(C_0\), we need to know which data points (scanned earlier) that have thermal influence on it spatially or/and temporally. The idea of CAPL (described in Chapter 3.1) is applied. An auxiliary data structure, contact graph \((G = (V, E))\), is constructed. The contact graph is comprised of a set of \(V\) of vertices, where each vertex represents a data point, together with a set \(E\) of edges corresponding to contacts between adjacent data points. Since the energy of the laser scan is highly concentrated, the thermal impact area of a data point is enclosed in a circle centered at \(C_0\). The radius of the circle is a predefined parameter \(r_0\), which could be determined by empirical data or simulations. Two data points \(i\) and \(j\) contact if the Euclidean distance between them is less than or equal to \(2r_0\) (i.e., \(d_{i,j} \leq 2r_0\)). Contact detection is applied to build the contact graph. The data points are stored in an array sequentially according to the process plan and each data point keeps track of the indexes of its contacting points.

The formulas of both neighborhood features are described below. The neighborhood effect of the \(i\)-th data point is only related to data points that have been previously scanned. Assuming the indexes of data points are ordered from 1 to \(n\) by their scanned time and the \(i\)-th data point is scanned at \(t_i\), the temporal neighborhood feature \((X_{i,\text{temp}})\) of the \(i\)-th data point is expressed as

\[
X_{i,\text{temp}} = \sum_{j \in g_{i,\text{temp}}} \frac{p_j}{v_j d_{i,j}},
\]
where $S^\text{temp}_i$ is the temporal locality set including all the data points ($j < i$ and $d_{i,j} \neq 0$) scanned during time interval $[t_i - p, t_i]$, $p$ is a parameter that controls the size of $S^\text{temp}_i$. $P_j$ is the laser power of the $j$-th data point, $v_j$ is the scan speed of the $j$-th data point, $d_{i,j}$ is the euclidean distance of the $i$-th data point and the $j$-th data point (i.e., $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, $d_{i,j} > 0$). It should be noted that data points with zero scan speed ($v = 0$) are removed when cleaning the data. The temporal neighborhood feature can be interpreted as the weighted summation of energy densities of the temporal neighborhood. The inversion of euclidean distance is used as the weight. The closer in distance the data point is, the more it contributes.

The spatial neighborhood feature ($X^\text{spat}_i$) of the $i$-th data point is expressed as

$$X^\text{spat}_i = \sum_{j \in S^\text{spat}_i} \frac{P_j}{v_j(i - j)}, \quad (6.2)$$

where $S^\text{spat}_i$ is the spatial locality set including all data points ($j < i$) that contact with $i$-th data point in contact graph. The spatial neighborhood feature can be interpreted as the weighted summation of energy densities of the spatial neighborhood. Since the index is ordered by scanned time, index differences indicate the scanned time differences. The inversion of the index difference is used as the weight. The closer in time the data point is, the more it contributes.

Essentially, the temporal neighborhood feature and the spatial neighborhood feature work together to encode the laser scan path information. The $Y$ variable includes both melt pool length and width. A multiple regression model is built with a neural network. The structure of the neural network is plotted in Figure 6.8. A feedforward neural network, which contains one hidden layer with 10 nodes is used. The rectified activation function is used between the input layer and the hidden layer.
6.2.4 Train and test

Before the model is trained, the 4,700 data points are cleaned (i.e., the outliers such as zero melt pool length/width are detected and removed). There are 4,676 data points left. The data set is randomly shuffled and split into a training set and a test set. The training set contains 3,600 data points and the testing set contains 1,076 data points. Adam [39] is used to train the neural network.

The mean absolute percentage error (MAPE) achieved by the surrogate model on the test set is 8.3%. The MAPE of the melt pool length prediction on the test set is 9.3%. The MAPE of the melt pool width prediction on the test set is 7.2%.

To demonstrate the performance of the surrogate model, we plot the predicted melt pool of the testing set and the ground truth melt pool size (from experimental data) together in Figure 6.9. In the figure, each melt pool is plotted as an ellipse (i.e., long axis = melt pool length, short axis = melt pool width). The predicted melt
pool is plotted in green, the experimental data are plotted in red. From the figure, we can see the predicted melt pools agree very well with the experimental data. To investigate the sensitivity of our neural network, we changed the structure of the neural network (i.e., use deeper and wider neural networks) and observed little performance difference. Due to the relatively small size of the training set and the relatively simple structure of the neural network, it takes only a couple of seconds to train the model. Additionally, evaluating the resulting surrogate model takes nearly no time, showing the data-driven model to be much more computationally efficient than traditional simulations.

### 6.2.5 Ensemble methods

In this section, different ensemble methods are applied to improve the performance of the surrogate model. Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking) [19]. An ensemble model combines the decisions from multiple models to improve the overall performance. We focus on the melt pool width prediction. The most direct way of manipulating the training set is called bagging. On each run, bagging presents the learning algorithm with a training set that consists of a sample of \( n \) training examples drawn randomly with replacement from the original training set of \( n \) items. We trained 10 base models. The base models are combined to one ensemble model by averaging their predictions. Figure 6.10 shows the performance of the bagging ensemble with different numbers of base models. The x-axis indicates the number of base models used. From left to right, the \( i \)-th (1 ≤ \( i \) ≤ 10) blue point indicates the performance of \( i \)-th base model, the \( i \)-th red point indicates the performance of ensemble model, which is built from the first \( i \) base models. From it, we can see the MAPE of the ensemble model on the test set is usually less than an individual base model. The best ensemble model is the last one, which is combined with all the 10 base models. This model achieves 8.06% MAPE.

For boosting, we implemented the Adaboost.R2 algorithm [21] to improve the
performance of the surrogate model. Ten base models are trained sequentially using weighted training data. The ensemble model achieves 8.15% MAPE on the test set.

Stacking, also called stacked regression, is a class of algorithms that involves training a second-level “meta-learner” to find the optimal combination of the base learners. Unlike bagging and boosting, the goal in stacking is to ensemble strong, diverse sets of learners together. Here, 10 base learners are trained. A neural network is used to train the ensemble model using the predictions of base learners
as training data. Figure 6.11 plots the performance of stacking. From it, we can see that by using stacking the ensemble model can achieve 8.16% MAPE on the testing set. We have thus demonstrated that ensemble methods can improve the performance of the surrogate model and that developing the neighborhood-based neural network was only the first step to developing an algorithm that can effectively predict melt pool behavior.

### 6.2.6 Model interpretation

The trained surrogate model can be seen as the collected knowledge from all observations (i.e., experimental data). In order to interpret the surrogate model to gain some insight, the influence of the laser-power/scan-speed on the melt pool length/width is studied. The surrogate model is a high-dimensional function, so in order to visualize the influence of one feature on the melt pool size, the other features are fixed at their mean value. For example, when plotting the influence of laser power on melt pool width, all the other three features (i.e., scan speed,
temporal/spatial neighborhood feature) are fixed at their mean value, laser power is varied from $P_{\text{min}}$ to $P_{\text{max}}$, hence the influence can be plotted in 2D. Figure 6.12 shows the influence of laser power on melt pool length and width. From it, we can see that laser power has a positive correlation with melt pool length/width, which agrees with the physical reality. Figure 6.13 shows the influence of laser speed on melt pool length and width. It is clear that scan speed has a negative correlation with melt pool width. The influence of scan speed on melt pool length is more complicated. Within the vast majority of scan speed ranges, scan speed has a negative correlation with melt pool length. In the range $[500 \text{ mm/s}, 800 \text{ mm/s}]$, the scan speed has a positive correlation with melt pool length, which is not quite reasonable. There are two possible reasons. One reason is that higher scan speed leads to more apparent “long tail” effects, which usually cause longer melt pool measurement. The other reason is there are not enough data points expanding this part of the design space.

More importantly, the trained surrogate model provides a quantitative mapping
from manufacturing parameters to the melt pool size. As it is very efficient to evaluate the surrogate model, many opportunities are opened to solve the inverse problem. A novel laser power design algorithm, which uses the above two features of the surrogate model, is proposed in the next section.

![Figure 6.12: Influence of laser power on melt pool length/width.](image)
Figure 6.13: Influence of scan speed on melt pool length/width.
6.3 Application

In this section, two applications of the data-driven melt pool prediction model are introduced. In the first application, a laser power design algorithm is proposed and the trained surrogate model is fed into the algorithm as a forward solver, which can design a laser power plan to keep melt pool size as constant as possible for any given scan path. In the second application, the trained surrogate model is integrated into our mesoscale thermal simulation of PBF process to guide element width initialization.

6.3.1 Laser power design

Like we discussed in Section 6.2.3, the melt pool size is related to both manufacturing parameters (i.e., laser power and scan speed) and the laser scan path. Constant laser power and scan speed do not always lead to constant melt pool sizes due to different neighborhood effects caused by the laser scan path. The trained surrogate model takes the neighborhood effects into account by encoding the scan path information into two neighborhood features. The surrogate model provides a quantitative mapping from the manufacturing parameters to the melt pool size. And the mapping from the laser power to the melt pool size is monotonic. The idea of binary search can be applied to search for proper laser powers to achieve constant melt pool sizes. Here we again focus on the melt pool width. Given the laser scan path and scan speed plan, we want to design a laser power plan to keep the melt pool width as constant as possible. The laser power design algorithm is summarized in Algorithm 4. First, the scan path is discretized into \( n \) points. The goal is to design a laser power solver on these points. The design procedure is conducted sequentially from the first point to the last point. It should be noted that the neighborhood features of the \( i \)-th point are only related to previous points. \( P_{\min} \), \( P_{\max} \) and \( \epsilon \) are predefined parameters. Both minimum laser power \( (P_{\min}) \) and maximum laser power \( (P_{\max}) \) are nonnegative integers \( (P_{\max} \) depends on the AM machine). In the below example, we choose \( P_{\min} = 0 \) \( W \) and \( P_{\max} = 250 \) \( W \). \( \epsilon \) stands for the tolerance. There is a trade-off when choosing \( \epsilon \). We choose \( \epsilon = 2 \) \( \mu m \). \( v_i \) is the
scan speed of the i-th point, which can be obtained from the input scan speed plan. 
$f'$ represents the trained surrogate model. Any forward models (e.g., surrogate 
model, empirical formulas, simulations), which have a positive correlation between 
laser power and melt pool size, can be fed into this algorithm.

**Algorithm 4:** Laser power design algorithm.

**input**: scan path, scan speed plan, $P_{\text{low}}$, $P_{\text{high}}$ and $\epsilon$

**output**: laser powers on discrete points on scan path

discretize laser scan path into $n$ points;
compute scan speed on these points ($\{v_i|1 \leq i \leq n\}$);

for $i \leftarrow 1$ to $n$ do

compute temporal neighborhood feature $X_{i}^{\text{temp}}$;
compute spatial neighborhood feature $X_{i}^{\text{spat}}$;

$P_{\text{low}} \leftarrow P_{\text{min}}$;
$P_{\text{high}} \leftarrow P_{\text{max}}$;
while $P_{\text{low}} < P_{\text{high}}$ do

$P_{\text{mid}} \leftarrow (P_{\text{low}} + P_{\text{high}})/2$;
$w_{\text{predict}} = f'(P_{\text{mid}}, v_i, X_{i}^{\text{temp}}, X_{i}^{\text{spat}})$;
if $\text{abs}(w_{\text{predict}} - w_{\text{target}}) < \epsilon$ then

return $P_{\text{mid}}$;
else if $w_{\text{predict}} > w_{\text{target}}$ then

$P_{\text{high}} \leftarrow P_{\text{mid}} - 1$;
else

$P_{\text{low}} \leftarrow P_{\text{mid}} + 1$;
end

end

return $P_{\text{low}}$;

To demonstrate the proposed laser power design algorithm, the scan path in 
Figure 6.1 and a constant scan speed ($v = 668$ mm/s) are fed into the algorithm. 
The target melt pool width is set to be 119 $\mu$m, which is the mean melt pool width 
of experimental data. The designed laser power field is plotted in Figure 6.14. From 
it, we can see that the algorithm mitigates the overheating issue at the center of 
each spiral path by decreasing the laser power.
The surrogate model is used to evaluate the designed process plan. Figure 6.15 plots the resultant melt pools of designed laser power and constant speed. The target melt pool width is 119 µm. All the melt pool widths that lie in [117 µm, 121 µm] are plotted in blue, melt pools with a width less than 117 µm are plotted in green, and melt pools with a width greater than 121 µm are plotted in red. From the figure, we can see all the resultant melt pools are plotted in blue, which means they all lie in the target range [117 µm, 121 µm].

For comparison, the resultant melt pool width field with constant laser power
and scan speed is plotted in Figure 6.16. From it, we can see the variation of melt pool width is much larger than that from the designed laser power plan. The resultant melt pool widths lie in [105 µm, 153 µm]. Points at the center of the spiral path tend to have a large melt pool width due to overheating. And points on the boundary of each island tend to have small melt pool widths due to satisfactory cooling conditions and low residual heat. These phenomena agree well with physical reality. From the above comparison, we can see the laser power design algorithm can offset variations caused by a scan path by adjusting the laser power.

In addition to its predictive accuracy, the proposed laser power design algorithm is very efficient. Since binary search is used, each evaluation of the surrogate model can halve the current design space. It only requires 3-5 evaluations of the surrogate model to design the laser power of one point. It takes seconds to design for the
Another application of the data-driven melt pool prediction model is to integrate it into our mesoscale thermal simulation of PBF to guide the initialization of element width. There are two advantages of the data-driven model, which make it suitable for this purpose. Firstly, the data-driven model can be trained once and used later.
Secondly, it is very efficient to evaluate the data-driven model. In order to integrate it into our simulation, the weights of the trained neural network are extracted and the forward neural network is implemented as a function, which is added to the pre-processing stage of the simulation. In this way, the initial width of each element is set based on its own conditions (i.e., laser power, scan speed) extracted from the process plan. Evaluating the width of one element takes $O(1)$ time, so it does not change the time complexity of the whole simulation. Also using the data-driven model to set element width reduces the work of element growth dramatically, which improves computational efficiency. We call the integrated model as a hybrid model which combines data-driven model and physical simulation. This shows a way to incorporate knowledge from other models (e.g. micro-scale models) inside our mesoscale simulation.

For realistic process plans, some melt pool overlap between adjacent laser scans is necessary for proper bonding and a smoother surface finish, which means melt pool width is usually larger than hatching distance. In our framework, element overlap is not allowed. According to the algorithm described in Chapter 5.2.1, an element will contact an element in the adjacent laser scan, and their half-width is set at half of the hatching distance. It should be noted that the data-driven model is not a prerequisite of the proposed mesoscale simulation. Based on our experiments, the current way of using a predefined initial width and width growth mechanism has nearly the same behavior as the simulation using data-driven model to initialize element width.

\section*{6.4 Discussion}

In this chapter, a multiple regression model, which can predict melt pool length and width, is built with a neural network and trained using experimental data from NIST. The proposed model takes account of the influence of both the manufacturing parameters (i.e., laser power and scan speed) and laser scan path on melt pool size. A novel way of encoding information of the scan path into two neighborhood features through locality is provided. The trained surrogate model achieves 8.3 \%
MAPE on the testing set. Multiple ensemble methods (i.e., bagging, boosting, stacking) are implemented to improve the performance of the trained model. We also proposed a way to interpret the trained surrogate model. Finally, two applications of the surrogate model are discussed. A laser power design algorithm, which can keep melt pool size as constant as possible for any given scan path, is proposed. The surrogate model is fed into the laser power design algorithm as a forward solver. The algorithm is implemented and validated using numerical experiments. The proposed data-driven model can also be integrated into our mesoscale thermal simulation. Its ability to predict melt pool size based on a process plan is used to initialize the element width in the pre-processing stage.

In addition, the way to encode scan path information into two neighborhood factors can be applied to build other data-driven models (e.g., models predicting temperature evolution). The data-driven model can be used to provide insight on scan path design while other models, such as physical simulations, can serve as the forward solver in the laser power design algorithm. Finally, the ability to efficiently predict melt pool size opens many opportunities for solving the inverse problem of optimizing manufacturing parameters for a given geometry and building feedback-control mechanisms to guide against failures during the manufacturing process.
In this thesis, we proposed and developed a novel general framework that acts as a template to build scalable thermal simulation of any AM process driven by a moving heat source based on the as-manufactured geometry. The discretization - contact-aware path-level (CAPL) discretization, is based on the manufacturing primitives described by the tool path plan. A data structure, called contact graph, is used to represent the discretized domain and capture all possible thermal interactions during the simulation. Scalability is achieved by exploiting temporal and spatial locality, which allows updating at most constant number of elements in each time step. The framework has been successfully applied to two different AM processes, fused deposition modeling (FDM) and powder bed fusion (PBF). Also, we built and trained a neighborhood-based neural network using the experimental data, which can predict melt pool size based on manufacturing parameters and scan path of PBF process. It could also be used in our mesoscale PBF simulation and improve the performance. A laser power design algorithm, which uses the data-driven melt pool prediction model as a forward solver, is developed to keep melt pool size as constant as possible for any given scan pattern.

The proposed framework is applicable to all AM processes driven by a moving heat source. Actually, it is applicable to AM processes driven by a constant number of moving heat sources. For example, directed energy deposition (DED) fabricates parts by utilizing a moving focused energy source (e.g., laser beam) to melt the material deposited by a nozzle in the form wire or powder [26]. In order to apply the framework to DED, the element could be defined as material deposited through nozzle along the corresponding sub-path which is similar to that in FDM. Just like in FDM, the contact graph is a static data structure that is built once in the preprocessing stage. Since the heat source is identical to that of PBF, both the lumped-capacitance model and active body design could be used in DED simulation without much modification. Similarly, the framework can be applied to other AM processes. We also note that the ability to efficiently simulate the thermal field evolution of
AM processes opens up many possibilities for investigating mechanical properties of the manufactured part that are influenced by the thermal history, solving the inverse problem of optimizing manufacturing parameters for a given geometry, and building feedback-control mechanisms to guide against failures during the manufacturing process.

Our path-level simulation is designed to simulate realistic process plans. One challenging aspect of this research is that we are not aware of any other experimental measurements of realistic parts that could be used to validate the correctness of the computed results. Also, the PBF simulation has two limitations. Firstly, it does not capture the stochastic nature of the process. Secondly, it runs slower than the actual manufacturing process. Both issues should be studied in the future.

Currently, the framework is only for thermal simulation. It can be extended to other types of simulation (e.g., structural analysis). Also, the current active body design is determined by a set of constant parameters. Even though, with the current active body design, both simulations of FDM process and PBF process achieve linear time performance and linear time complexity is the best asymptotic behavior of this problem, a more sophisticated active body design may lead to constant times speed up, which can be a huge improvement in computational efficiency. For example, we can design an adaptive active body, which keeps the size of active body as small as possible, so that the number of temperature updates is minimized. Clearly, there is a trade-off, more sophisticated active body design may lead to a smaller active body, however, it usually takes more time on updating the active body. In the current work, we provide one way to utilize temporal and spatial locality to design the active body, which can make the simulation scalable. Optimizing the performance of active body design is not our main focus. Finally, the idea of localizing computation in a sub-domain is not limited to thermal simulations of moving heat source. It can be extended to any problems that satisfy the principle of locality in physics.
BIBLIOGRAPHY


[86] Yan, Wentao, Wenjun Ge, Jacob Smith, Gregory Wagner, Feng Lin, and Wing Kam Liu. 2015. Towards high-quality selective beam melting technologies: modeling and experiments of single track formations. In *26th annual international symposium on solid freeform fabrication, austin, texas*.


[105] Zhou, Yong, Timo Nyberg, Gang Xiong, and Dan Liu. 2016. Temperature analysis in the fused deposition modeling process. In 2016 3rd international conference on information science and control engineering (icisce), 678–682. IEEE.


