Topology and Toolpath Optimization via Layer-Less Multi-Axis Material Extrusion

Joseph R. Kubalak

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

Christopher B. Williams, Co-Chair
Alfred L. Wicks, Co-Chair
Robert A. Canfield
Erik E. Komendera
Robert L. West

December 17, 2020
Blacksburg, Virginia

Keywords: additive manufacturing, material extrusion, robotics, topology optimization, design methodology
Copyright 2020, Joseph R. Kubalak
Topology and Toolpath Optimization via Layer-Less Multi-Axis Material Extrusion

Joseph R. Kubalak

ABSTRACT

Although additive manufacturing technologies are often referred to as “3D printing,” the family of technologies typically deposit material on a layer-by-layer basis. For material extrusion (ME) in particular, the deposition process results in weak inter- and intra-layer bonds that reduce mechanical performance in those directions. Despite this shortcoming, ME offers the opportunity to specifically and preferentially align the reinforcement of a composite material throughout a part by customizing the toolpath. Recent developments in multi-axis deposition have demonstrated the ability to place material outside of the XY-plane, enabling depositions to align to any 3D (i.e., non-planar) vector. Although mechanical property improvements have been demonstrated, toolpath planning capabilities are limited; the geometries and load paths are restricted to surface-based structures, rather than fully 3D load paths.

By specifically planning deposition paths (roads) where the composite reinforcement is aligned to the load paths within a structure, there is an opportunity for a step-change in the mechanical properties of ME parts. To achieve this goal for arbitrary geometries and load paths, the author presents a design and process planning workflow that concurrently optimizes the topology of the part and the toolpath used to fabricate it. The workflow i) identifies the optimal structure and road directions using topology optimization (TO), ii) plans roads aligned to those optimal directions, iii) orders those roads for collision-free deposition, and iv) translates that ordered set of roads to a robot-interpretable toolpath.

A TO algorithm, capable of optimizing 3D material orientations, is presented and demonstrated in the context of 2D and 3D load cases. The algorithm achieved a 38% improvement in final solution compliance for a 3D Wheel problem relative to existing TO algorithms with planar orientation optimization considerations. Optimized geometries and their associated orientation fields were then propagated with the presented alignment-focused deposition path planner and conventional toolpath planners. The presented method resulted in a 97% correlation between the road directions and the orientation field, while the conventional methods only achieved 77%. A planar multi-load case was then fabricated using each of these methods and tested in both tension and bending; the presented alignment-focused method resulted in a 108.24% and 29.25% improvement in each load case, respectively. To evaluate the workflow in a multi-axis context, an inverted Wheel problem was optimized and processed by the workflow. The resulting toolpaths were then fabricated on a multi-axis deposition platform and mechanically evaluated relative to geometrically similar structures using a conventional toolpath planner. While the alignment in the multi-axis specimen was improved from the conventional method, the mechanical properties were reduced due to limitations of the multi-axis deposition platform.
Topology and Toolpath Optimization via Layer-Less Multi-Axis Material Extrusion

Joseph R. Kubalak

GENERAL AUDIENCE ABSTRACT

The material extrusion additive manufacturing process is widely used by hobbyists and industry professionals to produce demonstration parts, but the process is often overlooked for end-use, load bearing parts. This is due to the layer-by-layer fabrication method used to create the desired geometries; the bonding between layers is weaker than the direction material is deposited. If load paths acting on the printed structure travel across those layer interfaces, the part performance will decrease. Whereas gantry-based systems are forced into this layer-by-layer strategy, robotic arms allow the deposition head to rotate, which enables depositions to be placed outside of the XY-plane (i.e., the typical layer). If depositions are appropriately planned using this flexibility, the layer interfaces can be oriented away from the load paths such that all of the load acts on the (stronger) depositions.

Although this benefit has been demonstrated in literature, the existing methods for planning robotic toolpaths have limits on printability; certain load paths and geometries cannot be printed due to concerns that the robotic system will collide with the part being printed. This work focuses on increasing the generality of these toolpath planning methods by enabling any geometry and set of load paths to be printed. This is achieved through three objectives: i) identify the load paths within the structure, ii) plan roads aligned to those load paths, iii) order those roads such that collisions will not occur. The author presents and evaluates a design workflow that addresses each of these three objectives by simultaneously optimizing the geometry of the part as well as the toolpath used to fabricate it. Planar and 3D load cases are optimized, processed using the presented workflow, and then fabricated on a multi-axis deposition platform. The resulting specimens are then mechanically tested and compared to specimens fabricated using conventional toolpath planning methods.
Acknowledgments

I would like to thank my parents and sister for being constant sources of motivation and inspiration. Your love and support is greatly appreciated, and I could not have gotten through this without you. My fiancé, Callie Zawaski, was a tremendous support, both intellectually and emotionally, as we went through our graduate programs together.

I would like to thank my committee co-chairs, Dr. Christopher B. Williams and Dr. Alfred L. Wicks, for taking a chance on me and multi-axis deposition. I did not know anything about 3D printing before joining Dr. Williams’ senior design team (DreamVendor 2.0), but I immediately found a passion for the field. I consider myself incredibly fortunate; multi-axis deposition was not a focus area for either group when I started, and I am so thankful that I had the chance to work on this topic.

My committee members have also been immensely helpful through this process. Dr. Robert A. Canfield is acknowledged for his help with topology optimization, Dr. Erik E. Komendera for his help with robotic kinematics and path planning, and Dr. Robert L. West for his help with mechanical testing and programming.

I would like to thank all of my lab mates, both from the DREAMS lab and the Mechatronics lab; in particular: Viswanath Meenakshisundaram, Camden Chatham, Charles Tenney, and Logan Sturm for the many problem-solving discussions and late-night D&D sessions. I could not have asked for better friends or better players. I also acknowledge the contributions from the entire SLAMR senior design team for designing and fabricating the initial robotic printing platform, Mona Dandan for printing test specimens, Tadek Kosmal and Kieran Beaumont for designing and fabricating the tilt-turn platform, and Kelsey Rogers and Om Bhavsar for their work in designing new tool heads.

A special acknowledgement goes to the Institute for Creativity, Arts, and Technology (ICAT) both for financially supporting my graduate career and showing me the importance of being able to communicate science to a wider audience. You gave me the opportunity to work on so many interesting projects, with such incredible technology, and with such amazing people. I cannot express the positive impacts that had on both me and my research.

I also acknowledge the Advanced Research Computing (ARC) Center at Virginia Tech for providing computational resources that contributed to the results reported in this work and Krister Svanberg for kindly providing the MMA optimizer code used in the presented topology optimization algorithm.
Contents

1 Introduction 1
  1.1 Three Degree of Freedom Toolpath Planning 4
  1.2 Existing Multi-Axis Toolpath Planning 5
  1.3 Research Goal 7
  1.4 Research Questions 7
    1.4.1 Research Question 1 8
    1.4.2 Research Question 2 10
    1.4.3 Research Question 3 12
    1.4.4 Research Question 4 13
  1.5 Roadmap 15

2 Topology Optimization with 3D Material Orientations 16
  2.1 Research Gap 18
  2.2 3D Orientation Parameterizations 18
    2.2.1 Euler Angles 18
    2.2.2 Quaternions 19
    2.2.3 Natural Quaternions 20
  2.3 3D Material Orientation Problem Statement 20
    2.3.1 Sensitivity Analysis 22
  2.4 Comparison of Orientation Parameterizations on Benchmark Problems 23
    2.4.1 Implementation 24
    2.4.2 Material Properties 25
    2.4.3 Visualizations 25
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.4</td>
<td>MBB Beam Results</td>
<td>26</td>
</tr>
<tr>
<td>2.4.5</td>
<td>Wheel Results</td>
<td>30</td>
</tr>
<tr>
<td>2.4.6</td>
<td>Multi-Load Results</td>
<td>36</td>
</tr>
<tr>
<td>2.4.7</td>
<td>Computational Requirements</td>
<td>36</td>
</tr>
<tr>
<td>2.5</td>
<td>Summary of Contribution</td>
<td>40</td>
</tr>
<tr>
<td>2.5.1</td>
<td>Integration into the Layer-less Multi-Axis Workflow</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>Support Generation</td>
<td>43</td>
</tr>
<tr>
<td>3.1</td>
<td>Research Gap</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Support Generation Algorithm</td>
<td>44</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Project Node to Find Contacting Voxel</td>
<td>44</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Define Road and Build Directions</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Algorithm Validation</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Summary of Contribution</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>Deposition Path Planning</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Research Gap</td>
<td>50</td>
</tr>
<tr>
<td>4.2</td>
<td>Flow Visualization with Streamlines</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>Volumetric Deposition Path Generation Algorithm</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Algorithm Validation</td>
<td>57</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Case Study: Optimized MBB Beam</td>
<td>57</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Case Study: Optimized Wheel</td>
<td>60</td>
</tr>
<tr>
<td>4.5</td>
<td>Streamlines Mechanical Property Evaluation</td>
<td>61</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Toolpath Planning</td>
<td>63</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Material</td>
<td>64</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Mechanical Testing Setup</td>
<td>64</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Results</td>
<td>66</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary of Contribution</td>
<td>71</td>
</tr>
<tr>
<td>5</td>
<td>Deposition Path Ordering</td>
<td>73</td>
</tr>
</tbody>
</table>
5.1 Research Gap ................................................................. 74
5.2 Assembly Planning ......................................................... 74
  5.2.1 Task Ordering with Precedence Constraints ....................... 75
5.3 Toolpath Planning Algorithm ............................................. 75
  5.3.1 Establish Precedence Constraints ..................................... 76
  5.3.2 Order Deposition Paths ............................................... 77
5.4 Multi-Axis Toolpath Planning Evaluation .............................. 80
  5.4.1 Computation Time ...................................................... 83
5.5 Summary of Contribution ................................................ 83

6 Multi-Axis Deposition Platform .......................................... 85
  6.1 Extruder Design ........................................................... 85
  6.2 Travel Movements ........................................................ 85
  6.3 Toolpath Translation ...................................................... 88
  6.4 Communication Architecture ........................................... 89
  6.5 Quarter-Wheel Demonstration Print ..................................... 89
  6.6 Summary of Contribution ............................................... 90

7 Fabricating Optimized Layer-less Multi-Axis Composite Structures 91
  7.1 Inverted Wheel Load Case ................................................. 91
  7.2 Inverted Wheel Toolpath Planning ...................................... 91
    7.2.1 Conventional Toolpath Planning .................................... 93
    7.2.2 Toolpath Comparison ............................................... 95
  7.3 Fabrication and Mechanical Testing ................................... 95
  7.4 Summary of Contribution ............................................... 98

8 Summary ................................................................. 100
  8.1 Intellectual Merit ........................................................ 100
    8.1.1 Research Question 1 .................................................. 100
    8.1.2 Research Question 2 .................................................. 101
8.1.3 Research Question 3 ........................................ 103
8.1.4 Research Question 4 ........................................ 104
8.2 Publications .................................................... 105
8.3 Contributions ................................................... 108
  8.3.1 Intellectual Merits ......................................... 108
  8.3.2 Broader Impacts ........................................... 109
8.4 Future Work .................................................... 111

Bibliography ...................................................... 118

Appendix A Problem Statements for Specific Parameterizations 133
  A.1 Euler Angles ................................................ 134
  A.2 Quaternions ............................................... 135
  A.3 Natural Quaternions ....................................... 135

Appendix B Orientation Field Alignment to Principal Stresses 137

Appendix C Discussion of Stress and Modulus Measurements 139
List of Figures

1.1 Defining and aligning roads to the load paths within a structure. (a) load case, (b) load paths, (c) discretized orientation field, (d) XY-planar deposition does not result in good alignment with the load paths, whereas (e) multi-axis deposition enables strong alignment with the load paths. .................................................. 2

1.2 The author’s 6-DoF ME system. It consists of a robotic manipulator, a desktop-scale extrusion head that has been modified for multi-axis ME, and a heated build platform. More on this system is presented in Chapter 6. . . . 3

1.3 General toolpath planning algorithm for XY-planar ME. A geometry is input and decomposed into layers where the surface of the geometry is represented by contours. Overhanging regions requiring support structure are found and similarly decomposed, and both the model and support layers are populated with roads. These roads are then ordered layer-by-layer to minimize deposition head travel distance. ................................................................. 4

1.4 (a) The deposition head is orthogonal to the 45° deposition direction in the multi-axis specimen, and (b) the tensile testing results. The results show a 153% improvement in yield tensile strength in the ZYX specimens. . . . . . . . . . 7

1.5 The proposed LL-MA workflow that i) generates an optimized geometry and orientation field to address an input set of loading conditions, ii) processes that geometry and orientation field into a printable toolpath, and iii) fabricates that toolpath using a multi-axis deposition system. ........................................ 8

1.6 (a) In XY-planar printing, projecting unsupported geometry along the global Z-axis creates appropriate support structure. (b) This does not hold true in multi-axis printing as a substrate is not necessarily created for steep build directions. ................................................................. 11

2.1 (a) Diagram of the MBB beam problem and (b) Model taking advantage of the symmetry across the XZ-plane. ................................................................. 23

2.2 (a) Diagram of the Wheel problem and (b) Model taking advantage of the symmetry across the XZ-plane. ................................................................. 24
2.3 (a) Diagram of the pure tension load case and (b) Diagram of the three-point bend load case. The grey regions marked in each diagram represent fixed-density regions, where the TO algorithm only optimizes for orientation.

2.4 Sample output of four elements from (a) overhead and (b) isometric viewpoints. The shading of the element denotes the density and the colored vectors denote the material orientation. (c) A planar example of the output showing deposition directions (red) and build directions (blue) and (d) example propagated roads with an example deposition head along the respective build direction.

2.5 Sample output of nine elements colorized to represent the deposition direction. (a) the voxels visualized as described in Figure 2.4, (b) the corresponding pixels colorized to represent the $E_{1,1}$ direction, and (c) the color legend.

2.6 YZ-plane views of the MBB beam results using the CFAO algorithm and 3D parameterizations. (a) CFAO, (b) Euler angles, (c) Quaternion, and (d) Natural quaternion.

2.7 Final objective function value comparison between the different orientation parameterizations for the MBB beam problem (1x60x20 mesh). CFAO: $c_{\eta=3} = 25.077$, $c_{\eta=1} = 18.031$; Euler angles: $c_{\eta=3} = 26.203$, $c_{\eta=1} = 18.283$; Quaternion: $c_{\eta=3} = 24.836$, $c_{\eta=1} = 18.147$; Natural quaternion: $c_{\eta=3} = 24.964$, $c_{\eta=1} = 18.146$.

2.8 Convergence history of the first three continuation cycles for each orientation parameterization for the MBB beam problem.

2.9 The results of the MBB beam load case using the natural quaternion parameterization at a mesh size of: (a) 1x120x40, (b) 1x240x80, (c) 1x480x160, and (d) 1x960x320. Mesh coloring is performed according to the legend shown in Figure 2.5c.

2.10 Isometric views of the final topologies from CFAO and the 3D orientation parameterizations. (a) CFAO, (b) Euler angles, (c) Quaternion, and (d) Natural quaternion.

2.11 Final objective function value comparison between the different orientation parameterizations for the Wheel problem (15x15x15 mesh). CFAO: $c_{\eta=3} = 2.513$, $c_{\eta=1} = 2.276$; Euler angles: $c_{\eta=3} = 2.121$, $c_{\eta=1} = 2.035$; Quaternion: $c_{\eta=3} = 1.721$, $c_{\eta=1} = 1.636$; Natural quaternion: $c_{\eta=3} = 1.731$, $c_{\eta=1} = 1.636$.

2.12 Select cross-sections from the final Wheel topologies using CFAO and the natural quaternion parameterization. Cross-sections are taken along the YZ-plane and XZ-plane to demonstrate $E_{1,1}$ alignment to the density features in a single plane with CFAO and in both planes with the natural quaternion parameterization.
2.13 Convergence history of each orientation parameterization for the Wheel problem. 35

2.14 Plot of the final compliance values of the Wheel problem for each orientation parameterization against the number of elements in the design space. The quaternion parameterization could only be run for the coarsest mesh size due to computational requirements and is directly overlapped by the natural quaternion result. 35

2.15 The results of the multi-load case using the natural quaternion parameterization at a mesh size of: (a) 1x120x40, (b) 1x240x80, (c) 1x480x160, and (d) 1x960x320. Mesh coloring is performed according to the legend shown in Figure 2.5c. 37

2.16 The results of the multi-load case using: (a) CFAO with a mesh size of 1x960x320, (b) Euler angles (1x960x320), and (c) Quaternions (1x120x40). Mesh coloring is performed according to the legend shown in Figure 2.5c. 38

2.17 Plot of the final compliance values of the multi-load case for each orientation parameterization against the number of elements in the design space. The quaternion parameterization could only be run for the coarsest mesh size due to computational requirements. 39

2.18 The computation times for each parameterization plotted against the number of elements in the design space (all load cases are shown together). A linear fit is shown to aid in visualizing the effect of the parameterization. All of the meshes that were able to be completed are plotted; the explicit quaternion parameterization was only able to complete coarse meshes due to large memory and computation time requirements. 39

2.19 (a) The deposition head, with interference angle $\theta_T$, is perpendicular to the build platform, (b) the deposition head is at an angle $\theta_B$ but does not collide with the build platform, and (c) if $\theta_B$ gets too large, the deposition head collides with the build platform. 42

3.1 (a) Model voxels shown in grey with build directions shown in blue, (b) Populate $O$ with the model voxels (yellow) and project nodes of voxel 11 to find supporting voxels, (c) Defining $A$ after projecting all voxels in $O$, (d) Assign build directions and add all newly defined support voxels to $O$, (e) Repeat projection step, and (f) Fully defined support material for the multi-axis model. 46

3.2 (a) One quarter of an optimized Wheel geometry. The optimized build directions require dedicated support structure for successful fabrication and (b) Quarter-Wheel geometry with support voxels propagated (blue). 48
4.1 VDPG algorithm flowchart. $B$ is a seed point buffer used to hold seeds that have not been advected yet, and $Q$ is the streamline length-weighted priority queue. Highlighted stages are elaborated in the text. 

4.2 Illustration of the advection and seeding process. (a) The particle velocity is mapped from the nodal orientations, (b) a step is taken along the velocity vector closest to the previous step, (c) the streamline is terminated either when exiting the orientation field or when in too close proximity to another streamline, and (d) additional seed points are created around the streamline to continue propagation.

4.3 VDPG algorithm applied to an optimized 2.5D MBB beam geometry. (a) the input structure and deposition orientation field, (b) map the orientations to the nodes, (c) propagate streamlines through the structure, (d) linearize the streamlines for deposition, and (e) visualization of the final toolpath.

4.4 Toolpath generated via Slic3r for the comparison of volumetric coverage.

4.5 Optimized quarter-Wheel orientation field. The deposition directions are shown with two-sided red vectors and the build directions are shown with blue vectors.

4.6 Roads propagated through the quarter-Wheel geometry. Model material paths are shown in black, and support material is shown in blue.

4.7 The multi-load case used throughout this work. The design space experiences two load cases: (a) tension and (b) three-point bending and has fixed regions allocated for test fixturing.

4.8 (a) The optimized multi-load case geometry. (b) Material orientations are denoted by the color legend. The legend is symmetric across the Y-axis; a desired deposition direction along, for instance, the +X axis is the same as being aligned to the -X axis.

4.9 VDPG specimens positioned for (a) tensile testing and (b) three-point bending.

4.10 Toolpaths from each (a) VDPG, (b) uniform, and (c) contour, and printed specimens from the corresponding toolpaths: (d) VDPG, (e) uniform, and (f) contour.

4.11 Measurement of toolpath alignment relative to the stress-aligned orientation field. 1 would denote a fully aligned toolpath and 0 would denote a toolpath fully orthogonal to the input orientation field.

4.12 The maximum load applied to the printed structures in each load case are normalized by the mass of the structure. The VDPG toolpath outperformed the uniform toolpath by 108.24% and 29.25% in tension and bending, respectively.
4.13 (a) Catastrophic failure of the bending sample produced a higher maximum load, and (b) a premature failure at the point of support reduced the maximum load.

5.1 As the deposition head (grey) deposits road A, aligned to the build direction of road A (blue arrow), it needs to occupy a certain volume (black dotted line). If another road (B) occupies that volume, road A needs to be printed first in order to ensure the volume is available during printing.

5.2 High level overview of the ordering algorithm used to create a collision-free toolpath. Considerations for precedence constraints, road continuity, and minimizing deposition head movement are included. Highlighted regions are discussed in more detail in the text.

5.3 Collision matrices with model path collisions shown in black and support paths shown in blue: (a) the unordered collision matrix, (b) the collision matrix after the first ordering is upper triangular, but required the removal of 1312 roads, and (c) the second ordering also produced an upper triangular matrix, but required the removal of 1455 roads.

5.4 Continuity matrices with model path continuities shown in black and support paths shown in blue: (a) the unordered continuity matrix has perfect continuity as the paths are still in the order they were generated, (b) the first ordering resulted in 3548 properly sequenced continuous paths out of a total of 7098 (1357 continuous model roads were sequenced properly out of 3013), and (c) the second ordering resulted in 4248 properly sequenced continuous paths (1701 properly sequenced continuous model roads).

5.5 Ordered toolpath for the quarter-Wheel geometry, colorized based on the order in which roads should be deposited.

6.1 6-DoF ME system used to fabricate the example geometry and orientation field. It consists of a robotic manipulator, a desktop-scale extrusion head (inset) that has been modified for LL-MA ME, and a heated build platform.

6.2 High-level algorithm showing the insertion of travel movements between discontinuous roads. If the path is not collision-free with a convex approximation of the current print, the path is iteratively offset along its build direction and bisected until it is collision-free.

6.3 Example translation from augmented GCode to RAPID. Highlighted are the (blue) point and (orange) quaternion pair defining the pose of the robot, (yellow) extrusion length information, and (green) linear tool speed. Additional call outs are defined in the text.
6.4 Fabricated quarter-Wheel geometry. (a) The support structure was too sparse along the Z-axis to maintain the typical linear roads, resulting in poor deposition quality, but it was still sufficient to support the deposition of the model material. (b) After removing the support structure, the printed part qualitatively matches the input geometry and orientation field.

7.1 The inverted Wheel loading case is used to validate the LL-MA workflow for true 3D load paths. The load is applied to the top face (rather than the bottom face in the standard Wheel problem) to enable a bending test set up to be used.

7.2 Optimized design space for the inverted Wheel load case leveraging the symmetry across the XZ- and YZ-planes. The orientation field approximately aligns with the density features, obeying Michell’s theorem.

7.3 (a) The ordered collision matrix for the inverted Wheel, and (b) the continuity matrix. While the toolpath is collision-free, not all roads were able to be printed sequentially.

7.4 Toolpath for the inverted Wheel with roads colored by print order. Inset images highlight regions where roads were forced to be printed discontinuously.

7.5 The conventional toolpath (a) with and (b) without support structure.

7.6 Fabricated multi-axis specimen (a) with and (b) without support; fabricated conventional specimen (c) with and (d) without support. Features in the center of the multi-axis specimen were damaged during the support material removal process.

7.7 Three-point bend test setup for the inverted Wheel geometry.

7.8 Plot of the maximum load normalized by the mass of the specimen for each the conventional and multi-axis toolpath. The conventional toolpath achieved a normalized maximum load of 30.68 $N/g$ while the multi-axis toolpath only achieved 15.96 $N/g$.

7.9 (a) Fracture surface of a multi-axis inverted Wheel specimen. The right portion of the specimen is laying flat, rather than standing as designed. Deposition path fraying can be seen in many of the middle features, highlighting deficiencies in the toolpath and deposition platform. (b) Fracture surface of a conventional inverted Wheel specimen. The failure occurred along the layer interfaces, as anticipated.
8.1 VR environment showing the processing planning workflow for the quarter-Wheel geometry: (a) an intermediate TO iteration with load case shown, (b) deposition path planning during an intermediate iteration, and (c) printing the final toolpath with a simulated 6-DoF robot. .................................................. 110

8.2 Summary of future work, grouped in terms of the expected impact on the performance and functionality of the workflow. The coloration of each item corresponds to its approximate priority. .................................................. 111

8.3 Large-scale pellet-fed robotic extrusion platform (photo credit: Ben Woods) 116

8.4 Tilt-turn table for adding a 7th and 8th axis to the robotic deposition platform (photo credit: Tadek Kosmal and Kieran Beaumont) ......................... 117

B.1 Mesh convergence studies using the natural quaternion parameterization and principal stress method for the (a) MBB beam, (b) Wheel, and (c) MBB beam and tensile combined load cases. .................................................. 137

B.2 Combined load case results using the principal stress method for the most refined mesh size; results exhibit checkerboarding in the orientation field. . 138

B.3 Computation time comparisons between each parameterization, including the principal stress method. .................................................. 138

C.1 Average cross-sectional area measurements of the specimens produced using each toolpath planning strategy. The cross-section is taken across the central XY-plane (i.e., the sum of the three truss members that cross that plane). . 140

C.2 Stress-strain curves for an example (a) uniform tensile test, (b) contour tensile test, and (c) VDPG bending test. The blue dashed line represents a linear fit with an $R^2 > 0.99$ from the first point of load, and the orange dashed line represents the tangent modulus as determined using the method described in ASTM E111 [1]. .................................................. 140
List of Tables

1.1 Roadmap that correlates the RQs and objectives with each chapter. . . . . . . 15

2.1 Shared parameters for the test case examples . . . . . . . . . . . . . . . . . . 25

2.2 Mesh sizes for the three test problems . . . . . . . . . . . . . . . . . . . . . 26

4.1 Selected process parameters for validation of the VDPG algorithm. . . . . . . 59

4.2 Road propagation parameters used for the model and support structures. . . 60

4.3 Toolpath planning parameters for VDPG and the control (uniform and con-
tour) toolpath planning strategies. Parameters were kept as consistent as
possible between the strategies to minimize the effects of the specific param-
eterization. For the control parameters, single-valued parameters apply to
both uniform and contour toolpaths; for two-valued parameters, the value in
the parentheses refers to the contour toolpath. . . . . . . . . . . . . . . . . . . . . . . . . 64

7.1 Multi-axis toolpath planning parameters for model and support material. . . 92

7.2 Parameters used for the conventional inverted Wheel toolpath generated by
Slic3r. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95

8.1 Journal and conference publications directly relevant to the presented work
with references to the associated chapter and RQ. . . . . . . . . . . . . . . . . . . 106
List of Abbreviations

ABS  Acrylonitrile butadiene styrene
AM   Additive manufacturing
CAD  Computer aided design
CAM  Computer aided manufacturing
CF   Carbon fiber
CFAO Continuous fiber angle optimization
CFD  Computational fluid dynamics
CLS  Curved layer slicing
DMO  Discrete material optimization
DoF  Degree(s) of freedom
FMO  Free material optimization
GJK  Gilbert-Johnson-Keerthi
HPM  Heaviside projection method
LL-MA Layer-less multi-axis
MBB  Messerschmitt-Bölkow-Blohm
ME   Material extrusion
MMA  Method of moving asymptotes
NP   Nondeterministic polynomial
PLA  Polylactic acid
PRH  Primary research hypothesis
RK   Runge-Kutta
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ</td>
<td>Research question</td>
</tr>
<tr>
<td>RRT</td>
<td>Rapidly exploring random trees</td>
</tr>
<tr>
<td>SIMP</td>
<td>Solid isotropic material with penalization</td>
</tr>
<tr>
<td>SO</td>
<td>Shape optimization</td>
</tr>
<tr>
<td>STL</td>
<td>Stereolithography (file format)</td>
</tr>
<tr>
<td>TO</td>
<td>Topology optimization</td>
</tr>
<tr>
<td>TSP</td>
<td>Traveling salesman problem</td>
</tr>
<tr>
<td>VDPG</td>
<td>Volumetric deposition path generation</td>
</tr>
<tr>
<td>VR</td>
<td>Virtual reality</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

While additive manufacturing (AM) is often referred to as “3D printing,” the technologies typically only deposit material in 2D planes on a layer-by-layer basis. These repetitive, stacked layer interfaces result in a part with anisotropic properties wherein weakness is aligned with the build direction [2]. This is especially true of components made with material extrusion (ME; e.g., fused filament fabrication) AM processes, in which poor inter-layer [3] and intra-layer [4] (i.e., between adjacent depositions) bonding results in mechanical properties that are much weaker than those found in their traditionally manufactured counterparts [5]. However, the anisotropy of extruded fiber-reinforced materials (e.g., short fiber-reinforced polymers and continuous fiber-reinforced coextrusions) presents an opportunity to offer a transformative step-change in the properties of AM parts. Specifically, at the nexus of multi-axis robotics, AM technologies, and topology optimization (TO), there is a vision for a multi-axis AM design workflow that enables concurrent optimization of part topology and printing toolpath such that the anisotropy of the deposited material is preferentially and specifically aligned within the printed structure to maximize part performance (Figure 1.1). Such a workflow could enable a carbon fiber-reinforced thermoplastic to be deposited such that it is aligned to a topology-optimized part’s anticipated three-dimensional (3D) load paths (which collectively form an orientation field). As such, the author proposes a new engineering paradigm in which AM moves away from layered 2D printing and towards leveraging the inherent anisotropy of the deposited material in a true 3D deposition process to maximize part performance.

The overall research goal (Section 1.3) of this work is to improve the structural efficiency and mechanical performance of composite structures through a comprehensive design workflow that optimizes both part topology and printing toolpath for multi-axis AM technologies (e.g., the author’s multi-axis deposition platform shown in Figure 1.2). A key outcome of this research is a computational framework for optimizing part topology and multi-axis printing toolpath for any arbitrary geometry and any 3D orientation field. While this framework allows for concurrent optimization of structure and any performance objective (e.g., heat transfer, vibration mitigation, etc.), the primary validation focus of this research is on mechanical performance (i.e., maximizing part stiffness). As such, the primary research hypothesis (PRH) is that tailoring topology and toolpath to 3D load paths will increase
Figure 1.1: Defining and aligning roads to the load paths within a structure. (a) load case, (b) load paths, (c) discretized orientation field, (d) XY-planar deposition does not result in good alignment with the load paths, whereas (e) multi-axis deposition enables strong alignment with the load paths.

structural efficiency (i.e., increase stiffness and/or decrease mass) by 150%, relative to conventional printing strategies. This hypothesis stems from a previous study (Figure 1.4) performed with homogeneous acrylonitrile butadiene styrene (ABS); if composite-reinforced materials were used instead, the percent improvement is expected to increase.

Testing this hypothesis requires a process planning workflow that optimizes part performance while concurrently considering part topology, multi-axis printing toolpath, and potential collisions between the deposition platform and the part being printed. In general, existing TO approaches are insufficient as they generally only consider density distribution (i.e., the shape of the part); while some TO methods incorporate manufacturability of the designed shape during the analysis (e.g., by imposing minimum feature size [6] and overhang constraints [7]), they do not include considerations for the toolpath used to fabricate the structure. The presented research aims to avoid such performance compromises by realizing a TO approach that can determine both material distribution and orientation in 3D space.

Multi-axis AM has been demonstrated in literature, but the focus has typically been on reducing support structure (e.g., [8, 9, 10]) or conformally printing on non-planar substrates (e.g., [11, 12]). Recent works have demonstrated mechanical property improvements using multi-axis deposition, but their toolpath planning capabilities heavily limit the classes of geometries and associated orientation fields that can be fabricated (Section 1.2). Specifically, while 3-DoF ME is supported by a number of robust toolpath planning algorithms (reviewed in Section 1.1) that allow any arbitrary geometry to be input, multi-axis ME introduces a number of challenges not addressed in XY-planar ME:
Support structure. The support structure in XY-planar ME relies on the deposition head maintaining a single orientation throughout the print. If the deposition head changes orientation, as in multi-axis ME, support structure generated using XY-planar algorithms may not produce a substrate suitable for deposition.

Planning roads. Typical XY-planar ME places constraints on the directions of the roads and the orientation of the deposition head. Specifically, all roads must be located in the XY-plane, and the deposition head only ever assumes a single orientation. In multi-axis ME, these constraints are intentionally removed, but this renders existing slicers insufficient for planning multi-axis roads.

Collision avoidance. The layer-by-layer deposition strategy used in XY-planar ME always places the deposition head above the previously deposited layers. Therefore, collisions between the deposition head and the printed part are inherently avoided. In multi-axis ME, it is not necessarily safe to offset the deposition head in this fashion; by nature of depositing material along arbitrary vectors, material could be both above and below the deposition head at any point in the toolpath. Additional considerations must be taken to prevent the deposition head from colliding with previously deposited material.

Due to the complexity of these outlined issues, existing multi-axis toolpath planning methods (reviewed in Section 1.2) make assumptions in efforts to simplify the toolpath planning problem. The research questions (RQ) to be addressed in this work, focusing on generalizing these functionalities and evaluating the mechanical properties possible with multi-axis deposition, are presented in Section 1.4, and a roadmap of the research is presented in Section 1.5.
Chapter 1. Introduction

1.1 Three Degree of Freedom Toolpath Planning

There are a variety of toolpath planning algorithms for XY-planar ME [13], but at a high-level, each executes the same steps, shown in Figure 1.3:

![Diagram of toolpath planning steps: Import Geometry, Slice into Layers, Propagate Support, Plan Roads, Export Toolpath.]

Figure 1.3: General toolpath planning algorithm for XY-planar ME. A geometry is input and decomposed into layers where the surface of the geometry is represented by contours. Overhanging regions requiring support structure are found and similarly decomposed, and both the model and support layers are populated with roads. These roads are then ordered layer-by-layer to minimize deposition head travel distance.

1. **Assign build orientation.** The part is oriented to the desired build direction, which may be different than the orientation in which it was designed.

2. **Slice into layers.** Once oriented to the chosen build direction, the geometry is cut into discrete, parallel layers.

3. **Propagate support.** Support structure is created in regions of the part that feature steep overhangs to enable successful printing.

4. **Plan roads.** Each sliced layer, consisting of boundary paths that describe the input geometry at a single Z-height, is filled with roads.

5. **Order roads.** The roads are ordered within each layer, typically by minimizing the travel time of the deposition head.

6. **Export toolpath.** The resulting ordered roads constitute the toolpath and are typically sent to the printer in the form of a GCode program.

The toolpath planning process can be tailored depending on the requirements of the part. For instance, if the mechanical performance of the part is paramount, each layer could be completely filled with roads; this would increase the material usage and manufacturing time over a less dense toolpath. If a high quality surface finish is required, the part orientation could be assigned to match the curvature of the part. Depending on the part geometry, this may incur a trade-off with manufacturing time and mechanical properties though. As
such, although XY-planar ME is supported by robust algorithms capable of handling very complex geometries, the process is constraining and often requires design compromises.

Advances in adaptive slicing techniques (e.g., [14, 15]) have relaxed some of these trade-offs by allowing the geometry to be sliced using different layer heights throughout the part. These different layer heights enable the discrete layers to more accurately follow the part curvature in highly curved regions (to improve surface finish) while minimizing print time in regions that more closely approximate a 2.5D geometry. Trade-offs are still present though; in particular, adaptive slicing does little to mitigate the issue of anisotropic mechanical properties (especially along the build direction).

In terms of mechanical performance, a number of different infill patterns and road propagation strategies have been demonstrated in literature (e.g., [2, 16, 17]). For a planar loading case that only acts within the slicing plane, these strategies are effective in removing the weaker intra-layer bonds from the anticipated load paths. The roads are still constrained to the slicing plane though, which limits the ability for the process to accommodate more complex (e.g., multi-axial) loading conditions. In these cases, it is often not possible to fully remove the inter-layer bonds from the load paths, reducing the mechanical properties of the printed part.

Curved layer slicing (CLS) still divides the geometry into discrete layers but does so using curved (i.e., non-planar) layers [18]. This can both improve the surface finish of the final part [19, 20] and the final mechanical performance by allowing roads to be partially aligned with the build direction [21, 22]. Critically, depositing material along curvature requires material to be both above and below the deposition head at certain parts of the tool path. Therefore, collision concerns during travel movements must be managed appropriately to prevent the deposition head from colliding with previously deposited material [19]. This imposes limitations on the allowable curvature of the layer, as the layer can only be so highly curved before collisions between the deposition head and the layer are unavoidable (similar to global collisions in subtractive manufacturing [23]). In turn, this limits the degree to which roads can be aligned to the build direction. Additionally, for complex loading conditions, the anticipated load paths may not conform to extractable surfaces. In order to properly avoid collisions and remove these limitations, additional flexibility is required from the deposition system that 3-DoF systems cannot provide. Specifically, the rigid orientation of the deposition head relative to the printed part only allows a single build direction to be used throughout the part.

1.2 Existing Multi-Axis Toolpath Planning

The reorientation capabilities of multi-axis ME systems enable the use of multiple build directions throughout a single part. Leveraging this freedom, the toolpath used to create the part can be highly tailored to the manufacturing requirements of the part. For instance,
build directions can be selected to minimize (and often completely remove) the support structure underneath overhanging features [8, 9, 10]. These methods are often embodied by decomposing the overall part geometry into distinct regions, each assigned a unique build direction [24, 25]. Each distinct region can then be sliced individually using slicing software similar to those used in XY-planar ME [26]. These planar multi-axis ME algorithms have also been developed for printing partitioned geometries onto cuboid inserts [12] and improving surface finish [24]. Planar multi-axis methods benefit from being able to leverage existing slicing software to plan roads and partially mitigate the issue of collisions by depositing discrete layers (although collisions can still exist depending on the geometry [24]).

Improving mechanical properties is difficult with planar multi-axis ME due to poor bonding at the interfaces between regions with different build directions. Specifically, these inter-region bonds have been demonstrated to have worse performance than the inter-layer bonds in XY-planar prints [10]. Therefore, planar multi-axis ME suffers from the same shortcoming as XY-planar ME; unless the geometry’s anticipated load paths travel along stratified planar layers, it is not possible to remove the inter-layer, nor inter-region, bonds from the load paths.

Non-planar (i.e., curved layer) multi-axis ME provides additional flexibility in the customization of mechanical properties. This benefit was demonstrated by Yerazunis et al. with a 5-DoF system, used to fabricate hemispherical pressure caps with stress-aligned roads [27]. These stress-aligned specimens ruptured at pressures 4.5 times greater than specimens fabricated through XY-planar ME. Tam and Mueller fabricated topology optimized 2.5D surface geometries using a 6-DoF robotic arm, which demonstrated increased maximum compressive load when compared to geometrically similar XY-planar parts [28]. Fang et al. populated a principal stress field with roads propagated along isosurfaces to produce multi-axis tool-paths [29]. Fabricated specimens withstood 6.35 times greater load than conventionally printed (i.e., via XY-planar layers) specimens. Previous work from the author demonstrated the improvement of tensile specimens through the multi-axis ME of conformal surface reinforcement using a 6-DoF robotic arm [30]. The yield tensile strength of the multi-axis reinforced specimens was 59% greater than unreinforced XY-planar specimens. In each case, the anticipated load paths experienced by the structure developed stratified surfaces. Each curved surface could be deposited independently to prevent collisions between the deposition head and printed part, provided the surface was not too highly curved, similar to CLS. If the load paths did not follow stratified surfaces though, these non-planar multi-axis methods would be unable to completely align the deposition directions with the load, resulting in suboptimal material orientation.

Handling load paths that do not follow stratified layers (referred to here as volumetric load paths) requires layer-less deposition, which completely removes layer-like structures from the toolpath. While imparting a large amount of flexibility to the process, the road propagation problem becomes much more difficult. Additionally, collisions between the deposition system and previously deposited material must be handled carefully in the process planning stage to enable successful fabrication. Currently, layer-less multi-axis (LL-MA) ME is limited to the fabrication of wireframe structures (e.g., [31, 32]). While demonstrative of LL-MA ME,
wireframe structures do not allow for the alignment of roads with volumetric load paths. Therefore, a gap has been identified in the scope of a workflow for processing an arbitrary geometry with volumetric load paths into a manufacturable LL-MA toolpath.

1.3 Research Goal

The goal of this research is to create a holistic multi-axis process planning workflow that concurrently optimizes both part topology and its printing toolpath. Such a workflow could be used to align highly anisotropic composite extrudate with a part’s load paths (i.e., a strain-based orientation field), resulting in high strength, lightweight structures that could find application in aerospace, automotive, prosthetics, and athletic equipment markets. The author’s previous work drives the PRH (Figure 1.4); multi-axis tensile specimens, fabricated with roads aligned with the direction of applied load, showed up to 150% improvement in strength when compared to specimens fabricated with conventional planar printing techniques [26]. This result occurred despite the specimens being fabricated from homogeneous ABS, and the degree of improvement is expected to increase with composite-reinforced materials wherein the extrudate’s anisotropy is directly aligned with the load.

Figure 1.4: (a) The deposition head is orthogonal to the $45^\circ$ deposition direction in the multi-axis specimen, and (b) the tensile testing results. The results show a 153% improvement in yield tensile strength in the ZYX specimens.

1.4 Research Questions

As previously established, existing multi-axis toolpath planning techniques are insufficient for the proposed method of LL-MA deposition. To address this gap, a workflow for creating,
processing, and fabricating geometries with volumetric load paths is presented in Figure 1.5. This workflow can be interpreted as a 3D generalization of the typical toolpath planning workflow for XY-planar deposition: i) geometry generation, ii) process planning (i.e., slicing in the context of XY-planar deposition), and iii) printing. RQs that drive the generalization of these sub-functions are presented below with a brief summary of the challenges associated with each.

![Figure 1.5: The proposed LL-MA workflow that i) generates an optimized geometry and orientation field to address an input set of loading conditions, ii) processes that geometry and orientation field into a printable toolpath, and iii) fabricates that toolpath using a multi-axis deposition system.](image)

**1.4.1 Research Question 1**

LL-MA deposition imparts a high degree of customization in both the geometry of the part and the toolpath used to fabricate it by allowing material deposition in full 3D. Therefore, the use of LL-MA deposition, in particular the deposition directions, should be performance-driven (i.e., it should be used when the performance of parts printed using XY-planar layers is insufficient). In the case of previous multi-axis work (e.g., [27, 28, 29]), a performance-driven orientation field was explicitly created. Analysis tools determined lines of principal stresses, and roads were propagated along those lines. For [30], the orientation field was not
explicitly created, as the load paths were simplified in the tensile specimen geometry, but load paths still informed the toolpath. As previously described though, these toolpath planning methods are only applicable to surface geometries, and a novel method of identifying these orientation fields is required for arbitrary volumetric geometries. Therefore, the primary goal of this research phase (RQ 1) is to establish both i) an optimized geometry and ii) an associated orientation field that describes the optimal topology and deposition directions for a set of anticipated load cases.

**Research Question 1**

What are i) the optimal geometry and ii) optimal material orientations for an arbitrary set of loading conditions for a given design space?

Existing methods in literature have explored identifying material orientations for printing in a planar sense. For example, Hoglund and Smith used a TO method called Continuous Fiber Angle Optimization (CFAO) to develop a voxel-based orientation field corresponding to the optimal deposition directions [33, 34]. While CFAO was demonstrated to develop an appropriate orientation field for planar loading conditions, there are no accommodations for 3D loading conditions and material orientations. Level-set methods have also been developed that implicitly assume the form of the orientation field as concentric level-sets within the external contours of the geometry [35, 36]. In this case, the form of the orientation field is coupled to the optimized geometry, which could be limiting for complex or multi-load cases where the orientation field does not necessarily match the contours of the geometry (e.g., 37).

The following two objectives serve to address RQ 1. As discussed, existing TO methods can determine optimal topology and deposition directions for planar loading cases. Objective 1 generalizes these solutions for the 3D case, where the loading cases are not necessarily planar.

**Objective 1**

Identify the optimal topology and 3D orientation field corresponding to an arbitrary set of loading conditions.

As will be discussed in Chapter 2, 3D orientations are difficult to optimize due to highly-curved, non-convex design spaces. Therefore, a second objective seeks a computationally efficient orientation parameterization for the 3D orientation (Objective 2).
Objective 2

Identify a computationally efficient orientation parameterization for the 3D material orientation design space.

Geometric Representation

The de facto geometric representation for AM technologies is the stereolithography (STL) file, which represents the surface of a geometry using a triangle tessellation [38]. While commonplace, the STL file is frequently reported to have deficiencies in terms of incorporating information regarding the fabrication of the part [39]. Critically, this implies that surface-based representations are incapable of directly associating an orientation field with the geometry. Therefore, a different geometric representation is required to enable LL-MA fabrication.

A number of file formats have been suggested to replace the STL file [40], but voxel-based formats are of particular interest for applications in LL-MA ME. Voxels are uniquely defined throughout the geometry, which enables the integration of other useful information for manufacturing including material type [41] and material properties [37]. With this in mind, the workflow presented in this work requires that a voxel-based representation of the desired geometry can be created with an associated orientation field (e.g., Figure 1.1c).

1.4.2 Research Question 2

Once both the geometry and orientation field have been established, multi-axis roads must be propagated through the structure with appropriate support material. As discussed, the voxel-based results from a TO algorithm are not directly suitable for deposition via the ME process. Although roads could be propagated through each voxel individually, this would produce discrete interfaces between the voxels and reduce mechanical performance [10]. Instead, a method of propagating roads through the orientation field continuously (i.e., neighboring voxels are connected with the same roads) is required to achieve the desired mechanical performance.

Research Question 2

How does LL-MA deposition change the road propagation problem for an arbitrary geometry and orientation field?

Before roads can be planned, the optimized structure must be properly supported. Multi-axis
1.4. Research Questions

ME utilizes multiple build directions throughout a single part in order to place roads with increased flexibility. Many multi-axis algorithms select these build directions specifically to minimize (or eliminate) the support structure required during fabrication (e.g., [8, 9, 10]). If the build directions are determined using a specified orientation field though, it may be necessary to have explicit support structure to enable successful fabrication.

Support structure in XY-planar ME is propagated along the global Z-axis (i.e., the build direction), as shown in Figure 1.6a, to support overhanging features by providing i) a substrate for deposition and ii) support during the cooling process to preserve feature positioning. In the context of multi-axis ME, this method of support propagation does not necessarily generate structures that satisfy both criteria. As shown in Figure 1.6b, the support structure propagated along the global Z-axis does not provide a substrate for deposition. Therefore, some more intelligent method of propagating support structure along multiple build directions is required. Objective 3 focuses on the generalization of the XY-planar support material propagation process to an arbitrary set of build directions.

![Figure 1.6](image)

Figure 1.6: (a) In XY-planar printing, projecting unsupported geometry along the global Z-axis creates appropriate support structure. (b) This does not hold true in multi-axis printing as a substrate is not necessarily created for steep build directions.

**Objective 3**

Propagate suitable support structure for LL-MA structures with constrained and variable build directions.

Propagating roads through a volumetric orientation field requires connecting discrete voxels, each with unique orientations as shown in Figure 1.1c, with continuous roads. Methods for propagating continuous roads have been shown in literature (e.g., contour-based methods [34] and level-set methods [35, 36]), but these methods do not scale up to 3D structures.
Specifically, they rely on extracting the roads directly from the layers of an optimized geometry. In 3D, these extracted shapes would be surfaces, rather than lines and also would not necessarily follow the load paths. Instead, a volumetric approach to creating roads is preferred. Ezair et al. presented a method for volumetrically generating roads using a series of isosurfaces throughout the geometry [42]. Although they demonstrated the ability to propagate a 3D structure with a predefined path shape, the ability to follow an orientation field was not addressed. Therefore, a novel method for propagating roads through a voxel-based geometry with a supplied orientation field must be created (Objective 4).

**Objective 4**

Translate arbitrary geometries and corresponding orientation fields into roads.

### 1.4.3 Research Question 3

In XY-planar ME, layers are printed sequentially in terms of ascending Z-height. This prevents the deposition head from colliding with previously deposited material, as it guarantees that no material will ever be above the deposition head [43, 44]. In LL-MA ME however, roads cannot be ordered in this manner, as they often change Z-heights along their length. This Z-height variation combined with non-uniform build directions throughout the part does not allow LL-MA ME to make this assumption; an improperly planned toolpath could introduce collision concerns where previously deposited material prevents the deposition head from accessing undeposited regions of the part [45]. Therefore, a more intelligent ordering algorithm is required for LL-MA ME (RQ 3).

**Research Question 3**

How does LL-MA deposition change the problem of collision-free printing for an arbitrary set of unconstrained roads?

Rather than making an assumption that relies on the structure of the print (e.g., sequential layer stacking), a LL-MA ordering algorithm must take into account the build direction and deposition head footprint required by each road. This consideration is similar to the issue of tool head access experienced in assembly problems with traditional manufacturing (reviewed in Section 5.2). In both cases, the tool head must have access to a certain region in order to perform a given task. The assembly problem looks at a set of tasks and orders them such that the tool head has access to the required region at every point in the process. Wu et al. presented a similar algorithm for fabricating wireframe meshes on a 5-DoF system [32].
Although their implementation enabled the fabrication of a wide range of mesh structures, the algorithm occasionally produced an unprintable result for printable meshes and resulted in large computation times for relatively small numbers of roads.

One consideration not accommodated in typical assembly problems is the desire to perform certain tasks in a sequential order, such as depositing certain roads continuously. Specifically, in order to improve the final mechanical performance of the printed structure, it is desirable to print roads that share an end point one after the other in order to maintain continuity. If this continuity is not maintained, the shared end point would require bonding similar to the inter- and intra-layer bonds in typical XY-planar ME, resulting in a reduction of mechanical performance. A suitable ordering algorithm must therefore satisfy these precedence constraints while promoting continuous roads without requiring a full combinatoric exploration of the possible road orders; these considerations are addressed in Objectives 5 and 6.

### Objective 5

Establish precedence constraints for all of the LL-MA roads given a deposition head geometry.

### Objective 6

Order the roads to satisfy the precedence constraints while promoting continuous deposits.

### 1.4.4 Research Question 4

A number of works have been published exploring the effects of composite reinforcements on both the deposition process and the resulting mechanical properties [46, 47, 48, 49, 50, 51]. The characteristic ME deposition process preferentially aligns the reinforcement within a composite material along the deposition direction, increasing mechanical performance along that direction [52]. By tailoring the deposition direction, the mechanical properties of the resulting parts can be tuned to the end-use application [53].

Despite the benefits to the in-plane mechanical properties of XY-planar deposition, the strength along the build direction actually decreases when using a reinforced material [5]. This increases the overall anisotropic performance of parts fabricated via ME and, for multiaxial loading cases, could result in worse performance than a similar part fabricated using a homogeneous material. As such, a number of works have pointed out the necessity of developing some method of obtaining reinforcement alignment in the build direction [54, 55,
Additionally, even in the context of XY-planar printing, current infill patterns often align the reinforcement to sub-optimal directions that do not properly leverage composite materials [16].

While composite materials nominally have the desired mechanical properties for many end-use components, XY-planar ME is incapable of actualizing those properties in printed parts [57]. With the ability to obtain fiber alignment in the build direction, LL-MA deposition has the potential to fully align the reinforcement of a composite material with 3D load paths [16]. RQs 1, 2, and 3 outline the objectives to achieve a generalized LL-MA toolpath planning workflow (Figure 1.5), and RQ 4 focuses on evaluating the possible mechanical property improvements when using the workflow in the context of composite materials.

**Research Question 4**

How does LL-MA deposition change the mechanical properties of optimized parts printed via the ME process?

In order to fabricate the ordered roads resulting from the process planning stage of the workflow, joint trajectories must be calculated for the multi-axis deposition system. Specifically, these joint trajectories must move the deposition head along the roads and not cause the robot to collide with the printed structure or the environment. The process of generating these joint trajectories for the ordered roads is captured in Objective 7. Objective 8 regards the fabrication and mechanical testing of geometries using the workflow outlined in Figure 1.5.

**Objective 7**

Convert the ordered set of roads into a list of joint trajectories and extrusion commands for the multi-axis deposition system to execute.

**Objective 8**

Fabricate and mechanically evaluate LL-MA geometries and compare them to geometrically similar parts fabricated using the same material and XY-planar layers.
1.5 Roadmap

Table 1.1 correlates the chapters with their respective RQ and objectives. Chapter 2 presents a novel TO problem statement that efficiently solves the simultaneous material distribution and orientation optimization problem in 3D. Chapter 3 outlines a support material generation algorithm that is compatible with LL-MA deposition. Chapter 4 discusses a deposition path planning algorithm, based on streamline placement algorithms, that converts the voxel-based results of the TO algorithm into roads. Chapter 5 describes the algorithm used to evaluate precedence constraints and order the propagated roads in a collision-free manner for execution on a multi-axis system. Chapter 6 outlines the multi-axis printing platform, communication architecture, and robotic toolpath converter. Chapter 7 discusses the printing and mechanical evaluation of multi-axis specimens using a reinforced filament material. Finally, Chapter 8 summarizes the work, contributions, and paths for future work.

<table>
<thead>
<tr>
<th>RQ</th>
<th>Chapter</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1 &amp; 2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5 &amp; 6</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Chapter 2

Topology Optimization with 3D Material Orientations

TO is a family of optimization techniques that find the optimal material distribution (i.e., structure) within a given design space for a prescribed set of loading and boundary conditions [58]. The geometric freedom afforded by AM technologies has better enabled the fabrication of TO-generated structures [59, 60], but AM-specific manufacturing constraints and considerations are still necessary. For instance, TO formulations have been presented in literature to ensure self-supporting optimized geometries [61, 62], enforce minimum feature size constraints [6], and optimally distribute multiple material types [7].

In the same vein, the challenges of anisotropic mechanical properties must be considered in the optimized geometries. A commonly used TO method, Solid Isotropic Material with Penalization (SIMP), finds the optimal material layout through a finite element approximation of the design space [63, 64]. SIMP has been adapted to enable the use of orthotropic and anisotropic material properties, but in its typical formulation, SIMP does not have considerations for material orientation. When designing for anisotropic materials though, the orientation has a large impact on the final performance of the structure. Therefore, extensions to SIMP have been developed that incorporate material orientation considerations.

By Michell’s theorem, the truss members within an optimal structure coincide with the paths of principal stresses when subjected to a single loading condition [65]. This theorem has been used to develop methods of determining material orientation along lines of principal stresses [66, 67] and principal strains [68]. These methods have also been adapted for use in multiple loading cases [69, 70]. For complex loading conditions though, more flexibility is often required of the orientation optimization technique [71].

A more direct and flexible method of optimizing material orientation is to directly control the orientation through additional design variables [71]. One method of parameterizing the orientation space is through CFAO, whereby an Euler angle design variable is assigned to each element in the design space [72, 73]. The CFAO method has been demonstrated in the context of cellular automata [74] and for shell structures [75]. CFAO has also been used to inform a 3-DoF (XY-planar) ME process [34]; printable toolpaths were generated using contour-based
roads as, by Michell’s theorem, the orientations followed the contours of the density paths. Mechanical testing of optimized Messerschmitt-Bölkow-Blohm (MBB) beams demonstrated a 30% improvement in sustained compressive load by the CFAO specimens relative to a non-reorientable orthotropic material model [34]. Bodetti et al. used an extension of the CFAO algorithm to create geometries for a planar multi-material jetting AM process [37]. The microstructure of each element (consisting of a two-phase short-fiber composite) was oriented within each element using two design variables to enable non-planar orientations. To create a toolpath, the microstructure was voxelized along the build direction such that the fibers were printed across multiple layers.

The CFAO method has issues with local minima though, stemming from the Euler angle parameterization of the orientation design space [71]. To address this issue, Nomura et al. used two orientation design variables to represent the planar orientation of each element and mapped a natural coordinate system onto a physically meaningful orientation space [71]. While this is not a minimum representation of the orientation space (i.e., the parameterization requires additional design variables per element to represent the orientation), the method was shown to have little issue with converging to local minima. Discrete Material Optimization (DMO) also does not use a direct Euler angle variable, but instead uses a weighted sum of discrete material orientations to reduce issues with local minima [76, 77, 78]. Free Material Optimization (FMO) optimizes for the values of the material properties directly, rather than the material orientation, which enables significant freedom in material design [79, 80].

Although there are a number of TO formulations that enable the variation of material orientation, not all of them are suitable for application to the ME process. For instance, FMO allows for the infinite variation of material properties throughout the design space, but fabrication of the results may not be feasible. The ME process is often limited to the deposition of a single material, which would require constraints to be placed on the FMO algorithm in order to produce a realizable result. On the other hand, DMO relies on a set number of allowable material orientations, which restricts orientation variation. As seen in [34], optimal roads are typically highly organic curves that would be difficult to capture with a predetermined set of allowable orientations. Additionally, the results of DMO algorithms are dependent on the chosen weighting function [76] and require unit weight constraints to ensure physically realistic results [77]. Therefore CFAO and the extension shown in [71], which allow for the infinite variation of a single set of material properties, are used as the basis for the presented algorithm. There are a number of parameterizations for 3D orientations though, and as discussed in [71], the chosen parameterization has a large impact on both the efficiency of the algorithm and the quality of the final result.
2.1 Research Gap

To fully leverage the capabilities of LL-MA ME, material alignment must be varied in 3D, which requires treating the 3D orientation of the material in each element as a design variable. Due to the non-convexity of these orientation spaces, the parameterization chosen for the optimization process has a large impact on the performance of the algorithm. Specifically, singularities and local minima are large issues in orientation optimization, both for the final solution fitness and the speed of convergence [81]. To this end, three parameterizations of the orientation design space are explored in this work: i) Euler angles, ii) quaternions, and iii) natural quaternions. An overview of each parameterization is given in Section 2.2, and a general problem statement that is applicable to each orientation parameterization is given in Section 2.3 (with parameterization-specific problem statements given in Appendix A). Each parameterization is demonstrated using two benchmark compliance minimization problems, the MBB beam and the Wheel problem, and a combined loading case featuring i) pure tension and ii) three-point bending (Section 2.4). The parameterizations are compared in terms of their final solution fitness and the number of iterations required for convergence. As a point of comparison to existing TO algorithms, a CFAO algorithm is also used for each problem. It is hypothesized that i) for the 2D loading cases, the parameterizations will produce equivalently compliant structures to the CFAO algorithm and ii) the 3D parameterizations will result in improved compliance values for 3D loading conditions, relative to the CFAO algorithm, due to the increased orientation flexibility. A summary of the contribution is then presented in Section 2.5.

2.2 3D Orientation Parameterizations

Many different parameterizations exist for representing orientations including Euler angles, axis-angle parameters, direction cosines, and quaternions [82]. In the context of an optimization problem, the choice of parameterization has a large effect on the efficiency of the algorithm as well as the quality of the final solution. Three different parameterizations, described in the following subsections, are explored in this work in an effort to find an effective and efficient set for TO with considerations for 3D material orientations. While this work focuses on explicit control of the orientation field through design variables, Appendix B explores coupling the orientation field to the displacement field.

2.2.1 Euler Angles

CFAO uses a planar simplification of the Euler angle parameterization and has been extended to non-planar orientations, but both CFAO and the implementation in [37] are insufficient to describe the orientation variation possible with LL-MA ME. Specifically, the jetting process
2.2. 3D Orientation Parameterizations

used in [37] only requires two Euler angles to properly describe, as it has no ability to reorient material properties orthogonal to the fiber direction. In contrast, the LL-MA ME process does have that flexibility and therefore requires three Euler angles.

In the Euler angle parameterization, a rotation angle and axis defines an individual rotation, and the concatenation of three rotations can produce any desired orientation [82]. Although a minimum representation of orientation, Euler angles have issues with numerical stability and the poorly shaped design space is difficult to optimize, often resulting in convergence to local minima [83]. In order to represent any orientation with three rotations, sequential rotations cannot act on the same axis. This results in 12 possible sets of rotation angles that can be used; in this work, the Y-Z-X set was selected, but similar performance is expected from all of the sets. The corresponding rotation matrix ($R$) is shown in Equation 2.1 where $\theta$, $\phi$, and $\psi$ are the rotation angles about axes Y, Z, and X, respectively. $C(y)$ and $S(y)$ are used as shorthand for $\cos(y)$ and $\sin(y)$, respectively.

$$
R(\theta, \phi, \psi) = \begin{bmatrix}
C(\theta) & 0 & S(\theta) \\
0 & 1 & 0 \\
-S(\theta) & 0 & C(\theta)
\end{bmatrix} \times \begin{bmatrix}
C(\phi) & -S(\phi) & 0 \\
S(\phi) & C(\phi) & 0 \\
0 & 0 & 1
\end{bmatrix} \times \begin{bmatrix}
1 & 0 & 0 \\
0 & C(\psi) & -S(\psi) \\
0 & S(\psi) & C(\psi)
\end{bmatrix}
$$

(2.1)

2.2.2 Quaternions

Originally developed by Hamilton, a quaternion (shown in Equation 2.2) is a noncommutative 4-tuple [84], and unit length quaternions can be directly related to the axis-angle representation of a rotation [82]. When compared to Euler angles, quaternions are more numerically stable for denoting rotations as they do not have singularity issues [85]. They are also a symmetric representation of rotation (i.e., each component of the quaternion equally influences the resulting rotation) [81]. In contrast, the cascading effect of the Euler angle representation creates imbalanced contributions from each angle. Additionally, quaternions only require algebraic operations to calculate rotation matrices, instead of trigonometric functions, which decreases computation time [86]. A rotation matrix can be calculated from a quaternion using Equation 2.3 [85].

$$
q = \{q_w, q_x, q_y, q_z\}^T
$$

(2.2)

$$
R(q) = \begin{bmatrix}
q_w^2 + q_x^2 - q_y^2 - q_z^2 \\
2(q_xq_y + q_wq_z) \\
2(q_xq_z - q_wq_y)
\end{bmatrix} \times \begin{bmatrix}
q_x^2 - q_y^2 + q_z^2 \\
2(q_xq_y - q_wq_z) \\
2(q_xq_z + q_wq_y)
\end{bmatrix} \times \begin{bmatrix}
q_y^2 - q_z^2 + q_w^2 \\
2(q_yq_z - q_wq_x) \\
2(q_yq_x + q_wq_z)
\end{bmatrix}
$$

(2.3)

In order to represent a pure rotation, the quaternion must be of unit length. Otherwise, when executing the TO algorithm, the elastic matrix will be scaled during reorientation,
artificially changing the properties of the element. To prevent this, a unit length constraint must be imposed on each element in the design space. This requires a large number of additional constraints, increasing the computational time required for the optimization [87].

2.2.3 Natural Quaternions

In order to remove the explicit unit length constraint on each element in the design space, the parameter space can be relaxed to a natural coordinate system. Nomura et al. presented a method of mapping a 2D natural coordinate space to a planar orientation [71]. This can be applied to the quaternion using Equation 2.4, where the natural coordinate system (w, x, y, and z) is mapped onto the constrained coordinate system (q_w, q_x, q_y, and q_z).

\[
\begin{bmatrix}
q_w \\
q_x \\
q_y \\
q_z \\
\end{bmatrix} = \begin{bmatrix}
w\sqrt{1 - \frac{x^2 + y^2 + z^2}{2}} + \frac{x^2 y^2 + x^2 z^2 + y^2 z^2}{3} - \frac{x^2 y^2 z^2}{4} \\
x\sqrt{1 - \frac{w^2 + y^2 + z^2}{2}} + \frac{w^2 y^2 + w^2 z^2 + y^2 z^2}{3} - \frac{w^2 y^2 z^2}{4} \\
y\sqrt{1 - \frac{w^2 + x^2 + z^2}{2}} + \frac{w^2 x^2 + w^2 z^2 + x^2 z^2}{3} - \frac{w^2 x^2 z^2}{4} \\
z\sqrt{1 - \frac{w^2 + x^2 + y^2}{2}} + \frac{w^2 x^2 + w^2 y^2 + x^2 y^2}{3} - \frac{w^2 x^2 y^2}{4}
\end{bmatrix} \tag{2.4}
\]

By limiting the values of w, x, y, and z between -1 and 1, the resulting quaternion is restricted to have a magnitude of ≤ 1. Although not a minimum parameterization, this method implicitly enforces the unit length constraint and only requires side constraints. While this does allow quaternions of less than unit length, elements in this state would be artificially weaker. The optimization algorithm therefore naturally drives the quaternion to unit length, as that is the state of minimum compliance.

2.3 3D Material Orientation Problem Statement

The CFAO problem statement (shown in Equation 2.5) is stated to minimize the compliance of a structure. This is accomplished through a simultaneous optimization of material distribution and orientation, using two design variables for each element in the design space: i) a pseudo-density and ii) an Euler angle. Each pseudo-density scales the elastic matrix, defined by the material being printed, between solid and void, and each Euler angle rotates
2.3. 3D Material Orientation Problem Statement

the scaled matrix corresponding to the optimized deposition direction.

$$\min_{\rho, \theta} \ c(\rho, \theta) = \sum_{k=1}^{N_{lc}} U_k^T K(\rho, \theta) U_k$$

subject to:

$$\frac{V(\rho)}{V_0} \leq f$$

$$: K(\rho, \theta) U_k = F_k$$

$$: 0 < \rho_{min} \leq \rho \leq 1$$

$$: -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$$

(2.5)

$U_k$ is the global displacement vector associated with load case $k$, $K(\rho, \theta)$ is the global stiffness matrix as defined in Equation 2.6, and $N_{lc}$ is the number of load cases experienced by the structure. $V(\rho)$ is the mass of the current solution and $V_0$ is the mass of a fully dense design space. The ratio of the two calculates a volume fraction, which must be less than or equal to a maximum allowable volume fraction $f$. $F_k$ is the forcing vector associated with load case $k$ acting on the design space and $\rho_{min}$ is the minimum allowable element pseudo-density.

$$K(\rho, \theta) = \sum_{i=1}^{N_e} \rho_i^0 L_i^T \int_{\Omega_i} \int_{\Omega_i} (B^T T_i^T(\theta_i) E_0 T_i(\theta_i) B) \partial \Omega_i L_i$$

(2.6)

$N_e$ is the number of elements in the design space, and $\Omega_i$ is the volume of element $i$. $L_i$ is the locator matrix that places the element stiffness matrix into the global stiffness matrix, and $B$ is the strain-displacement matrix for the element. The strain transformation matrix $T_i(\theta_i)$ can be written in terms of the rotation matrix $R_i(\theta_i)$ using Equation A.2 (given in Appendix A, from [88]). $E_0$ is the matrix of elastic constants describing the fully dense material, and $\eta$ is the SIMP penalty factor.

The configuration of design variables used by CFAO can be extended to allow for 3D orientations by changing the definition of the transformation matrix in Equation 2.6 to incorporate a 3D rotation instead of a strictly XY-planar one. Although three different 3D orientation parameterizations are discussed in this work, the general problem statement remains the same. The design variable vector can be partitioned into pseudo-density design variables, $\rho$, and orientation design variables, $Q$ (Equation 2.7). The corresponding full problem statement is shown in Equation 2.8 and the definition of the global stiffness matrix is given in Equation 2.9. The details of $Q$ change depending on the chosen orientation parameterization as discussed in Section 2.2. The specific problem formulation for each orientation
parameterization is provided in Appendix A.

\[
x = \{ \rho^T, \; Q^T \}^T
\]

\[
\min_x : c(x) = \sum_{k=1}^{N_{lc}} U_k^T K(x) U_k
\]

subject to:

\[
\frac{V(\rho)}{V_0} \leq f
\]

\[
K(x) U_k = F_k
\]

\[
0 < \rho_{min} \leq \rho \leq 1
\]

\[
Q^- \leq Q \leq Q^+
\]

\[
K(x) = \sum_{i=1}^{N_e} \rho_i^B L_i^T \iint_{\Omega_i} (B^T T_i^T(Q_i) E_0 T_i(Q_i) B) \partial \Omega_i L_i
\]

\( Q^- \) and \( Q^+ \) are the lower and upper bounds on the orientation design variables, respectively. \( \rho_i \) and \( Q_i \) are the pseudo-density and the vector of orientation variables associated with element \( i \), respectively. The specific form of \( R_i(Q_i) \), and consequently \( T_i(Q_i) \), is determined by the selected orientation parameterization as described in Section 2.2.

### 2.3.1 Sensitivity Analysis

Gradients of the compliance objective function are calculated using two different equations depending on the type of variable: i) pseudo-density design variables (Equation 2.10) and ii) orientation design variables (Equation 2.11). Regardless of the orientation parameterization used, the general forms of the gradients remain the same. As an example, specific gradient equations are provided for the Euler angle parameterization in Appendix A. Due to the decoupled nature of the elements in the design space, the sensitivity calculations can be performed at the element level.

\[
\frac{\partial c}{\partial \rho_i} = -\eta \rho_i^{\eta-1} \sum_{k=1}^{N_{lc}} u_{k,i}^T \iint_{\Omega_i} (B^T T_i^T(Q_i) E_0 T_i(Q_i) B) \partial \Omega_i u_{k,i}
\]

\[
\frac{\partial c}{\partial Q_{i,j}} = -\rho_i \sum_{k=1}^{N_{lc}} u_{k,i}^T \iint_{\Omega_i} (B^T \frac{\partial T_i(Q_i)}{\partial Q_{i,j}} E_0 T_i(Q_i) + T_i(Q_i)^T E_0 \frac{\partial T_i(Q_i)}{\partial Q_{i,j}} B) \partial \Omega_i u_{k,i}
\]
2.4 Comparison of Orientation Parameterizations on Benchmark Problems

$u_{k,i}$ is the nodal displacement vector associated with element $i$ for load case $k$. $Q_{i,j}$ is contained within the vector of orientation design variables associated with element $i$, $Q_i$.

2.4 Comparison of Orientation Parameterizations on Benchmark Problems

Three example problems were used to validate the presented TO formulation and make comparisons between the three orientation parameterizations described in Section 2.2 in terms of i) iterations to convergence, ii) final compliance, iii) and mesh convergence. For comparison, a CFAO algorithm is also executed for each example problem.

A 2.5D MBB beam structure [89] is used to validate the functionality of the presented formulation. The load case is shown in Figure 2.1, leveraging the symmetry around the central XZ-plane to reduce the number of required elements. Due to this planar loading case, the 3D orientation parameterizations are not expected to outperform the CFAO algorithm. Additionally, the Euler angle parameterization is expected to have less difficulty with local minima and converge to a similar result as the quaternion parameterizations.

![Figure 2.1: (a) Diagram of the MBB beam problem and (b) Model taking advantage of the symmetry across the XZ-plane.](image)

The 3D Wheel problem [90] is used to compare the parameterizations in terms of their ability to follow 3D load paths. The load case is shown in Figure 2.2, and symmetry is leveraged across both the central XZ- and YZ-planes. The aspect ratio and the fixed corner nodes create a non-planar load case. As such, the 3D orientations are expected to demonstrate significant improvement in compliance over the CFAO algorithm, and the quaternion parameterizations are expected to converge in fewer iterations than the Euler angle parameterization.

A multi-loading case is then used to evaluate the algorithm’s performance in regards to optimizing multi-loaded structures with fixed-density regions. The first loading case is pure tension along the Y-axis and the second is three-point bending in the YZ-plane as shown in Figure 2.3. Fixed density regions (marked in gray) are allotted in anticipation of test fixturing; the algorithm is only able to optimize the orientation in those regions.
Chapter 2. Topology Optimization with 3D Material Orientations

Figure 2.2: (a) Diagram of the Wheel problem and (b) Model taking advantage of the symmetry across the XZ-plane.

Figure 2.3: (a) Diagram of the pure tension load case and (b) Diagram of the three-point bend load case. The grey regions marked in each diagram represent fixed-density regions, where the TO algorithm only optimizes for orientation.

2.4.1 Implementation

The presented TO algorithm was implemented in MATLAB® 2018a, using the Method of Moving Asymptotes (MMA) [91] as the optimizer. To prevent mesh instabilities (e.g., mesh dependency and checkerboard patterning [92]) and to produce nearly binary solutions, the Heaviside projection method (HPM) [93] was implemented for the pseudo-density design variables. As suggested in [94], a few modifications were made to the default parameters of the MMA algorithm to create a more conservative subproblem: move was decreased from 0.5 to 0.3, asyinit was changed from 0.5 to $0.5/\beta + 1$ (as suggested in [93]), and asyincr was decreased from 1.2 to 1.1. Finally, due to the highly curved orientation design spaces, the minimum asymptote multiplier was decreased from 0.01 to $2.5 \times 10^{-4}$ to allow the optimizer to move closer to the local minimum.

The initial conditions for the element orientations were driven by the CFAO algorithm to allow orientation variation in the YZ-plane and were kept constant between the different parameterizations. The pseudo-density of each element was initialized to the allowable volume fraction, 0.2; in the multi-load case, the fixed-density elements were set to 0.5. The continuation method was also implemented to promote convergence to the global minimum [95]. In this work, the SIMP penalty factor was iteratively increased from 1 to 3 by an increment of 0.1 after the MMA algorithm executed 30 iterations; each 30 iterations is referred to as a continuation cycle. A termination criteria using the first-order necessary condition was created using a tolerance of $10^{-4}$, but while all problems converged, none satisfied that criteria. The Heaviside filter radius ($r_{min}$) is set to 1.4 voxels relative to the size of the coarsest mesh.
Table 2.1: Shared parameters for the test case examples

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>0.2</td>
<td>Allowable volume fraction</td>
</tr>
<tr>
<td>$L$</td>
<td>1</td>
<td>Point load magnitude</td>
</tr>
<tr>
<td>$r_{\text{min}}$</td>
<td>1.4</td>
<td>Heaviside density filter radius</td>
</tr>
<tr>
<td>$\beta$</td>
<td>2</td>
<td>Heaviside approximation curvature parameter</td>
</tr>
<tr>
<td>$E_{1,1}$</td>
<td>10</td>
<td>Young’s modulus for deposition direction</td>
</tr>
<tr>
<td>$E_{2,2}$</td>
<td>4</td>
<td>Young’s modulus for intra-layer bonds</td>
</tr>
<tr>
<td>$E_{3,3}$</td>
<td>1</td>
<td>Young’s modulus for inter-layer bonds</td>
</tr>
<tr>
<td>$v$</td>
<td>0.35</td>
<td>Isotropic Poisson’s ratio</td>
</tr>
<tr>
<td>$G$</td>
<td>1.4</td>
<td>Isotropic shear modulus</td>
</tr>
</tbody>
</table>

and scales linearly as the mesh is refined. Other optimization parameters are described in Table 2.1, and the mesh sizes used to evaluate convergence for each problem are given in Table 2.2.

### 2.4.2 Material Properties

As shown in Figure 1.4, the tensile properties resulting from LL-MA ME are independent of the build direction [26]. That is to say, a material deposition at any arbitrary orientation will have the same mechanical performance as one printed in the XY-plane (i.e., by more typical 3-DoF deposition). Therefore, the elastic matrix does not have any orientation dependence; the deposition direction within an element only changes the orientation of the elastic matrix in the global coordinate system, not the value in the local coordinate system. In this work, an orthotropic elastic matrix is used to model the properties resulting from the LL-MA ME process, which is kept constant throughout the example problems. The matrix is derived from the material properties listed in Table 2.1.

### 2.4.3 Visualizations

The output from the TO algorithm is a discretized geometry, where each element has an associated pseudo-density and orientation. Due to the information density, two plotting techniques are leveraged as described below. For visual clarity, elements with a pseudo-density of $\rho_i < 0.25$ are not displayed in either visualization.

For small numbers of elements, it is useful to display all of the information together, as shown in Figure 2.4. The shading represents the relative density, where darker shading represents
more solid elements, and the orientations are displayed using colored vectors within each element. Black elements typically represent fully dense regions, but in this work, the element shading is scaled such that fully dense elements are represented by a dark grey (as seen in the center of the main truss of each solution in Figure 2.6). This is done to promote visibility of the material orientations. Red arrows denote the deposition ($E_{1,1}$) direction, green arrows denote the intra-layer bond ($E_{2,2}$) direction, and blue arrows denote the inter-layer bond ($E_{3,3}$) direction.

For larger numbers of elements, the visualization of the orientation within each element is too small to be useful. Therefore, in 2D meshes, the deposition direction ($E_{1,1}$) is displayed by coloring the corresponding pixel according to Figure 2.5c. These plots are used to demonstrate mesh convergence in the 2D meshes; density is not shown due to the nearly binary solution space of the refined meshes. An example of this visualization is shown in Figure 2.5 using a set of nine elements. Note that the color legend is mirrored across the Y-axis, as for example, a deposition direction aligned to the positive X-axis is equivalent to the deposition direction aligned to the negative X-axis.

### 2.4.4 MBB Beam Results

The MBB beam results for each parameterization (at the coarsest mesh size) are shown in Figure 2.6. With the exception of the Euler angle parameterization, each of the resulting topologies are qualitatively similar regardless of the orientation parameterization. The main truss leading from the point of load to the support occupies the majority of the allowed volume fraction, and similar supporting features appear beneath the main truss. The Euler angle parameterization deviates from this design with an additional secondary supporting truss but with a reduced volume fraction allotted to each of those supporting trusses. For each parameterization, the $E_{1,1}$ direction follows the truss direction (as predicted by Michell’s truss theorem [65]), and the $E_{3,3}$ direction is largely removed from the load paths.

The objective function evaluations, plotted in Figure 2.7, demonstrate that the 3D material orientation parameterizations produced structures with similar final compliance values to
2.4. Comparison of Orientation Parameterizations on Benchmark Problems

Figure 2.4: Sample output of four elements from (a) overhead and (b) isometric viewpoints. The shading of the element denotes the density and the colored vectors denote the material orientation. (c) A planar example of the output showing deposition directions (red) and build directions (blue) and (d) example propagated roads with an example deposition head along the respective build direction.

Figure 2.5: Sample output of nine elements colorized to represent the deposition direction. (a) the voxels visualized as described in Figure 2.4, (b) the corresponding pixels colorized to represent the \( E_{1,1} \) direction, and (c) the color legend.
Figure 2.6: YZ-plane views of the MBB beam results using the CFAO algorithm and 3D parameterizations. (a) CFAO, (b) Euler angles, (c) Quaternion, and (d) Natural quaternion
the CFAO algorithm. This similarity is largely attributed to the problem’s planar loading case, which does not leverage the advantages of a 3D material orientation. The increase in the penalized evaluation of the Euler angle design is likely due to the increased use of intermediate densities in the secondary truss structures.

Figure 2.7: Final objective function value comparison between the different orientation parameterizations for the MBB beam problem (1x60x20 mesh). CFAO: \( c_{\eta=3} = 25.077, \ c_{\eta=1} = 18.031 \); Euler angles: \( c_{\eta=3} = 26.203, \ c_{\eta=1} = 18.283 \); Quaternion: \( c_{\eta=3} = 24.836, \ c_{\eta=1} = 18.147 \); Natural quaternion: \( c_{\eta=3} = 24.964, \ c_{\eta=1} = 18.146 \)

Although all of the parameterizations arrived at similar final compliance values, the rate of convergence differed between the parameterizations during optimization. The first three continuation cycles are shown in Figure 2.8. As evidenced in the figure, the Euler angle parameterization converged the slowest on each continuation cycle, requiring approximately eight additional iterations to converge within 1% of the final objective function value. After the first cycle though, the CFAO parameterization also demonstrated difficulties during convergence relative to the quaternion parameterizations, requiring approximately two additional iterations to converge. This slower convergence is likely due to the poorly shaped design spaces of the CFAO and Euler angle parameterizations, as described in [83]. Even though the quaternion parameterizations require additional design variables to denote orientations, the additional iterations required to converge at each continuation cycle would increase computation time for large optimization problems.

In terms of mesh convergence, the results of the remaining mesh sizes for the natural quaternion parameterization are plotted in Figure 2.9. For the sake of brevity, the other orientation parameterizations are not presented, as each parameterization converged to the same general design (i.e., the additional secondary truss used by the Euler angle parameterization in Figure 2.6b is removed from the design with mesh refinement). That said, due to the unit length constraints required in the explicit quaternion parameterization, the memory requirements for mesh sizes beyond 1x120x40 exceeded computational limitations (as discussed further in Section 2.4.7).
2.4.5 Wheel Results

Solutions for the Wheel problem using each parameterization (at the coarsest mesh) are shown in Figure 2.10. For visual clarity, only the elements on the outer surface of each structure are displayed. Qualitatively, the results of the 3D orientation parameterizations have similar geometric features; each contains curved features moving from the point of support to the point of load at the intersection of the planes of symmetry. While the 3D parameterizations produced symmetric final structures, as expected from the double-symmetric load case, the CFAO algorithm preferentially distributed material within the plane of variation (the YZ-plane). This deficiency is also reflected in the final compliance values, as shown in Figure 2.11; while the quaternion parameterizations achieved similar final values, the CFAO algorithm produced a structure with a 38% increased final compliance relative to the natural quaternion result. The Euler angle parameterization also demonstrated difficulty with this loading case, producing a structure with 24% increased compliance. This is likely due to the orientations converging to local minima.

The decrease in the CFAO algorithm’s performance is due to the inflexibility of the algorithm; the deposition directions are unable to accurately follow 3D load paths. To illustrate this, cross-sections of the CFAO result are shown in Figure 2.12 relative to the same cross-sections in the natural quaternion result. Cross-sections are taken in both the YZ-plane, where the CFAO algorithm allows deposition direction variation, and the XZ-plane, where no variation is allowed. The cross-sections are displayed as individual layers of elements along that axis (e.g., the 10th cross-section in the YZ-plane is the 10th layer of elements along the X-axis).
Figure 2.9: The results of the MBB beam load case using the natural quaternion parameterization at a mesh size of: (a) 1x120x40, (b) 1x240x80, (c) 1x480x160, and (d) 1x960x320. Mesh coloring is performed according to the legend shown in Figure 2.5c.
Figure 2.10: Isometric views of the final topologies from CFAO and the 3D orientation parameterizations. (a) CFAO, (b) Euler angles, (c) Quaternion, and (d) Natural quaternion.
2.4. Comparison of Orientation Parameterizations on Benchmark Problems

Differences do appear between the CFAO algorithm and the natural quaternion cross-sections, but the main density features are similar. As shown in Figure 2.12, the main factor in the objective function discrepancy comes from the CFAO algorithm’s inability to follow the 3D load paths. The CFAO algorithm demonstrates $E_{1,1}$ alignment with the density paths in the YZ-plane, as expected by Michell’s theorem [65]. In the XZ-plane though, the $E_{1,1}$ directions do not accurately follow the density paths. This is in contrast to the cross-sections of the natural quaternion result, which display $E_{1,1}$ alignment with the density paths in both planes.

The Euler angle and CFAO parameterizations also required an increased number of iterations to converge, as shown in Figure 2.13. Although the explicit quaternion took nearly the full 30 iterations to converge on the first cycle, the Euler angle and CFAO parameterizations required approximately five additional iterations to converge on subsequent cycles than the explicit and natural quaternion parameterizations. The increased number of iterations, coupled with the oscillating objective function evaluations, demonstrate a difficulty converging.

In terms of mesh convergence, the final objective function evaluations for each of the parameterizations are plotted in Figure 2.14 against the number of elements in the design space. Due to computational requirements, discussed in Section 2.4.7, the explicit quaternion parameterization was only able to be executed for the coarsest mesh size (15x15x15 or 3375 elements). It is also important to note that the single data point for the explicit quaternion parameterization directly overlaps with the natural quaternion parameterization. As seen, the final compliance values for the other parameterizations do converge by the finest mesh size, but the natural quaternion produces the structure with the minimum compliance at each mesh size.
Figure 2.12: Select cross-sections from the final Wheel topologies using CFAO and the natural quaternion parameterization. Cross-sections are taken along the YZ-plane and XZ-plane to demonstrate $E_{1,1}$ alignment to the density features in a single plane with CFAO and in both planes with the natural quaternion parameterization.
2.4. Comparison of Orientation Parameterizations on Benchmark Problems

Figure 2.13: Convergence history of each orientation parameterization for the Wheel problem.

Figure 2.14: Plot of the final compliance values of the Wheel problem for each orientation parameterization against the number of elements in the design space. The quaternion parameterization could only be run for the coarsest mesh size due to computational requirements and is directly overlapped by the natural quaternion result.
2.4.6 Multi-Load Results

The multi-load case is used to evaluate the algorithm’s mesh convergence for a complex loading condition. Figure 2.15 shows the structures produced using the natural quaternion result; for the sake of brevity, Figure 2.16 only shows the most refined meshes for each of the other three parameterizations. As seen in the natural quaternion results, the main features of the geometry do not change with increasing mesh refinement, but the figures do demonstrate a settling of the secondary features between Figure 2.15b and 2.15c. Although there is some variation in the optimized orientations between Figure 2.15c and 2.15d, the final compliance values (plotted in Figure 2.17) show little variation after the first mesh refinement.

There are qualitative differences between geometric features in Figure 2.16; while each parameterization features the same main density features, the quaternion result does not exhibit other secondary features. Although the placement of those supporting features varies between the CFAO, Euler angle, and natural quaternion (Figure 2.15), the main difference in final compliance lies in the optimized orientations. The CFAO and Euler angle results seem to have converged to local minima, exhibited by the unaligned orientations near the bottom of the two results. The gradients of the corresponding design variables are below $10^{-10}$, well below the convergence criteria, but the orientations themselves do not agree with expectation (i.e., Michell’s truss theorem).

In the case of the explicit quaternion result, the increased compliance relative to the natural quaternion result is attributed to the more restricted orientation design space. During each iteration, the MMA algorithm can only make a small step in the orientation design space in order to maintain the unit length constraint. As such, the density features are likely developed ahead of the orientation field, limiting its optimality.

2.4.7 Computational Requirements

The optimization time also needs to be considered when dealing with large optimization problems. As shown in Figure 2.18, the parameterizations each have similar computation times relative to the number of elements in the design space with the exception of the explicit quaternion parameterization. For the coarsest Wheel mesh of 3375 elements, the CFAO, Euler angle, and natural quaternion parameterizations took under 30 minutes to complete. The inclusion of explicit unit length constraints in the quaternion parameterization saw the optimization time increase to 32.8 hours. These times are not included for the purposes of benchmarking but as an order-of-magnitude comparison.

In addition to increasing the computation time, the inclusion of unit length constraints in the quaternion parameterization also increases the memory requirements. During the execution of the MMA algorithm, multiple non-sparse matrices of size number of constraints by number of design variables must be created. These matrices are therefore of size $(N_e + 1) \times (5N_e)$ for the explicit quaternion parameterization and exceed the memory limits for all
Figure 2.15: The results of the multi-load case using the natural quaternion parameterization at a mesh size of: (a) 1x120x40, (b) 1x240x80, (c) 1x480x160, and (d) 1x960x320. Mesh coloring is performed according to the legend shown in Figure 2.5c.
Figure 2.16: The results of the multi-load case using: (a) CFAO with a mesh size of 1x960x320, (b) Euler angles (1x960x320), and (c) Quaternions (1x120x40). Mesh coloring is performed according to the legend shown in Figure 2.5c.
2.4. Comparison of Orientation Parameterizations on Benchmark Problems

Figure 2.17: Plot of the final compliance values of the multi-load case for each orientation parameterization against the number of elements in the design space. The quaternion parameterization could only be run for the coarsest mesh size due to computational requirements.

Figure 2.18: The computation times for each parameterization plotted against the number of elements in the design space (all load cases are shown together). A linear fit is shown to aid in visualizing the effect of the parameterization. All of the meshes that were able to be completed are plotted; the explicit quaternion parameterization was only able to complete coarse meshes due to large memory and computation time requirements.
of the tested meshes that had more than 5000 elements. For large optimization problems, the increased computation time and memory requirement render the explicit quaternion parameterization infeasible.

2.5 Summary of Contribution

This chapter presented a method for the simultaneous optimization of material distribution and orientation in full 3D to enable LL-MA ME by extending a CFAO algorithm to optimize for 3D orientation design spaces. CFAO has previously been demonstrated to be effective for planar orientation variation [72, 73] and for non-planar fiber orientation [37], but these methods do not have enough flexibility in the orientation design space to properly model the LL-MA ME process. While the distribution of design variables used in CFAO (i.e., one set defining the material distribution and another defining the orientation) can scale up to 3D, the selection of the orientation parameterization has a large impact on the performance and efficiency of the TO algorithm. Therefore, three parameterizations of 3D orientations were explored: i) Euler angles, ii) quaternions, and iii) natural quaternions. These parameterizations were compared through two established minimum compliance example problems, the MBB beam and the Wheel problem, as well as a multi-load case featuring pure tension and three-point bending. The final objective functions, iterations to convergence, and mesh convergence were compared for each parameterization, using the CFAO algorithm as a baseline.

The MBB beam problem featured a planar load case, and the plane of allowed orientation variation for the CFAO algorithm was chosen to be in the same plane as the load case. As hypothesized, all three of the 3D parameterizations and the CFAO algorithm arrived at similar final objective function values (Figure 2.7). The final topologies qualitatively agreed with literature by showing the characteristic density paths, and the strongest directions ($E_{1,1}$) aligned between the point of load and points of support.

The Wheel problem was a true 3D mesh and loading case, using (at the coarsest) a mesh of 15x15x15 elements. In this case, the CFAO algorithm was only able to follow the YZ-component of the load paths, as that was the allowable plane of variation. In contrast, the 3D orientation parameterizations demonstrated significant improvement in $E_{1,1}$ alignment between the point load and supporting locations. The final objective function values of the quaternion parameterizations demonstrated an approximately 38% improvement over that of the CFAO algorithm (Figure 2.11). The Euler angle parameterization also demonstrated difficulties relative to the quaternion parameterizations, converging to a solution with 24% increased compliance. In this case, the increased compliance is attributed to convergence to a local minima in the orientation design space.

In the multi-load case, the Euler angle and the CFAO algorithm produced similarly compliant structures, but the natural quaternion parameterization outperformed all of the other
parameterizations of the orientation design space. This is attributed to i) the Euler angle and CFAO parameterizations converging to local minima and ii) the inflexibility of the explicit quaternion parameterization. There is a limited range of motion during the optimization of a unit length quaternion, which restricts flexibility at each iteration.

The 3D parameterizations also saw differentiation in terms of the number of iterations required to converge to a solution; the Euler angle parameterization required approximately six additional iterations for each continuation cycle relative to the explicit and natural quaternion parameterizations. Additionally, the quaternion parameterization required the inclusion of a unit length constraint for each element in the design space. This sharply increased the computation time; while the other parameterizations took less than 30 minutes each to finish the 3D Wheel problem with a 15x15x15 mesh, the quaternion parameterization took approximately 32.8 hours. As the mesh is refined, the computation time and memory requirements of the quaternion parameterization became prohibitive.

2.5.1 Integration into the Layer-less Multi-Axis Workflow

The presented TO algorithm solves the simultaneous material distribution and orientation problem for input loading and boundary conditions. With the ability to follow 3D load paths, the algorithm can be used to generate the geometry and orientation field inputs to the presented LL-MA toolpath planning algorithm. The natural quaternions (more specifically the rotation matrix they represent) are directly related to the optimal deposition directions and deposition head orientations for the input loading and boundary conditions.

In the context of the LL-MA ME process, the results of the TO algorithm are not necessarily manufacturable. Specifically, the deposition head can take any orientation in space, imposing complex kinematic constraints on manufacturability; the deposition head could collide with the environment (e.g., the build platform) or the print in-progress. In particular, collisions with the build platform can be predicted as shown in Figure 2.19. If the sum of the angle between the build direction and the global Z-axis ($\theta_B$) and the angle of the deposition head collision volume ($\theta_T$) exceeds $\pi/2$, the deposition head will collide with the build platform.

More on this issue and future plans for accommodation are described in Section 8.4, but as an initial effort to improve the printability of the results, a penalization function is imposed on the elastic matrix for elements with steep build directions. The modified mapping of design variables to the element’s elastic matrix is shown in Equation 2.12. The Heaviside approximation function (similar to the one used in HPM) is used here; if the build direction creates the scenario shown in Figure 2.19b, the elastic matrix is near its nominal value (i.e., the penalty $p(Q_i) = 1$). If $\theta_B$ becomes too large though, $p(Q_i) = 0$. This artificially reduces the stiffness of the element when it tries to achieve steep build directions, forcing
Figure 2.19: (a) The deposition head, with interference angle $\theta_T$, is perpendicular to the build platform, (b) the deposition head is at an angle $\theta_B$ but does not collide with the build platform, and (c) if $\theta_B$ gets too large, the deposition head collides with the build platform.

The algorithm away from those solution spaces.

$$E_i(\rho_i, Q_i) = p(Q_i)\rho_i^n T_i(Q_i)E_0(\rho_i, Q_i)T_i(Q_i)$$  \hspace{1cm} (2.12)
Support structure in the ME process serves two purposes: i) to physically support recently deposited features for stability during cooling and ii) to give the deposition head a substrate onto which material can be deposited. Although the exact strategy for generating support structure for XY-planar printing may vary, the general concept is the same; the geometry is projected into the XY-plane (i.e., the build platform) and the volume between the model and that XY-projection is filled with support structure. This can result in a large amount of material required for the supports (relative to the part geometry), which is discarded as waste after the print. In efforts to both reduce print time and material waste, a number of techniques have been presented in literature. For instance, Vanek et al. use the critical (i.e., self-supporting) angle of the support material to produce self-supporting structures underneath overhangs [96]. As a result, the support material required by a print was reduced by an average of 40%. Hu et al. reduce the amount of support structure used by optimizing the part orientation and examining the critical angle in local regions [97]. Hongyao et al. use a segmented substrate to raise the build platform in selective regions to reduce the area between the model and the XY-projection [98].

Other works have explored minimizing support material through the optimization of the printed geometry. Gaynor and Guest incorporated considerations for maximum overhang angles in a SIMP algorithm [7, 61]. By changing the neighborhood set in the Heaviside density filter [93] with respect to the maximum overhang angle, they were able to directly obtain self-supporting structures. Langelaar implemented a layer-wise density filter that removed unsupported material from the optimized design space [99]. Although these methods could be implemented into the TO algorithm presented in Chapter 2, these additional constraints have been shown to increase the compliance (and therefore reduce the mechanical performance) of the optimized structures [61].

Multi-axis deposition can also be used to minimize the support structure required during printing. The amount of overhang on a per-layer basis can be reduced by changing the build direction, which reduces or eliminates the need for support structure [25]. By decomposing a part geometry into discrete regions, build directions that minimize this overhang can be selected [100, 9, 25, 101]. Other methods rotate the part underneath the deposition head to change the direction of gravity with respect to the part [102, 103, 10]. These methods
require the selection of specific build directions, which constrains the deposition directions. By constraining the deposition directions, it is likely that the resulting roads will be unable to accurately follow the anticipated load paths.

3.1 Research Gap

The challenges with generating support structure for LL-MA deposition are outlined in [26]. Specifically, the method of propagating support structure along the global Z-direction, as in typical XY-planar deposition does not necessarily result in a viable support structure. As illustrated in Figure 1.6, when the build direction orthogonal to the global Z-axis, the support structure generated using the conventional method does not provide a substrate for deposition. This issue also occurs with a steep build direction relative to the global Z-axis; support material may be generated, but it may not sufficiently provide a substrate for all depositions. Therefore, a support generation algorithm that accommodates variable build directions is presented in Section 3.2. The algorithm is then validated on a quarter-Wheel example in Section 3.3, and a summary of the contribution is presented in Section 3.4.

3.2 Support Generation Algorithm

As previously described, support structure for an XY-planar deposition process is generated by projecting the desired geometry onto the build platform. The LL-MA support generation algorithm (Algorithm 1) generalizes this projection to arbitrary build directions. The optimized orientation field, composed of quaternions, describes the deposition and build directions for each voxel when converted to rotation matrices. At a high level, the algorithm initializes $O$, the list of voxels to check for sufficient support, with all of the voxels in the geometry. Each voxel in $O$ is projected along the negative of its build directions to find regions without material (i.e., void voxels). The voxels contained in those empty regions become support voxels with assigned deposition and build directions. Some of the created support voxels may themselves be unsupported along their assigned build direction, so the process repeats using those recently created support voxels. The process terminates when no additional support voxels were created during the previous iteration. Figure 3.1 shows a 2D example of propagated support material. Additional details on specific subfunctions are given in their corresponding subsections.

3.2.1 Project Node to Find Contacting Voxel

In order to find the voxels required to support a certain voxel $i$, each node of the voxel is projected along the voxel’s associated build direction (Figure 3.1b). If the first voxel
Algorithm 1: Layer-less multi-axis support generation algorithm

Input: \( B, R, T \)

Output: \( B', R', T' \)

1. \( B \): Vector of build directions for the voxels
2. \( R \): Vector of road directions for the voxels
3. \( T \): Vector denoting voxel type (0 = void, -1 = support, 1 = model)
4. \( n \) = number of voxels
5. \( m \) = number of nodes per voxel
6. \( B' = B = [b_i] \)
7. \( R' = R = [r_i] \)
8. \( T' = T = [t_i] \)
9. \( O = [o_i] \) where \( o_i = (t_i = = 1) \forall i = 1, \ldots, n \)
10. \( A = [a_i] \) where \( a_i = \emptyset \forall i = 1, \ldots, n \)

while any(\( O \)) do

12. for (\( i = 1; i <= n; i = i + 1 \) )

13.  if \( o_i \) then

14.      for (\( j = 1; j <= m; j = j + 1 \) )

15.          Project node \( j \) of voxel \( i \) along \(-b'_i\) // Section 3.2.1

16.          \( v = \) number of the first contacted voxel

17.          if \( t'_v = = 0 \) then

18.             Append \( i \) to \( a_v \)

19. \( o_i = FALSE \forall i = 1, \ldots, n \)

20. for (\( i = 1; i <= n; i = i + 1 \) )

21.  if \( a_i \neq \emptyset \) then

22.      Set \( b'_i \) and \( r'_i \) // Section 3.2.2

23.      \( t'_i = -1 \)

24.      \( o_i = TRUE \)

25.      \( a_i = \emptyset \)
contacted during the projection is void, voxel \( i \) is not sufficiently supported as there will be no substrate underneath that node during deposition. Therefore, the first contacted voxel \((v)\) must become a support voxel; this is denoted in \( A \) by appending \( i \) to the vector contained within \( a_v \) (Figure 3.1c). \( A \) is then used to assign new build and road directions to the new support voxels (Figure 3.1d).

![Image](image.png)

Figure 3.1: (a) Model voxels shown in grey with build directions shown in blue, (b) Populate \( O \) with the model voxels (yellow) and project nodes of voxel 11 to find supporting voxels, (c) Defining \( A \) after projecting all voxels in \( O \), (d) Assign build directions and add all newly defined support voxels to \( O \), (e) Repeat projection step, and (f) Fully defined support material for the multi-axis model.

### 3.2.2 Define Road and Build Directions

After defining the region of voxels that must become support structure (denoted by the non-empty entries in \( A \)), the build and road directions of those voxels need to be defined. In order to interface well with the voxels being supported, it is ideal to have the road directions of
the support voxels oriented orthogonal to the road directions of the voxels being supported. This allows the depositions in the model voxels to contact multiple supporting depositions to better maintain deposition shape and positioning. If, for example, the support roads were parallel to the model roads, the model roads might slip in between support roads and make minimal contact with the support. Taking Figure 3.1c as an example, the road direction of voxel 10 should be orthogonal to the road directions of voxels 11 and 15. Although, depending on the road directions of the voxels being supported (i.e., voxels 11 and 15 for voxel 10), it may not be possible to assign a road direction orthogonal to both. In this case, an approximate road direction is calculated as the average of the road directions of the supported voxels; this approximate road direction (\( \tilde{n} \)) is then used to find the orthogonal direction.

This results in a plane of orthogonal directions suitable for the assigned road direction, rather than a singular vector; this plane can be constrained to a single vector by first selecting a build direction. In order to drive the generated support structure to the build platform as quickly as possible, the global Z-axis is assigned as the build direction. Therefore, the assigned road direction is given by Equation 3.1. \( \hat{r} \) is the assigned road direction, \( \hat{z} \) is the assigned build direction (i.e., the global Z-axis).

\[
\hat{r} = \frac{\tilde{n} \times \hat{z}}{|\tilde{n} \times \hat{z}|}
\]

(3.1)

Voxels that were assigned build directions in this way are not necessarily supported themselves; they are therefore added to \( O \) to repeat the process of checking for unsupported voxels (Figure 3.1d). Once an iteration is completed without creating new support structure (i.e., \( O \) is empty), all voxels are appropriately supported and the algorithm terminates (Figure 3.1f).

3.3 Algorithm Validation

The Wheel load case discussed in Section 2.4.5 was used to validate the presented support generation algorithm. In this case, the allowable volume fraction was reduced to \( f = 0.1 \) while maintaining all of the other algorithm parameters outlined in Table 2.1. For visual clarity, Figure 3.2a presents one quarter of the optimized structure by taking advantage of the symmetry across both the XZ- and YZ-planes and will be used as a running multi-axis example in this chapter as well as Chapters 4 through 6.

The structure features a number of large overhangs with road directions oriented along the density paths, and the build directions associated with each voxel are orthogonal to those road directions. Although the density features may have build directions that could be selected to enable self-supported printing, the optimized road directions constrain the
Figure 3.2: (a) One quarter of an optimized Wheel geometry. The optimized build directions require dedicated support structure for successful fabrication and (b) Quarter-Wheel geometry with support voxels propagated (blue).

available build directions. Therefore, support structure is required to print the optimized geometry.

The structure with propagated support material is shown in Figure 3.2b, with model voxels shown in black and support voxels shown in blue. Although the model voxels are fully supported for deposition, there are 413 model voxels and 680 support voxels. As a result, approximately 62% of the deposited material will be support.

3.4 Summary of Contribution

Propagating support material along the -Z direction, which corresponds to the build direction in XY-planar deposition, does not necessarily produce a substrate for deposition in the case of steep build directions. Therefore, existing algorithms for developing support structure are insufficient for LL-MA ME. The presented algorithm generalizes the problem of generating support structure for arbitrary build directions and was demonstrated on an optimized multi-axis Quarter-Wheel geometry.
Chapter 4

Deposition Path Planning

Typically, XY-planar toolpath planners (often referred to as slicers) operate by dividing the desired geometry into 2D contours along the build direction [44]. The contours are then filled with roads through a variety of methods, depending on the specific slicer and configuration settings. This produces a series of 2D layers, filled with roads, ordered in terms of ascending Z-height [43]. When printing load bearing parts, the roads within each layer are ideally oriented by the designer such that no load is applied across the inter-layer or inter-deposition bonds. For example, to print the geometry shown in Figure 1.1 for optimal mechanical performance, the roads should align with the load paths (Figure 1.1b). The directions of these load paths form an orientation field, as shown in Figure 1.1c. While this example orientation field is relatively simple, the orientation field can be complex for industrial applications.

Due to rigid infill patterns and limited tunable parameters (e.g., raster angle), typical slicing tools have difficulty following complex orientation fields. To this end, other toolpath planning techniques have been developed that are informed by these orientation fields. For instance, a TO method used by Hoglund and Smith generates a structure with an orientation field that denotes optimal deposition directions for stiffness [34]. In their work, the orientation field followed the density features of the geometry according to Michell’s truss theorem [65]. Using this feature, they chose appropriate slicer settings to produce toolpaths consisting of exclusively contour paths that coincidentally followed the orientation field. Liu and To and Liu and Yu developed a level-set TO method where roads were created by changing the threshold for the level-set, producing concentric roads within an external contour [35, 36]. The orientation fields in these works are not explicitly used for toolpath planning; they are used implicitly by assuming the orientation field follows the density features of the design space. This implicit method is not generalizable to any geometry and orientation field, as the orientation field does not necessarily need to follow the density features. A more robust method for deposition path planning would be to directly use the orientation field to propagate roads.
4.1 Research Gap

Orientation fields become more important when using multi-axis systems, which allow the deposition head and printed part to reorient relative to each other. Unlike XY-planar deposition, which is only able to follow planar load paths, the freedom of motion afforded by multi-axis deposition enables the alignment of depositions with 3D load paths. This capability provides significant freedom to customize a part’s mechanical properties, but toolpath planning techniques that leverage this freedom are limited. The layer-like structures used in XY-planar printing are not generally available to simplify the toolpath planning problem. Multi-axis roads must be planned for the entire volume rather than single layers. To circumvent this issue, mechanical performance-driven multi-axis ME (e.g., [27, 28, 30]) is currently limited to fabricating surface geometries. These surface-based geometries do not require volumetric planning, instead only requiring deposition planning on a curved layer or surface (similar to CLS [18, 104]).

While works have demonstrated the potential to improve mechanical performance with multi-axis deposition, a generalized deposition planner that is informed by performance-driven orientation fields does not exist. Existing methods that have been used for XY-planar printing (e.g., the contour [34] and level set [35, 36] methods) are not applicable to 3D orientation fields. The features extracted using these methods would be 3D surfaces, rather than printable 2D contours. Instead, a volumetric approach to toolpath planning would be more suitable for application to both planar and multi-axis geometries and orientation fields. Ezair et al. outlined a volumetric method for generating roads using a series of extracted isosurfaces [42]. Although their method allowed predetermined curvatures to define the shape of the generated roads, only simple curvatures that kept each road along a unique trajectory were demonstrated. For more complex orientation fields, such as those demonstrated in Figure 3.2a, prescribing the road shapes is not feasible.

To address this issue, the author adapts streamline placement algorithms (reviewed in Section 4.2) to volumetrically generate roads throughout the geometry aligned to the desired orientation field. These streamline placement algorithms are commonly used to visualize complex flow fields, and operate by advecting discrete particles through the design space and tracking their motion to create streamlines that accurately follow the input flow field. These algorithms have the same criteria for success as ME toolpaths (i.e., long, continuous streamlines with even spacing), and the resulting streamlines are easily converted to roads for the ME process. By directly using the orientation field to create roads, the adapted algorithm is expected to have no restrictions on the shape of the orientation field with respect to the geometry and predetermined path shapes are not required.

The volumetric deposition path generation (VDPG) algorithm is presented in Section 4.3 and is applied to both the 2.5D MBB beam and 3D Wheel geometries in Section 4.4. Both geometries were optimized for both material distribution and orientation by the TO algorithm presented in Chapter 2. As the roads created using VDPG do not form the regular struc-
4.2 Flow Visualization with Streamlines

Flow fields in computational fluid dynamics (CFD) are calculated for arbitrary geometries with associated boundary conditions. Using a CFD software, flow velocities are obtained throughout the geometry, either within the discretized elements of the design space or at their nodes [105]. Parsing these flow velocities can be difficult for a user, especially for turbulent or otherwise complex flows [106]. This difficulty spawned an area of research focused on conveying the motion of fluid flows in an easily understandable fashion by utilizing streamlines to depict the motion in a visual manner.

As reviewed by McLoughlin et al., a popular method for obtaining these streamlines is to place discrete particles in the flow field and track their motion [107]. There are a number of different algorithms dedicated to the creation of streamlines, but they follow the same general process. First, an initial particle is tracked through the flow field. Additional seed points are propagated from the created streamline that are then also tracked to create streamlines. These streamlines are used to propagate more seed points, and the process repeats. This general structure has been used for a variety of fluid flow problems including steady [108, 109] and unsteady [110] flows and is also applicable to 3D flow fields [111, 112].

Particle tracking is performed through numerical integration of the advection equation (Equation 4.1), where $x$ is a point in the flow field with a flow velocity $V(x)$. The choice of integrator varies between works, with common choices being the Runge-Kutta (RK) and Euler methods [113, 114]. These integration schemes provide smooth, high quality streamlines and enable relatively large step sizes ($d_{\text{step}}$) to minimize computation cost.

$$\frac{\partial x}{\partial t} = V(x) \quad (4.1)$$

Creating new seed points in the design space often takes place at each integration step [115]. For each new position along a given streamline, a set of seed points is created some desired distance ($d_{\text{sep}}$) away from the streamline. This strategy ensures that, at least for the first time step, each streamline has the desired separation. Recent works have pointed out that
this method of seeding can result in a collection of short streamlines, as the initial proximity to other streamlines can cause rapid termination. For these reasons, other seeding strategies have been developed such as farthest point \cite{114} and double-queueing \cite{108}.

During advection, the streamline is monitored in regards to its proximity to other streamlines \cite{116}. If the current streamline enters within some distance \(d_{\text{term}}\) of another streamline, its advection is stopped to prevent overlapping streamlines that would confuse the visualization of the flow field. In order to improve the appearance and coverage of the streamlines, the termination distance is typically shorter than the separation distance used for seeding \cite{117}. This process of advection and seeding results in volumetric coverage of the flow field in streamlines approximately separated by \(d_{\text{sep}}\).

### 4.3 Volumetric Deposition Path Generation Algorithm

VDPG volumetrically fills the input geometry with roads that follow the associated orientation field. While typical toolpath planners take inputs in the form of an STL file, this file format does not associate well with an orientation field; instead, this work focuses on a voxel-based geometry, where each voxel has a uniquely assigned orientation, forming the orientation field. Specifically, the inputs used in this work are created using the TO algorithm described in Chapter 2, but geometries and orientation fields could be generated using other means (e.g., finite element analysis results).

VDPG borrows the general structure of iterative advection from streamline placement literature. Figure 4.1 describes the algorithm at a high-level, and the highlighted stages are detailed below with additional emphasis on modifications from the general streamline placement algorithm. Overall, these changes were required as there was a change in application: rather than focusing on aesthetic quality, the results of VDPG are used to produce a physical part.

1. **Map orientations to nodes.** The orientation field representation introduces discontinuous velocity transitions across the voxel interfaces. If left unaddressed, these discontinuities would cause either convergence or divergence of neighboring streamlines. Both effects are undesirable; streamline convergence would produce collections of short streamlines that terminate near the interface, and divergence would result in irregular spacing with potentially large, but unfillable, gaps between streamlines.

   To smooth out these discontinuities, the orientation field is mapped from the voxel representation to a node-based representation (shown in Figures 4.3a and 4.3b). As will be discussed in stage 3, the nodal orientation field is then mapped onto the advected particle to encourage longer, more continuous streamlines.

   To perform the mapping from the voxels to the nodes, the orientation of each voxel that contains the node is averaged onto the node. Unlike the velocity fields seen in
Figure 4.1: VDPG algorithm flowchart. $B$ is a seed point buffer used to hold seeds that have not been advected yet, and $Q$ is the streamline length-weighted priority queue. Highlighted stages are elaborated in the text.
Chapter 4. Deposition Path Planning

CFD, the orientation fields here are bi-directional as shown in Figure 1.1c. This is exemplified by the fact that a road can be deposited by starting from either end point; the resulting fiber orientation is effectively the same. With this in mind, it is therefore possible to obtain an incorrect zero-result during the averaging step; if, for instance, one voxel has an orientation aligned to -X and another is aligned to +X, the average would result to zero rather than either axis. To ensure this does not happen, dot products are taken for the orientations of each relevant voxel relative to one of the orientations. The orientations of each voxel with a negative dot product are flipped, and the average is taken using the repaired orientations.

2. **Seed design space.** As previously mentioned, the initial seeding strategy [118] and in-situ seeding strategy [114, 108] impact the quality of the final result. An exhaustive exploration of these impacts was not performed as part of this study; rather, this preliminary work used an initial seeding strategy that places a seed point in the center of the bottom surface of each voxel in the design space. Although it would be preferable to place the seed point in the geometric center of the voxel, initial testing resulted in an uneven bottom surface that reduced bed adhesion. By placing the initial seeds on the bottom surface of each voxel and using an appropriate in-situ seeding strategy (see stage 5), the bottom surface was evenly populated with roads. Additionally, seeds were placed in each voxel to allow the resulting streamlines to converge to the longest average streamline length (for the selected set of process parameters) with increasing mesh refinement.

3. **Advect buffer.** To promote long, continuous roads, VDPG leverages a double-queue system, similar in purpose to [108]. Each seed point in the buffer (initially populated in stage 2 and subsequently by each saved streamline in stage 5) is advected, and the length of the resulting streamline is used as the priority for the associated seed. The advection process itself is detailed in Figures 4.2a-c. A fourth-order RK integrator is used to numerically integrate the advection equation (Equation 4.1); this integrator was chosen for its prevalence in streamline placement literature and for its ability to efficiently generate smooth streamlines with large step sizes [108, 112].

The advection process requires a single velocity to be defined for the current integration step, and as previously established in stage 1, the orientation field represents two equal but opposite velocities at each point. To start, the nodal orientations are mapped onto the integration step using the eight-node hexahedral shape function [88]. At the initial seed point, both directions are advected along to produce the full streamline. At subsequent points though, dot products are formed between the previous integration step and the two opposing velocities. The positive dot product is chosen, as it approximately represents the streamline continuing its current trajectory. In contrast, the negative dot product approximately corresponds to the streamline wrapping back on itself, which would often result in the streamline terminating.

As the integration process continues, the streamline is monitored in terms of its prox-
Velocity is calculated by:
1. Mapping the nodal velocities onto the current point using a shape function
2. Choosing the appropriate velocity (from the two equal directions) by the direction most near to the direction of the previous step

Selecting $V(x_n)$

In-plane direction is the streamline velocity

Create new seed points $d_{sep}$ away from current streamline

Figure 4.2: Illustration of the advection and seeding process. (a) The particle velocity is mapped from the nodal orientations, (b) a step is taken along the velocity vector closest to the previous step, (c) the streamline is terminated either when exiting the orientation field or when in too close proximity to another streamline, and (d) additional seed points are created around the streamline to continue propagation.
iminity to the other saved streamlines in the design space. If the streamline approaches within $d_{term}$ of another streamline or itself (or leaves the design space), its advection is stopped. If the length of the resulting streamline is above the minimum length threshold, $L_{min}$, the seed point is added to the priority queue, using its advected length as the priority.

4. **Advect highest priority seed.** The highest priority seed point is advected again to ensure the state of the saved streamlines has not changed since it was last advected. If the length of the re-advected streamline is larger than the next highest priority seed point, the streamline is saved.

5. **Propagate seed points.** When the streamline is saved, new seed points are propagated and added to the buffer. At each integration point, the hexagonal pattern shown in Figure 4.2d is used to place six new seed points at $d_{sep}$ from that integration point. The hexagonal pattern was selected as it produces the highest packing density of roads [111].

6. **Recycle seeds.** If the re-advected length is less than next highest priority seed, the current seed point is added back into the priority queue (using the re-advected length as its priority) if the re-advected length is larger than $L_{min}$.

7. **Linearize streamlines.** The streamlines resulting from VDPG are spline-based curves. Depending on the capabilities of the deposition system, it may only be possible to execute toolpaths consisting of linear movement paths. Any modification of the roads after advection affects the fidelity of the paths relative to the orientation field; to minimize the impact, the chord length method [119] is used to span each streamline with a set of linear line segments. The segments are placed to minimize the chord height (i.e., the maximum distance from the original curve) below some threshold value ($d_{lin}$).

8. **Assign build directions.** To maintain perpendicularity of the deposition head during deposition, build directions must be assigned perpendicular to the road direction. Depending on the information contained within the orientation field, the build direction for each road can be assigned in one of two ways. The first assumes no build directions were specified in the orientation field; the build directions ($\hat{b}$) are assigned as the closest vector to the global Z-axis ($\hat{z}$) that is also perpendicular to the deposition direction ($\hat{n}$). If build directions are specified in the orientation field, they are instead assigned in terms of the closest vector to the average build direction of the voxels containing the road ($\hat{z}$). This process is captured in Equation 4.2.

$$\hat{b} = \frac{\hat{n} \times \hat{z}}{|\hat{n} \times \hat{z}|} \times \hat{n}$$  \hspace{1cm} (4.2)
VDPG populates the orientation field with a set of volume-filling roads, and a number of parameters make it tunable to the specific deposition system being used. $d_{sep}$ should be set to the anticipated deposition width, and $d_{term}$ should be set to the minimum value that does not result in over-deposition. Minimizing $d_{term}$ allows for both better coverage and longer, more continuous streamlines. $L_{min}$ and $d_{lin}$ should both be set relative to the minimum printable feature size of the given system.

The result of VDPG is a set of linearized roads. To generate a toolpath for XY-planar deposition, these paths can be ordered in terms of ascending Z-height to be printed layer-by-layer. The paths are unordered though, and some additional sorting or minimization may be desired to reduce travel movements. The ordering process for LL-MA ME is described in detail in Chapter 5.

4.4 Algorithm Validation

Two geometries are used to validate the presented VDPG algorithm. First, a 2.5D MBB beam geometry was used for its prevalence in AM optimization literature [120, 61, 34]. The geometry was optimized in terms of both geometry and material orientation using the TO algorithm presented in Chapter 2. Although the algorithm allows for 3D orientations, the optimization process was constrained to optimize orientations within the XY-plane to create a planar toolpath for initial algorithm validation. Second, a 3D Wheel problem is used to verify VDPG’s ability to develop suitable roads for a LL-MA geometry.

4.4.1 Case Study: Optimized MBB Beam

The optimized MBB beam (shown in Figure 4.3a) has a mesh size of 60x8x1 elements where each element has an edge length $e = 4$ units, producing a coarse structure with rapidly changing element orientations as they attempt to align with the load paths. In regions where different density features converge, the orientations are nearly orthogonal to each other. Even for this relatively simple structure, the orientation field is quite complex and does not necessarily follow the density features, violating the assumptions made in existing orientation-informed path planners [34, 35, 36]. Prescribing suitable shapes for the roads throughout the design space (as required by [42]) would also be difficult.

The VDPG algorithm and mesh parameters used in this example are shown in Table 4.1, and the streamlines produced are shown in Figure 4.3c. To validate proper streamline termination, the distances between adjacent streamlines were calculated for each line segment (i.e., between each pair of integration points). The minimum distance between the spline-based streamlines was 0.2141, which is less than the chosen $d_{term}$ but demonstrates that no streamlines overlap. Although some of the streamlines appear to overlap, this is an artifact of the XY-projection of the 2.5D structure. These streamlines actually exist at different
Figure 4.3: VDPG algorithm applied to an optimized 2.5D MBB beam geometry. (a) the input structure and deposition orientation field, (b) map the orientations to the nodes, (c) propagate streamlines through the structure, (d) linearize the streamlines for deposition, and (e) visualization of the final toolpath.
4.4. Algorithm Validation

Z-heights within the structure. After linearization (shown in Figure 4.3d), the minimum distance decreased to 0.2091, showing that the linearization process also did not produce overlapping streamlines.

Table 4.1: Selected process parameters for validation of the VDPG algorithm.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{sep}$</td>
<td>0.7</td>
<td>Seed point separation distance</td>
</tr>
<tr>
<td>$d_{term}$</td>
<td>0.4</td>
<td>Streamline termination distance threshold</td>
</tr>
<tr>
<td>$d_{step}$</td>
<td>1</td>
<td>Default integration step size</td>
</tr>
<tr>
<td>$d_{lin}$</td>
<td>0.2</td>
<td>Maximum allowable deviation during linearization</td>
</tr>
<tr>
<td>$L_{min}$</td>
<td>1</td>
<td>Minimum allowable streamline length</td>
</tr>
<tr>
<td>$e$</td>
<td>4</td>
<td>Element edge size</td>
</tr>
</tbody>
</table>

The converging and diverging streamlines in the VDPG toolpath create voids that could weaken the resulting structure. To quantify the volume of these voids, the volumetric coverage of the geometry in roads is calculated by sweeping a circle with a diameter equal to the anticipated deposition width and height ($d_{sep}$) along each linear path. The union of all of these swept paths constitutes the planned volume and accounts for overlapping roads that are within $d_{sep}$ of each other (i.e., overlapping regions are only counted once in the planned volume). The volume covered by the planned roads is 10300 cubic units, whereas the actual volume of the beam structure is 13312 cubic units, resulting in a volumetric coverage of 77.37%. The total volume of all the streamlines is 14979 cubic units, which means 31.24% of the total planned volume is overlapping and the structure will be over-deposited by 12.52%. For comparison, a toolpath for the same geometry was created with Slic3r [121] (shown in Figure 4.4) using analogous process parameters but without accommodations for following the orientation field. Although the Slic3r toolpath appears to have complete volumetric coverage, the planned volume is 10210 cubic units with a volumetric coverage of 76.70%. There is no overlap in the planned volume, but the low packing density of the layer-constrained roads means the structure is under-deposited by 23.30%. Therefore, the VDPG algorithm produced a toolpath with similar volumetric coverage as Slic3r, but the overlapping volume and over-extrusions may result in poor print quality.

Figure 4.4: Toolpath generated via Slic3r for the comparison of volumetric coverage.

This issue of choosing between over- or under-deposition is common in ME toolpath planning; often small overlaps (i.e., a negative air gap) between roads are advised for improving
mechanical strength [2, 122]. This is due to the imperfect packing density of the roads, which in the ideal case can achieve only a 90.7% packing density [123]. Ezair et al. demonstrated varying the deposition width, which should improve the packing density, but still produces small voids between adjacent depositions [42]. This is to say, changing the $d_{term}$ parameter produces a similar behavior to changing the air gap in a typical slicer; volumetric coverage increases along with overlapping extrusions and the potential for over-deposition. Only one set of parameters was demonstrated in this study, but further optimization of the $d_{sep}$ and $d_{term}$ parameters is expected to improve the performance of VDPG.

Finally, a GCode toolpath, shown in Figure 4.3e, can be generated by ordering the roads in terms of ascending Z-height. Each point along the linearized streamlines was converted to a linear movement command with extrusion lengths and feed rates assigned using typical slicer parameters (e.g., road width, filament diameter, nozzle diameter, etc.).

### 4.4.2 Case Study: Optimized Wheel

The algorithm was also tested using the quarter-Wheel TO result (with propagated support material) shown in Figure 3.2. For comparison purposes, Figure 4.5 depicts the orientation field associated with the geometry where the deposition directions are shown with red vectors and the build directions are shown in blue. Roads for both the model and support structure were propagated using VDPG with the parameters outlined in Table 4.2. These parameters were experimentally established to result in a dense model structure with minimal support. The support material values for $d_{sep}$ and $d_{term}$ were increased from the model material in order to produce a support structure with a lower density. $d_{lin}$ was increased for the support material as the support roads did not need to follow the generated orientation field with the same level of accuracy. As will be discussed in Chapter 5, it was beneficial to decrease the number of support paths (by making each individual path longer) to improve performance in the collision detection and ordering steps. The propagated structure is shown in Figure 4.6, with model material shown in black and support material shown in blue.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Model Value</th>
<th>Support Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{sep}$</td>
<td>Seed point separation distance</td>
<td>1.3</td>
<td>1.4</td>
</tr>
<tr>
<td>$d_{term}$</td>
<td>Streamline termination distance</td>
<td>0.7</td>
<td>1</td>
</tr>
<tr>
<td>$L_{min}$</td>
<td>Minimum streamline length</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>$d_{lin}$</td>
<td>Maximum linearization chord length</td>
<td>0.35</td>
<td>0.5</td>
</tr>
<tr>
<td>$L_{max}$</td>
<td>Maximum linearized road length</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

As compared to Figure 4.5, the model road directions qualitatively follow the desired orientations. Additionally, there are continuous roads linking all of the individual truss features...
4.5 Streamlines Mechanical Property Evaluation

A multi-load case featuring i) tension and ii) three-point bending is used to evaluate the mechanical properties of printed specimens planned using VDPG. The load case allocates regions for test fixturing as shown in Figure 4.7. In contrast to other optimized printing studies that have leveraged Michell truss structures for toolpath planning (e.g., [34]), the multi-load case is used to encourage a result that is not a Michell truss structure. In the fixed regions, the TO algorithm is able to optimize the material orientations, but the material distribution is fixed. In the first load case, a distributed tensile load is applied to the right grip region of the structure with a fixed boundary condition applied to the left. 10 mm grip regions are allocated for fixturing the specimen in the grips, and the fixed boundary
Chapter 4. Deposition Path Planning

Gap between model and support due to discretized curve. The gap is approximately $d_{gap}$, due to the orthogonal interface. Continuous deposition paths between features.

Figure 4.6: Roads propagated through the quarter-Wheel geometry. Model material paths are shown in black, and support material is shown in blue.

Figure 4.7: The multi-load case used throughout this work. The design space experiences two load cases: (a) tension and (b) three-point bending and has fixed regions allocated for test fixturing.
Although the load cases are individually symmetric, they are symmetric around different planes; therefore, the structure was optimized as shown using a 2.5D mesh of 440x160x1 eight-node brick elements. A voxel size of 0.25x0.25x8 mm was used, resulting in a printed specimen with dimensions of 110x40x8 mm. The optimized structure is shown in Figure 4.8, using a 10% volume fraction. In terms of thresholding, elements with a pseudo-density of $\rho_i < 0.01$ are not displayed or included in the printed geometry; density is not otherwise represented in the figure. Instead, the coloring of each pixel represents the optimized material orientation as determined in the associated color wheel (as previously described in Figure 2.5).

Figure 4.8: (a) The optimized multi-load case geometry. (b) Material orientations are denoted by the color legend. The legend is symmetric across the Y-axis; a desired deposition direction along, for instance, the +X axis is the same as being aligned to the -X axis.

### 4.5.1 Toolpath Planning

Three toolpaths are generated for the geometry shown in Figure 4.8. First, VDPG is used to process the geometry as-is, propagating roads as described in Section 4.3. The parameters for VDPG are listed in Table 4.3, and all samples were printed on the multi-axis deposition platform described in more detail in Chapter 6.

For the control samples, uniform and contour infill patterns are used to create toolpaths in Simplify3D [124]. While VDPG was able to process the geometry directly in the voxel-based format, toolpath generation for the control samples required exporting the optimized geometry as an STL file. The STL file was not smoothed for toolpath generation; discretization artifacts were present, but they did not appear to affect toolpathing. The critical slicing parameters for Simplify3D are listed in Table 4.3. In an effort to mitigate the effects of testing the parameterization, parameters were kept as similar as possible between the toolpath planning techniques.
Table 4.3: Toolpath planning parameters for VDPG and the control (uniform and contour) toolpath planning strategies. Parameters were kept as consistent as possible between the strategies to minimize the effects of the specific parameterization. For the control parameters, single-valued parameters apply to both uniform and contour toolpaths; for two-valued parameters, the value in the parentheses refers to the contour toolpath.

<table>
<thead>
<tr>
<th>VDPG Parameters</th>
<th>Control Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{sep}$</td>
<td>1.1 mm</td>
</tr>
<tr>
<td>$d_{term}$</td>
<td>0.6 mm</td>
</tr>
<tr>
<td>$L_{min}$</td>
<td>5 mm</td>
</tr>
<tr>
<td>$d_{lin}$</td>
<td>2 mm</td>
</tr>
<tr>
<td>$L_{max}$</td>
<td>5 mm</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.5.2 Material

Carbon-fiber loaded polylactic acid (CF-PLA) from Ziro3D [125] was used for all of the experiments presented in this study. Although the maximum manufacturer-specified printing temperature is 240°C, 260°C was used to compensate for the elongated nozzle.

4.5.3 Mechanical Testing Setup

The following sections detail the specifics of the mechanical test setups for validating VDPG using the multi-load case geometry. All mechanical tests in this work were performed on an Instron 5984 [126] with a cross-head movement rate of 5 mm/min and a 10 kN load cell. Six specimens were printed using each of the toolpath planning strategies to allow for three replicates of each load case.

Tensile Test Setup

The geometries were placed in the grips as shown in Figure 4.9a such that the left hand side (i.e., the side with the fixed boundary condition) was in the stationary cross-head. Extension measurements were taken directly from the cross-head motion as the geometry was not conducive to mounting an extensometer.
4.5. Streamlines Mechanical Property Evaluation

Three-Point Bend Test Setup

For three-point bending, the specimens were placed as shown in Figure 4.9b. As prescribed by the load case, the top cross-head does not apply the load in a line as in the typical three-point bend test setup. Instead, a distributed load is applied using a larger diameter attachment to prevent the load application site from shifting during testing.

Measurement of Mechanical Performance

Typically, simple geometries and load cases (e.g., tensile bars) are used to characterize performance as they have predictable fracture regions that allow for direct comparison of intrinsic material properties like modulus and yield strength. When comparing toolpath planning strategies though, simple geometries do not allow for substantial differentiation. For example, the load paths in a tensile bar are almost perfectly followed by a uniform infill pattern, but this is not representative of a general geometry used in a load bearing application. To this end, the previously described multi-load case and complex geometry is used to expose differentiation between the toolpath planning strategies. The irregular specimen geometry meant the fracture surface could not be accurately predicted and intrinsic property comparison was difficult. Therefore, rather than normalizing the load by area, the mass of each specimen was measured to normalize the maximum experienced load by the amount of material in the specimen; this is used to quantify the relative efficiency of the structure. Further discussion of stress and modulus measurements is presented in Appendix C.
4.5.4 Results

Three different toolpathing techniques are explored and compared in this work: i) VDPG, ii) a uniform infill pattern, and iii) a contour infill pattern. The optimization of both the geometry and orientation field with respect to the anticipated stress fields within the structure are expected to enable VDPG (which actively aligns the toolpath to the orientation field) to outperform the other toolpathing strategies. The following sections i) compare the toolpaths for the optimized specimens in terms of potential stress concentrations and alignment to the optimized orientation field and ii) present mechanical testing results for the printed specimens in each of the two load cases.

Toolpath Comparison

Figures 4.10a-c show the toolpaths produced using each of the three techniques, and Figures 4.10d-f depict a printed specimen using each toolpath. Of particular note, there are many roads within the VDPG specimen that span across multiple truss members which should serve to improve mechanical performance by providing a continuous path for load to travel (as opposed to being broken up by inter-deposition interfaces). While the contour toolpath exhibits similar behavior, the outside-in nature of the contour toolpathing leaves gaps in many of the truss regions that could produce stress concentrations. In contrast, the volumetric approach of VDPG is inside-out, which helps to support those critical regions. On the uniform toolpath, the thin truss members (which are typical of TO structures) are heavily discretized by the roads. Additionally, some of the trusses sit at sub-optimal angles for the technique, resulting in only a single road supporting the cross-section.

The alignment of each toolpath is quantified relative to the stress-aligned orientation field (Figure 4.8) in Figure 4.11. The direction of each road is compared to the prescribed orientation in the containing element using Equation 4.3.

\[
c = \frac{\sum_{i=0}^{N_r} \sum_{j=0}^{N_s} (r_i \cdot r_{e_{i,j}}) l_{i,j}}{\sum_{i=0}^{N_r} \sum_{j=0}^{N_s} l_{i,j}}
\]  

(4.3)

\(N_r\) is the number of roads in the toolpath. Each road \(i\), with direction \(r_i\), is cut into a number of segments, \(N_s\), such that each segment is only contained within a single element, \(e\), with a length \(l_{i,j}\). The containing element has a stress-aligned road direction of \(r_{e_{i,j}}\). Equation 4.3 calculates the correlation, a value between 1 (i.e., the toolpath is completely aligned with the orientation field at all points) and 0 (i.e., the toolpath is completely orthogonal to the orientation field at all points).

As shown in Figure 4.11, the correlation of the VDPG toolpath to the input orientation field is 0.989. The slight discrepancy to full alignment is caused by i) the nodal mapping done during the advection stage in order to create smoother roads and ii) the linearization
Figure 4.10: Toolpaths from each (a) VDPG, (b) uniform, and (c) contour, and printed specimens from the corresponding toolpaths: (d) VDPG, (e) uniform, and (f) contour.
stage. In contrast, the uniform and contour toolpaths have correlation values of 0.7637 and 0.6954, respectively. This difference in alignment between VDPG and the other toolpathing strategies is expected to result in improved mechanical performance.

Although the fixed regions prevented the development of a true Michell truss structure, the vast majority of the geometry is truss-like. Therefore, it was expected that the contour toolpath would have an improved correlation value when compared to the uniform toolpath. The alignment of the uniform toolpath was enhanced by the roads that connected adjacent uniformly orientated roads, effectively creating a discretized contour around the uniform toolpath. Additionally, the grip regions had significant portions that aligned with the X-axis. This is demonstrative of the geometry-specific nature of these passive alignment options; not all orientation fields will conveniently line up with a passive infill pattern.

The VDPG and contour toolpaths featured significantly more travel movements, leading to some filament stringing on the exterior surfaces of the printed parts. Despite the additional travel movements, the contour and uniform toolpaths had similar print times at 38.5 minutes and 35 minutes, respectively. In contrast, the VDPG toolpath took 54.5 minutes. This increase in print time is largely due to the ordering algorithm used to organize roads at the same height. The algorithm used is described in more detail in Chapter 5, but in brief, the algorithm was greedy and selected the nearest available road at the same height. These print times would affect the strength of the inter- and intra-layer bonds. As VDPG is designed to align the roads with the orientation field (i.e., the load paths) this increase in print time should have little impact on the performance of the printed parts. On the other hand, the deposition order for the uniform toolpath would have an effect on the performance of those
4.5. Streamlines Mechanical Property Evaluation

parts. The short lengths of each road mean the adjacent deposition is placed rapidly, allowing for improved bond strength. In some areas though, the toolpath does not place the adjacent deposition immediately, leading to weak bond formation along the load paths.

Mechanical Testing

The mechanical testing results are summarized in Figure 4.12. The trends from the alignment measurements are maintained in the mechanical data; the VDPG samples outperformed the control samples, and the uniform samples did marginally better than the contour samples. The VDPG samples had an average maximum load of 138.77 $N/g$ and 70.71 $N/g$ in tension and bending, respectively. The uniform samples had averages of 64.47 $N/g$ and 54.71 $N/g$, respectively; the contour averages were 61.64 $N/g$ and 44.78 $N/g$, respectively.

![Graph showing mechanical property evaluation results](image)

Figure 4.12: The maximum load applied to the printed structures in each load case are normalized by the mass of the structure. The VDPG toolpath outperformed the uniform toolpath by 108.24% and 29.25% in tension and bending, respectively.

The high variance in the contour bending samples is due to the premature failure of one of the bending specimens. While the other two samples catastrophically failed along the length of multiple truss members (Figure 4.13a), one sample simply failed at the joint between the truss and the point of support as shown in Figure 4.13b. The premature failure was due to that region being under-deposited, relative to the other contour samples, which led to a stress concentration. As seen in the toolpath (Figure 4.10c), there is a gap in that region. This variability is likely not uncommon for different geometries; thin truss members are difficult to fill with an outside-in approach. Unless the thickness of a truss member is an integer multiple of the road width, there will be a gap in the toolpath, and this issue is exacerbated in regions where truss members converge. Due to the inside-out approach of VDPG, these interior gaps are eliminated.
Chapter 4. Deposition Path Planning

Figure 4.13: (a) Catastrophic failure of the bending sample produced a higher maximum load, and (b) a premature failure at the point of support reduced the maximum load.

While the control samples had relatively similar performance between the load cases, the VDPG samples withstood significantly less load in bending than in tension. This does not agree with the original TO problem statement, as the load cases were weighted evenly. This was likely caused by the relatively consistent failure mode in all nine of the bending samples near the right-hand support (Figure 4.13). The bending load is transferred more heavily to this region than the upper truss on the right-hand side. In contrast, the tensile load case more evenly distributes load across the right-hand truss members.

The geometry generation can be improved in a few ways, which are discussed further in Section 8.4. First, the bending load case imposes flexural and compressive loads on the structure that are not appropriately modeled. The elastic matrix used to describe the element in the TO algorithm assumes compressive and tensile strengths are equivalent. For homogeneous ABS, compressive strength has been shown to be up to five times greater than tensile strength [127]. The use of the eight-node brick element with a linear shape function also over-predicts the bending stiffness of the structure [128]. Using a higher order shape function and using an incompatible formulation of the element should give better results in the bending-dominated loading case [129].

Second, the TO algorithm did not optimize the structure for strength directly but rather stiffness, which is used as an analog for strength. Whereas stiffness is a global structural metric, stress is a local metric that introduces a large number of additional constraints [87]. To accommodate these computational requirements, a shape optimization (SO) step is typically performed following the TO to further refine the exterior surface of the geometry to satisfy the stress constraints.
4.6 Summary of Contribution

The ME process produces parts with anisotropic mechanical properties due to the thermal characteristics of material deposition. These anisotropic properties can be tuned by changing the orientations of roads, but their desired orientations can be quite complex depending on part geometry and performance requirements. Toolpath planning algorithms capable of generating roads that follow these orientations are limited. Level set [35, 36] and contour-based [34] methods are only applicable to orientation fields that follow the density features of the geometry, and existing volumetric approaches [42] rely on the road shapes to be provided as input. An algorithm for volumetrically generating ME toolpaths was presented that is agnostic to both the geometry and orientation field and does not require predetermined path shapes. The VDPG algorithm leverages streamline placement algorithms commonly used for CFD flow field visualization to place volume-filling roads throughout a structure that follow the desired orientation field. These spline-based streamlines are then linearized into roads, and build directions are assigned perpendicular to the roads.

A 2.5D MBB beam geometry (Figure 4.3), optimized in terms of both topology and material orientation (Chapter 2), was used to validate the VDPG algorithm. Using the parameters outlined in Table 4.1, the algorithm produced a toolpath with roads that covered 77.37% of the geometry’s volume. For comparison the same process parameters were given to a more traditional slicer, with no accommodations for following an orientation field, that produced a toolpath that covered 76.70% of the volume. Although the VDPG algorithm covers more of the volume, it plans to over-deposit the structure by 12.52%. On the other hand, the conventional toolpath under-deposits the structure by 23.30%, exemplifying the trade-off between over- and under-deposition in process parameter selection.

An optimized 3D Wheel geometry (Figure 4.5) and its associated support structure (Figure 3.2) was also propagated using VDPG. The propagated model material qualitatively follows the optimized orientation field, but the interface between the model and support material is mesh dependent which may result in poor deposition quality. Specifically, the discretization of geometric features may produce large gaps between the propagated support and model material roads. This is caused by an inability to fill the corners of the exterior voxels due to the minimum length parameter ($L_{\text{min}}$).

Finally, the presented toolpath planning algorithm was validated using a TO structure that was subjected to two load cases: i) tension and ii) three-point bending. The TO algorithm was allowed to vary density and material orientation throughout the structure to minimize structural compliance relative to the load cases. Allocations for test fixturing were also created in the design space, where the density distribution was fixed but the TO algorithm was still allowed to vary material orientation. The resulting geometry, shown in Figure 4.8, was input to the presented VDPG algorithm as well as uniform and contour toolpath planning methods for comparison.
Printed structures from each toolpath planning strategy were tested in both load cases. As shown in Figure 4.12, the VDPG specimens resulted in 108.24% and 29.25% improvement in tension and bending, respectively, when compared to the next best performing toolpath (i.e., the uniform toolpath). This improvement is attributed to a better alignment of the toolpath to the optimized orientation field (illustrated in Figure 4.11).
Chapter 5

Deposition Path Ordering

Due to the layer-by-layer nature of XY-planar deposition, it can be safely assumed that all previously deposited material is below the deposition head [130]. This allows layers to simply be ordered in terms of ascending Z-height, and individual roads within a layer can be ordered with regard to any metric (e.g., minimizing deposition head movement). With multi-axis deposition, roads are able to exist in 3D, rather than being constrained to the XY-plane. Current multi-axis strategies targeting improved mechanical properties (e.g., [27, 28, 30]) make the assumption that all roads lie on curved surfaces (similar to CLS [104, 18]). In this way, roads can be ordered on a per-surface basis without introducing collision concerns (provided the surface is not too highly curved). This limitation inhibits the degree of deposition alignment that can be achieved; if the anticipated load paths do not follow stratified surfaces, alignment of surface-constrained roads will not be optimal.

Shifting to LL-MA strategies is necessary to achieve full alignment, but the collision concerns inherent in this style of deposition must be explicitly handled. If roads are improperly ordered in the toolpath, previously deposited material could prohibit access to undeposited regions of the part. Either those regions are unable to be deposited, leaving voids in the structure, or the deposition head attempts to deposit them anyways and collides with the previously deposited structure. To guarantee a collision-free LL-MA toolpath, a more intelligent toolpath planning method is required.

Ezair et al. presented a method for creating a collision-free ordering of print paths for non-planar printing, but only for a single build direction [42]. Their method built on the underlying assumption that, from the single build direction, a small portion of the lowest road (in Z-height) could be freely printed at any given time. This assumption, called guaranteed fragment printability, enabled a collision-free toolpath to be produced, but the assumption is not maintained when considering the use of multiple build directions. Instead, a method similar to that presented by Wu et al. is more appropriate, which considers the collision footprint of the deposition head to create a dependency graph for all of the roads [32]. While the algorithm was shown to be effective for some mesh structures, the number of roads contained in a dense structure would lead to combinatorial explosion. Additionally, the algorithm had no considerations for printing long continuous paths. This limitation would heavily impact the final mechanical properties of the part as the discontinuous print paths would introduce
weakness to the final part; this becomes particularly important when considering composite materials were significant mechanical strength is leveraged from reinforcement length [16].

5.1 Research Gap

The proposed structures require the precedence (or dependency) of the roads to be evaluated on a by-road basis (as opposed to a by-layer or by-surface basis). This requires a large number of precedence constraints to be evaluated and satisfied in order to ensure a collision-free toolpath has been created. Task ordering techniques for assembly manufacturing (reviewed in Section 5.2) address the issues of precedence constraints and minimizing deposition head movement for a set of tasks. These existing assembly planning techniques, as will be discussed, do not take into account the desire to perform certain tasks (i.e., deposit roads) in a sequential order. For the purposes of reducing the number of start-stop voids and discontinuities along the load path and maintain the length of reinforcement in a composite material, it is preferable to deposit roads that share an end point sequentially. Incorporating this continuity consideration into the task ordering algorithm should improve final mechanical properties.

The initial algorithm for obtaining a collision-free toolpath with considerations for deposition continuity is presented in Section 5.3. An evaluation of the algorithm in the context of the Wheel toolpath (Figure 4.6) is given in Section 5.4. Finally, a summary of the contribution is discussed in Section 5.5.

5.2 Assembly Planning

By considering each road comprising the desired structure as an individual task, the ordering problem can be explored as an assembly problem similar to those found in traditional manufacturing (reviewed in [131, 132, 133, 134]). In this type of problem, it is well known that an exhaustive combinatorial search is infeasible due to the number of tasks [135]. Instead, these problems are explored in terms of the interactions between each component or task in the assembly to inform a final assembly plan (e.g., [136, 137, 138, 139]). These algorithms are often only demonstrated for assemblies with a small number of components (typically under 100) [139], making them infeasible for printing applications, which typically consist of over 10,000 roads.

Instead, it is often easier to plan the disassembly of the structure, rather than the assembly [140]. This is due to the fact that the assembled structure is more heavily constrained than the disassembled components, reducing the number of feasible options. A number of reviews have been produced on the topic of disassembly, primarily focused on disassembly for recycling, using similar algorithms to those discussed for assembly planning [141, 142, 143].
Critically, these disassembly plans can be inverted to create a valid assembly plan if the disassembly is not destructive and does not involve deformable components (often referred to as assembly-by-disassembly) [144].

Using the classification framework presented by Srinivasan et al., the disassembly problem relevant to LL-MA ME can be defined to assist in identifying an appropriate solution [145]. By the definitions presented in that work, the problem of toolpath planning for LL-MA ME is 1-disassemblable, as only a single linear tool head movement is required for any given deposition. The problem is also monotonic and sequential, as only a single deposition can be created (or removed) at a time. Finally, the toolpath requires a complete disassembly plan (i.e., all tasks must be disassembled individually) and the disassembly plan must be non-destructive. Due to the relative simplicity of this problem, only a precedence matrix (i.e., a matrix describing the order in which tasks can be completed) is required to develop a feasible toolpath [146, 147], but a continuity matrix (i.e., a matrix describing roads that form continuous depositions) should be considered to promote continuous depositions.

5.2.1 Task Ordering with Precedence Constraints

Ordering tasks with precedence constraints is a well-known problem [148] that has been shown to be NP-complete [149]. The precedence constraints can be represented as a square matrix \( C = [c_{i,j}] \) with dimensions equal to the number of tasks. If task \( i \) must precede task \( j \), \( c_{i,j} = 1 \), otherwise \( c_{i,j} = 0 \). An ordering that satisfies the precedence constraints can therefore be found by transposing the rows and columns of \( C \) to create an upper triangular matrix \( C^* \). This is shown by considering a task \( i \) and any task \( j \) that occurs before it (i.e., \( i > j \)); if \( c^*_{i,j} = 1 \), that means task \( i \) should precede task \( j \) but does not. If \( c^*_{i,j} = 0 \) \( \forall \ i > j \), that satisfies the criteria for an upper triangular matrix. If an upper triangular \( C^* \) cannot be found, there are cyclic precedence constraints that cannot be resolved. Although a polynomial-time algorithm has been presented for this problem [149], it does not consider the continuity of the tasks nor the movement of the tool head. Due to the additional complexity imposed by considering continuity and tool head movement, the author aims to develop an algorithm for creating a feasible, but not necessarily optimal, toolpath. Specifically, this criteria makes no guarantees in terms of achieving the minimum travel length or the maximum continuity in the final toolpath, just that the toolpath is collision-free.

5.3 Toolpath Planning Algorithm

The toolpath planning algorithm operates in two stages: i) establish precedence constraints (Section 5.3.1) and ii) order roads (Section 5.3.2). If the algorithm cannot find a valid (collision-free) ordering for the roads, it utilizes two continuation methods in an effort to break cyclic precedence constraints.
5.3.1 Establish Precedence Constraints

A precedence constraint occurs when the deposition head depositing a given road (A) requires volume occupied by another road (B). If road B lies in this volume, road A must be printed before road B (i.e., $c_{A,B} = 1$). This situation is illustrated in Figure 5.1.

![Figure 5.1: As the deposition head (grey) deposits road A, aligned to the build direction of road A (blue arrow), it needs to occupy a certain volume (black dotted line). If another road (B) occupies that volume, road A needs to be printed first in order to ensure the volume is available during printing.](image)

Algorithm 2 is used to establish these collision constraints. At a high level, a conservative model of the deposition head is created that fully encapsulates the actual volume of the deposition head. Then, for each road, the Minkowski sum of the deposition head model is taken over the length of each road, with the model oriented along the build direction of the road, creating the collision volume. A collision volume for each road is also created by sweeping the expected deposition cross-section along the length of the road. The deposition head collision volume for each road is then compared to the collision volumes of all of the other roads in the structure (resulting in $n^2 - n$ comparisons, where $n$ is the number of roads).

In this work, an initial check was performed between collision volumes to ensure they were in the same neighborhood. If so, the Gilbert-Johnson-Keerthi (GJK) algorithm [150] was used to verify a collision occurred (i.e., a precedence constraint needed to be created).

This method assumes the motion control system (e.g., the robotic arm) will never collide with the part and therefore does not need to be considered. For this work, the part being printed is small relative to the deposition head. If larger parts are to be fabricated, it may be necessary to also compare the collision volume occupied by the deposition system as each road is deposited. Regardless, the overall process remains the same.
5.3. Toolpath Planning Algorithm

Algorithm 2: Collision detection algorithm

| Input: Tool, \( R_s \), \( R_f \), \( B \) |
| Output: \( C \) |
| \( 1 \) Tool: Representative model of the deposition head and motion control system if necessary |
| \( 2 \) \( R_s \): Vector of start points for each road |
| \( 3 \) \( R_f \): Vector of end points for each road |
| \( 4 \) \( B \): Vector of build directions for each road |
| \( 5 \) \( n \) = the number of roads |
| \( 6 \) \( C = [c_{i,j} = FALSE] \forall i, j = 1, \ldots, n \) |
| \( 7 \) for \( (i = 1; i <= n; i = i + 1) \) |
| \( 8 \) Create the collision volume for road \( i \) by taking the Minkowski sum with Tool |
| \( 9 \) for \( (j = 1; j <= n; j = j + 1) \) |
| \( 10 \) Create the collision volume for road \( j \) |
| \( 11 \) Detect collisions between \( i \) and \( j \) |
| \( 12 \) if collision then |
| \( 13 \) \( c_{i,j} = TRUE \) |

5.3.2 Order Deposition Paths

The result of the collision detection algorithm is a square collision matrix \( C \) that must be permuted to an upper triangular matrix \( C^\ast \) to satisfy the precedence constraints. To maintain the correct constraint referencing, this reordering must be performed with coupled row-column permutations (i.e., if rows 1 and 2 are permuted, so must columns 1 and 2). The vector \( P \) that permutes \( C \) to upper triangular form is a collision-free ordering.

The algorithm used in this work to create \( P \) that considers precedence constraints, continuity, and deposition head movement is illustrated in Figure 5.2. At a high level, the algorithm searches the unordered sub-matrix of \( C \) (\( C^\ast \)) for collision-free roads, then attempts to chain those roads together to preserve continuity. When possible, the algorithm adds completed chains to the toolpath (\( P \)) and continues by creating a new sub-matrix \( C^\ast \). If it cannot continue (i.e., no collision-free roads can be found), the algorithm attempts to address conflicting precedence constraints by i) reorienting the build directions of the unordered roads and ii) removing roads in low density regions from the toolpath. Additional details on each of the highlighted regions in the flowchart are given in the following paragraphs.

Precedence Constraints. The precedence constraints represented in \( C \) are satisfied by first assembling a sub-matrix \( C^\ast \) of all of the unordered roads (i.e., the roads that have neither been added to \( P \) nor deleted from the toolpath). Second, this sub-matrix is searched for collision-free roads which are denoted by an empty row in \( C^\ast \). Without considerations for
Figure 5.2: High level overview of the ordering algorithm used to create a collision-free tool-path. Considerations for precedence constraints, road continuity, and minimizing deposition head movement are included. Highlighted regions are discussed in more detail in the text.
Continuity Considerations.  If two roads share an end point, it is ideal to print them sequentially to preserve the continuity of the roads. Therefore, the collision-free roads (found in the previous step) that share an end point are grouped together to form chains. If a chain is complete (i.e., the end points of the chain are not shared by any unordered road), it can be added to $P$ without introducing deposition discontinuities. In the case that no chains can be completed, the algorithm also permits adding incomplete support material chains to $P$, as those do not contribute to the final performance of the structure. Further, if all of the continuation methods have been used without anything being added to $P$, any started chain is added to $P$ to guarantee a toolpath is generated.

Tool Movement Considerations.  In the case that multiple chains are suitable for adding to $P$, the algorithm attempts to minimize the movement of the deposition head. The algorithm checks both end points of each candidate chain and minimizes the distance to the last-added road in $P$. Only the chain with the minimum distance is added to the $P$.

Continuation Methods.  It is often possible for $C$ to not have a possible upper triangular permutation. In these situations, there is not a $P$ that produces a collision-free toolpath. This is encountered when there are no collision-free candidates available to create chains.

In this case, the algorithm removes a subset of the roads from the toolpath in order to continue. To minimize the impact on the final structure, the algorithm calculates the cost to the final structure for freeing each road, shown in Equation 5.1. $s_i$ is the cost to free (i.e., make collision-free) road $i$. Each road is decomposed into segments, each with length $l_{j,k}$ and contained in a single element with a pseudo-density $\rho_k^e$. This calculates a weighted length for each road $j$; the sum of the weighted lengths of the roads that collide with road $i$ becomes the cost to free road $i$.

$$s_i = \sum_{j=1}^{n} \left( c_{i,j} \sum_{k=0}^{N_s} \rho_k^e \sum_{l_{j,k}=0}^{N_s} l_{j,k} \right)$$  \hspace{1cm} (5.1)

The roads restricting the lowest cost road (i.e., the road that requires deleting the least weighted-length roads) are removed from the toolpath by clearing their associated rows and columns in $C$ and $C^*$, allowing the algorithm to continue. Although this ensures the algorithm will finish, it may be necessary to delete a large number of roads for a complex toolpath, resulting in a structure with compromised integrity and strength.

An additional continuity method is implemented in order to improve chain length. Specifically, if chains cannot be finished, the unordered roads are searched for collision-free build directions. The build directions of the roads in $C^*$ are rotated, and collisions are recalculated.
(as described in Section 5.3.1). This process can be repeated any number of times, rotating the build directions of the unordered roads further each time.

### 5.4 Multi-Axis Toolpath Planning Evaluation

A convex approximation of the deposition head, shown in Figure 1.2, was used for the collision detection step. The geometry used a cone, with an angle of 50° and a height of 80 mm, and a cylinder with the same radius as the top of the cone that continued to the extent of the design space (approximately shown in Figure 5.1). This composite shape over-estimates the volume required by the deposition head, ensuring no false-negatives are produced during the collision detection step.

The reorientation continuation method parameters were $\Delta \theta = 5^\circ$ for each iteration of the build direction search with a maximum deviation of $\pm 30^\circ$ from the original build direction. To prevent collisions between the robot and the build platform, negative build directions were not allowed. Additionally, to prevent collisions between the deposition head and the build platform, build directions more than 40° away from the global Z-axis were not allowed below 95°. 34 mm (the radius of the deposition head model). This constraint prevents the deposition head from assuming steep orientations until the deposition head is sufficiently above the build platform.

Three metrics are used to evaluate the produced toolpath i) if the toolpath is collision-free (i.e., upper triangular), ii) how many roads that share an end point are printed sequentially, and iii) how many roads were deleted in order to obtain a feasible toolpath. In order to evaluate these metrics, Figures 5.3 and 5.4 show the collisions matrices ($C$) and the continuity matrices ($Y$) before and after the ordering algorithm, respectively. The continuity matrix represents the number of roads that could have been printed sequentially in order to preserve continuous road length. For example, if road $i$ shares an end point with road $j$, $y_{i,j} = 1$ and $y_{j,i} = 1$. Therefore, in a perfect ordering in terms of continuity, path $i$ will immediately precede or follow path $j$ for all $i$ and $j$. This places the non-zero indices of $Y$ on the second diagonals. Although the initial continuity matrix (Figure 5.4a) shows a perfect ordering for continuity, the collision matrix (Figure 5.3a) is not upper triangular.

The resulting toolpath, colorized by the order in which the roads should be placed, is shown in Figure 5.5. During the ordering step, multiple roads had conflicting build directions, in particular around the right hand side of the structure as shown in Figure 4.6. This was due to the slope of the model material relative to the angle of the deposition head. Although the reorientation routine attempted to repair these conflicts, the algorithm was unable to find suitable build directions for many of the roads and resorted to removing them from the structure. The ordered collision matrix, shown in Figure 5.3b, is collision-free as evidenced by the upper triangular shape of the matrix. A total of 1312 roads were removed, with 1225 of them being support paths and 87 model paths (out of a total 5909 and 4790, respectively).
5.4. Multi-Axis Toolpath Planning Evaluation

Figure 5.3: Collision matrices with model path collisions shown in black and support paths shown in blue: (a) the unordered collision matrix, (b) the collision matrix after the first ordering is upper triangular, but required the removal of 1312 roads, and (c) the second ordering also produced an upper triangular matrix, but required the removal of 1455 roads.

Figure 5.4: Continuity matrices with model path continuities shown in black and support paths shown in blue: (a) the unordered continuity matrix has perfect continuity as the paths are still in the order they were generated, (b) the first ordering resulted in 3548 properly sequenced continuous paths out of a total of 7098 (1357 continuous model roads were sequenced properly out of 3013), and (c) the second ordering resulted in 4248 properly sequenced continuous paths (1701 properly sequenced continuous model roads).
As shown in Figure 5.4b, a total of 7098 paths could have been ordered sequentially, but the ordering algorithm was only able to sequence 3548 of them (including 1357 sequenced model roads out of a possible 3013). As a result, 1656 discontinuities will be introduced into the part; at these points in the toolpath, the deposition platform will not complete a set of continuous roads in a single pass and will return later in the toolpath to finish it. These discontinuities are weaker than the deposition direction (i.e., the continuously deposited material) and are directly in line with the load paths, reducing the overall performance of the printed structure.

Figure 5.4c: Continuity matrix showing ordered toolpath with 4248 roads, 25% improvement over first ordering.

Due to the number of roads that were removed from the toolpath, it is possible that conflicting collisions that prevented continuous roads from being ordered sequentially would have been solved. Therefore, the updated roads (with deleted roads still removed) were input to the ordering algorithm a second time. This produced the continuity matrix shown in Figure 5.4c. The toolpath is still collision-free (as shown in Figure 5.3c), but the continuity matrix successfully sequenced 4248 roads including 1701 model roads (a 25% improvement over the first ordering).

Although the set of roads fed to the ordering algorithm the second time were previously able to be ordered in a collision-free manner, the second ordering resulted in an additional 143 support roads being removed. This should not occur, as a collision-free ordering was obviously available; this implies that conflicting collision constraints were created during the reorientation continuation method. As will be discussed in Chapter 8, the author aims to
create a more robust algorithm for ordering roads that does not require multiple iterations or the unnecessary deletion of roads to fix these issues.

5.4.1 Computation Time

The presented method for ordering roads is computationally expensive, requiring approximately 90\% of the computation time for the process planning stage (i.e., excluding the time required by the TO algorithm). The majority of this time is consumed by the continuation methods; the reorientation routine in particular necessitates \( n \) additional collision checks for each candidate road to find disassemblable build directions. Currently, candidates are chosen with the heuristic of low-collision roads in \( C^* \) (i.e., roads with fewer collisions than some threshold value, \( n_{col} \)). Essentially, the \( n_{col} \) parameter attempts to find roads that are nearly disassemblable.

The purpose of the continuation methods is to break cyclic precedence constraints in \( C \). Only looking at nearly disassemblable roads does not necessarily break cyclic precedence constraints and can actually introduce more. Discussed in more detail in Section 8.4, there are existing algorithms in literature for exploring this matrix sorting problem using graph representations of the matrix. Critically, Tarjan presented a method for identifying cycles in a directed graph in linear time [151]. If applied to this problem, the continuation method could be i) targeted at only the roads in cycles and ii) guaranteed not to create new cycles.

While estimating the computational complexity of the presented algorithm is outside of the scope of this work, matrix sorting and identifying cyclic constraints are the most computationally complex components of the process. Polynomial time [148] and linear time [151] algorithms have been presented in literature for each function, respectively. Applying these algorithms to this ordering problem, including introducing considerations for continuity, are points of future work and are discussed further in Section 8.4.

5.5 Summary of Contribution

Layer- and surface-based ME processes are able to create feasible (i.e., collision-free) toolpaths by ordering the roads in terms of ascending Z-height, and then optimize road ordering within each layer to minimize deposition head movement. LL-MA deposition, on the other hand, does not have layer-like structures. Therefore, toolpaths must be planned over the whole structure at once and are subject to more complex collision concerns. While some generalized toolpath planning algorithms exist (e.g., [9, 25, 10]), they restrict the ability to freely align deposition directions, which is critical to the mechanical performance of the structure. Other methods do enable the free alignment of deposition directions (e.g., [27, 28, 30]), but their application is restricted to surface geometries rather than truly volumetric parts.
An ordering algorithm capable of explicitly handling the collision concerns arising from un-constrained roads was presented. This method compares each road in the toolpath to determine precedence constraints. An ordering algorithm then satisfies those precedence constraints while attempting to maintain continuous roads and minimize deposition head movement. Although not necessarily an optimal ordering, the algorithm does guarantee a collision-free toolpath by utilizing continuation methods that attempt to i) reorient the build directions of unordered roads and ii) removing low-impact roads from the toolpath.

The algorithm was validated using the propagated quarter-Wheel geometry shown in Figure 4.6. A collision-free toolpath was output, but required the removal of 1312 roads (1225 support and 87 model) and also introduced 1656 discontinuities into the chained roads (out of 3549). Due to the number of removed roads, the algorithm was executed a second time to reduce the number of discontinuities. This second output resulted in 1425 discontinuities, a 25% improvement over the first run, but did result in the removal of an additional 143 support roads.
Chapter 6

Multi-Axis Deposition Platform

The multi-axis ME system used throughout this work is shown in Figure 6.1. It features a 6-DoF robotic arm (ABB IRB1200 7/0.7 [152]) outfitted with a desktop-scale extrusion head and a heated bed. The extruder was modified specifically to better enable multi-axis deposition (Section 6.1). Following toolpath ordering (Chapter 5), travel movements are inserted to facilitate collision-free movement between discontinuous roads (Section 6.2). A framework for translating this completed toolpath to a robot-interpretable toolpath is presented in Section 6.3. Discussions on the integration of the robot with the printing hardware (e.g., the extruder) are presented in Section 6.4. This complete hardware system is then demonstrated on the quarter-Wheel toolpath (Section 6.5). A summary of the contribution is provided in Section 6.6.

6.1 Extruder Design

A standard E3D-v6 extrusion head [153] was modified by elongating the hot-end and sharpening the nozzle to create a smaller interference angle (similar to [154]). This reduces the collision volume (shown in Figure 5.1), enabling a wider range deposition head orientations to be used without colliding with previously deposited material. As a result, the number of collisions found using Algorithm 2 is reduced as compared to a larger interference angle, increasing the overall flexibility of the system. It also reduces the impact of the build direction constraint imposed in Section 2.5.1, as the deposition head can take steeper angles near the build platform without creating collision concerns.

6.2 Travel Movements

Although the roads are assured to be collision-free by the ordering algorithm, the travel movements between discontinuous paths do not have this assurance. The flowchart in Figure 6.2 outlines the insertion of travel movements between two discontinuous roads to ensure the robot does not collide with the part.
Figure 6.1: 6-DoF ME system used to fabricate the example geometry and orientation field. It consists of a robotic manipulator, a desktop-scale extrusion head (inset) that has been modified for LL-MA ME, and a heated build platform.
6.2. Travel Movements

Figure 6.2: High-level algorithm showing the insertion of travel movements between discontinuous roads. If the path is not collision-free with a convex approximation of the current print, the path is iteratively offset along its build direction and bisected until it is collision-free.
A convex approximation of the current print progress is created, and the linear path between the two roads is checked for collisions. If it is collision-free, no further travel movements are necessary as the deposition head can safely move between the two end points. If collisions are found, the travel movement is offset along the build directions of the roads on either side of the travel movement. This offset path is then iteratively bisected and offset, checking for collisions at each iteration until each linear segment of the travel movement is determined to be collision-free. The resulting set of travel movements approximately follows the surface of the geometry.

### 6.3 Toolpath Translation

The result of the toolpath ordering (Chapter 5) is a set of ordered, collision-free roads with associated build directions. The toolpath for the system must be formatted as RAPID code, which is effectively a list of poses for the deposition head in the form of an XYZ-coordinate that describes the position and a quaternion that describes the orientation. Therefore, following the ordering step described in Chapter 5, each road and travel movement is converted into coordinate and quaternion pairs. Figure 6.3 demonstrates the translation from an augmented GCode line containing the necessary orientation information (output from the process planning stage of the workflow) to the analogous RAPID commands.

![Figure 6.3: Example translation from augmented GCode to RAPID. Highlighted are the (blue) point and (orange) quaternion pair defining the pose of the robot, (yellow) extrusion length information, and (green) linear tool speed. Additional call outs are defined in the text.](image)

The point and quaternion pair (blue and orange, respectively) is directly translated to RAPID with minor reformatting. The extrusion information (yellow) is contained in a message that will be communicated to the extruder over ethernet communication (discussed in more detail in Section 6.4). Linear tool speed information (green) is used to control the robot and the extruder.

Additional robot-specific information is also added to the RAPID toolpath. Multiple sets of joint angles can often be used to attain the same pose; the robot configuration is used to disambiguate these sets. External joint positions are unused in this work, as no external DoF are used (e.g., a tilt-turn table). The zone reference describes the freedom allowed to
the robot’s controller during trajectory planning. The “fine” callout refers to forcing the robot to stop at the end of each road; other zone sizes could be used, allowing the robot to move smoothly, but possibly with collisions, between each road. The tool reference frame describes the extruder tip relative to the end effector of the robot, which is brought to each denoted pose in the toolpath. The work object reference frame is defined relative to global space (i.e., the base of the robot) and is used to modify the pose denoted in the RAPID instruction. If for instance, the entire part should be rotated 90°, the work object can be rotated instead while keeping the rest of the toolpath constant (except perhaps the robot configurations). In sum, each “MoveL” command linearly moves the tip of the nozzle to the denoted pose relative to the specified work object. More information on the command structure and RAPID usage can be found in the ABB documentation [155, 156, 157].

6.4 Communication Architecture

The extrusion head is directly controlled by the robot over a TCP/IP socket connection. Commands for the extrusion head are included in the robot’s toolpath, as shown in Figure 6.3 and are transmitted to the extruder just before the robot begins the corresponding movement. This is another motivator for the usage of the “fine” zone reference; it forces the robot to explicitly reach each target before moving onto the next, ensuring accurate extrusion synchronization. If a more flexible zone was used instead, the robot would not necessarily be at the start of the road and could be just before or after the start point when the extrusion command is sent.

6.5 Quarter-Wheel Demonstration Print

The ordered quarter-Wheel toolpath (shown in Figure 5.5) has been fabricated on the multi-axis platform using the LL-MA workflow. During fabrication, the support structure lost significant structural integrity due to i) the sparsity of the roads along the Z-axis and ii) the deletion of a number of support material roads during the ordering step (as described in Section 5.4). These issues caused many of the support material roads to produce nonlinear, nearly sinusoidal depositions. Although undesirable, this did not cause issue with the overhanging geometries in the actual part, as the model material still adhered well to the support structure and cooled in the desired positions.

As will be discussed in Chapter 7, the process parameters were tuned following this print to allow a more dense support structure along the Z-axis while maintaining sparsity in the XY-plane. The deposition head was refined to reduce the collision volume, which is reflected in Figure 6.1. The communication architecture between the robot and the extruder was also improved through the use of ethernet communication (as described in Section 6.4). During
Figure 6.4: Fabricated quarter-Wheel geometry. (a) The support structure was too sparse along the Z-axis to maintain the typical linear roads, resulting in poor deposition quality, but it was still sufficient to support the deposition of the model material. (b) After removing the support structure, the printed part qualitatively matches the input geometry and orientation field.

This print, the architecture featured a handshaking routine that was prone to losing extrusion commands and desynchronizing the two subsystems.

6.6 Summary of Contribution

This chapter presented a deposition platform suitable for the proposed multi-axis deposition strategy including a protocol for translating the ordered set of roads to a robot-interpretable toolpath. The deposition head has been specifically modified for multi-axis deposition and is directly controlled by the robot to ensure robust, synchronized depositions. Movements between discontinuous roads, which are not guaranteed to be collision-free if moved between linearly, are also accommodated. An iterative bisection and offsetting algorithm is used to iteratively shift the travel movement such that it approximately follows the surface of a convex approximation of the in-progress print.
Chapter 7

Fabricating Optimized Layer-less Multi-Axis Composite Structures

The PRH of this work is that LL-MA deposition improves the structural efficiency of printed components by 150% relative to conventional printing strategies. While the presented workflow was evaluated in a planar context in Section 4.5, this chapter focuses on testing this hypothesis for a 3D load case. The Wheel problem was used as the running example through Chapters 3 to 6, but this load case is not suitable for testing on the available mechanical testing equipment. Instead, a modified Wheel problem is used here to facilitate mechanical testing using the same set up shown in Figure 4.9b (Section 7.1). This load case is processed by the presented LL-MA workflow (Section 7.2), and the resulting toolpath is fabricated on the deposition platform that was presented in Chapter 6. The printed parts are mechanically evaluated and compared to geometrically similar structures fabricated using an XY-planar deposition method (Section 7.3). A summary of the contribution is presented in Section 7.4.

7.1 Inverted Wheel Load Case

The Wheel problem is modified as shown in Figure 7.1. The four bottom corners are still fixed, but the load is instead applied to the top surface to facilitate mechanical testing. This load is also distributed in a circle on the top face to prevent the crosshead from slipping during testing (as discussed in Section 4.5). For the purposes of the TO algorithm, the double symmetry across the XZ- and YZ-planes is leveraged to reduce the number of elements required as shown in Figure 7.1b. A mesh of 20x60x50 elements with an edge length of 1 mm was used to model the geometry, resulting in a printed specimen of size 40x120x50 mm.

7.2 Inverted Wheel Toolpath Planning

The load case was optimized using the TO algorithm presented in Chapter 2. The critical toolpath planning parameters are outlined in Table 7.1; the TO parameters were experimen-
Chapter 7. Fabricating Optimized Layer-less Multi-Axis Composite Structures

Figure 7.1: The inverted Wheel loading case is used to validate the LL-MA workflow for true 3D load paths. The load is applied to the top face (rather than the bottom face in the standard Wheel problem) to enable a bending test set up to be used.

tally determined by optimizing multiple geometries using a range of parameters, planning toolpaths for the resulting solutions, and evaluating printability. While an exhaustive study of these effects is out of the scope of this work, the parameters in Table 7.1 produce a geometry that was fully printable (i.e., no roads needed to be deleted). The optimized geometry and orientation field is shown in Figure 7.2; the orientation field approximately follows the density paths, obeying Michell’s theorem.

Table 7.1: Multi-axis toolpath planning parameters for model and support material.

<table>
<thead>
<tr>
<th>General Process Parameters</th>
<th>Model Parameters</th>
<th>Support Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool collision angle</td>
<td>$r_{min}$</td>
<td>$d_{sep}$</td>
</tr>
<tr>
<td></td>
<td>$f$</td>
<td>$d_{sep,xy}$</td>
</tr>
<tr>
<td></td>
<td>Mesh size</td>
<td>$d_{sep,z}$</td>
</tr>
<tr>
<td></td>
<td>Element edge length</td>
<td>$d_{term}$</td>
</tr>
<tr>
<td></td>
<td>$d_{term}$</td>
<td>$d_{term,xy}$</td>
</tr>
<tr>
<td></td>
<td>$L_{min}$</td>
<td>$d_{term,z}$</td>
</tr>
<tr>
<td></td>
<td>$d_{lin}$</td>
<td>$L_{min}$</td>
</tr>
<tr>
<td></td>
<td>$L_{max}$</td>
<td>$d_{lin}$</td>
</tr>
<tr>
<td></td>
<td>2 mm</td>
<td>2 mm</td>
</tr>
<tr>
<td></td>
<td>1.1 mm</td>
<td>2 mm</td>
</tr>
<tr>
<td></td>
<td>0.6 mm</td>
<td>0.7 mm</td>
</tr>
<tr>
<td></td>
<td>0.6 mm</td>
<td>0.6 mm</td>
</tr>
<tr>
<td></td>
<td>5 mm</td>
<td>2 mm</td>
</tr>
<tr>
<td></td>
<td>2 mm</td>
<td>2 mm</td>
</tr>
<tr>
<td></td>
<td>1 mm</td>
<td>10 mm</td>
</tr>
</tbody>
</table>

For process planning, the optimized design space was mirrored across the planes of symmetry before propagating roads. This was to ensure smooth streamlines were generated, as the roads intersecting those planes may otherwise have discontinuous gradients, causing difficulties in the ordering step. Suitable support structure was generated as described in Chapter 3, and roads were propagated through the optimized geometry and support structure using the algorithm presented in Chapter 4. All of the roads, including support structure, were then
7.2. Inverted Wheel Toolpath Planning

Figure 7.2: Optimized design space for the inverted Wheel load case leveraging the symmetry across the XZ- and YZ-planes. The orientation field approximately aligns with the density features, obeying Michell’s theorem.

ordered for deposition as discussed in Chapter 5 (Figure 7.3). An upper triangular matrix (i.e., a viable toolpath) was able to be found, and as previously mentioned, no roads needed to be removed. The continuity matrix shows some continuous roads had to be ordered in a discontinuous fashion; 9179 of 10376 roads were able to be properly ordered sequentially including 7941 of 8756 model roads.

The ordered toolpath is shown in Figure 7.4 where the road deposition order is represented by the color of the road (as identified in the color bar on the right). The locations of the discontinuities are easily identified by the sharp color changes in the inset images. Qualitatively, the ordering is such that support material is deposited in each region until a feature of the desired geometry appears. From there, the geometry is deposited as continuously as possible. In regions where multiple density features converge (e.g., the inset images in Figure 7.4), the roads cannot be ordered continuously due to cyclic precedence constraints in the connected roads.

7.2.1 Conventional Toolpath Planning

For comparison, the same optimized geometry was processed using a conventional slicer (Slic3r [121]) with the parameters shown in Table 7.2. These parameters were selected to be approximately equivalent to the multi-axis process parameters, but a one-to-one matching
Figure 7.3: (a) The ordered collision matrix for the inverted Wheel, and (b) the continuity matrix. While the toolpath is collision-free, not all roads were able to be printed sequentially.

Figure 7.4: Toolpath for the inverted Wheel with roads colored by print order. Inset images highlight regions where roads were forced to be printed discontinuously.
of parameters was not possible. The conventional toolpath is shown in Figure 7.5.

Table 7.2: Parameters used for the conventional inverted Wheel toolpath generated by Slic3r.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer height</td>
<td>0.7 mm</td>
</tr>
<tr>
<td>Number of contours</td>
<td>2</td>
</tr>
<tr>
<td>Infill percentage</td>
<td>100%</td>
</tr>
<tr>
<td>Infill angle</td>
<td>±45°</td>
</tr>
<tr>
<td>Overlap</td>
<td>20%</td>
</tr>
</tbody>
</table>

Figure 7.5: The conventional toolpath (a) with and (b) without support structure.

7.2.2 Toolpath Comparison

The alignment of each toolpath in relation to the optimized orientation field was calculated using the method outlined in Section 4.5.4. As expected, the active alignment method used by VDPG resulted in the multi-axis toolpath achieving a 93.13% alignment to the orientation field. The reduction in the alignment of the multi-axis toolpath relative to the planar toolpath in Section 4.5.4 is hypothesized to be due to the coarser mesh size used in this experiment; exploring this mesh convergence problem is a focus of future work as discussed in Section 8.4. On the other hand, the conventional toolpath only achieved a 54.99% alignment. This poor alignment stems from the inability to align to the Z-component of the load paths (e.g., the main trusses moving from the fixed points to the point of load).

7.3 Fabrication and Mechanical Testing

Each toolpath was fabricated three times on the multi-axis deposition platform presented in Chapter 6 using the CF-PLA material described previously in Section 4.5.2. While the
conventional toolpath could be directly fabricated using the toolpath from the GCode to RAPID methods described in Section 6.3, the multi-axis toolpath required modifications. At various points in the toolpath, especially on the side of the structure nearest to the robot, recovery motions had to be inserted to allow the robot to unwind itself as it approached its joint limits. This introduced additional discontinuities that were not present in the as-ordered toolpath.

Fabricated specimens before and after support material removal are shown in Figure 7.6. In the multi-axis print, support material was generated using a planar seeding strategy instead of the hexagonal pattern shown in Figure 4.2 to enable variable spacing in the XY- and Z-directions. This improved the efficacy of the support structure from the quarter-Wheel demonstration in Section 6.5 such that the first few model roads in the region were placed appropriately. Support material removal was challenging with both toolpaths, but no features appeared compromised in the conventional toolpath. In contrast, the support structure was well adhered to a number of features in the multi-axis toolpath and removal damaged some of the trusses. As will be discussed, none of the affected trusses were on the fracture surface after mechanical testing.

![Figure 7.6: Fabricated multi-axis specimen (a) with and (b) without support; fabricated conventional specimen (c) with and (d) without support. Features in the center of the multi-axis specimen were damaged during the support material removal process.](image)

For the same reasons as described in Section 4.5, direct comparison via stress measurements is forgone in favor of normalizing the maximum load by the mass of the specimen. After taking the mass of each specimen, each was mechanically tested on an Instron 5984 [126] in
three-point bending as shown in Figure 7.7. An extension rate of 5 mm/min was used with a 50 kN load cell.

![Three-point bend test setup for the inverted Wheel geometry.](image)

Figure 7.7: Three-point bend test setup for the inverted Wheel geometry.

The mechanical testing results are shown in Figure 7.8; the conventional specimens had an average mass of 42.31 g and withstood an average maximum load of 1298.04 N while the multi-axis specimens had averages of 43.57 g and 695.61 N, respectively. In terms of normalized maximum load, the conventional specimens withstood 92.23% more load. While this result does not validate the PRH, deficiencies in the multi-axis toolpath can be identified from the fracture surfaces of the multi-axis specimens (Figure 7.9a); discontinuities in the toolpath served to create weakness in critical cross-sections of the geometry.

![Plot of the maximum load normalized by the mass of the specimen for each the conventional and multi-axis toolpath.](image)

Figure 7.8: Plot of the maximum load normalized by the mass of the specimen for each the conventional and multi-axis toolpath. The conventional toolpath achieved a normalized maximum load of 30.68 N/g while the multi-axis toolpath only achieved 15.96 N/g.

The discontinuities introduced to the toolpath in both the ordering step and the validation of the kinematic motion plan were primarily located in the convergent regions of the truss members shown in the inset images in Figure 7.4. As shown in Figure 7.9a, these were the initial points of failure during testing. By repeatedly introducing discontinuities in these regions, those truss members were weakened, causing them to fail prematurely. All of the load was then applied to the main truss arching from the point of load to the four fixed
Figure 7.9: (a) Fracture surface of a multi-axis inverted Wheel specimen. The right portion of the specimen is laying flat, rather than standing as designed. Deposition path fraying can be seen in many of the middle features, highlighting deficiencies in the toolpath and deposition platform. (b) Fracture surface of a conventional inverted Wheel specimen. The failure occurred along the layer interfaces, as anticipated.

points in the bottom corners. Additionally, many roads crossing the fracture surfaces were isolated, rather than forming a consolidated cross-section as would be desired. This is in contrast to the conventional toolpath, which had at least one continuous layer connecting each of the described truss members. As a result, the conventional specimens failed across the layer interfaces as predicted (shown in Figure 7.9b).

While the multi-axis toolpath was able to be ordered without any deletions, discontinuities were introduced to the toolpath at critical cross-sections due to physical limitations of the system, both in terms of deposition head collisions and limits of the kinematic chain. Specifically, subsequent support structure and density features prevented the continuous deposition of multiple truss members that were present in the fracture surface after mechanical testing. A thinner deposition head may have been able to properly (i.e., continuously) deposit those roads. Similarly, the robot frequently reached joint limits when depositing the side of the structure nearest the arm. This required the robot to move away from the specimen, unwind itself, and resume deposition, further introducing discontinuities. Addressing these deficiencies with an improved deposition head and a tilt-turn table (to mobilize the part relative to the robot) are points of future work as discussed in Section 8.4.

7.4 Summary of Contribution

3-DoF ME is insufficient to fully leverage the mechanical properties available with composite materials due to the inability to achieve fiber alignment in the build direction. Multi-axis deposition strategies have the potential to enable fiber alignment in any direction (including the build direction), but current toolpath planning strategies are limited. Specifically, layer- and surface-constrained roads do not enable the alignment of the composite reinforcement
to complex load paths. The presented LL-MA workflow enables material and reinforcement alignment to arbitrary load paths, allowing the composite material to be used more effectively in printed parts.

The PRH for this work was that multi-axis deposition produces a more efficient structure than conventional printing strategies. To test this hypothesis, the LL-MA workflow and a conventional slicing strategy were used to plan toolpaths for the inverted Wheel load case (Figure 7.1). As shown, the LL-MA workflow was able to optimize the geometry and orientation field with respect to the load case, plan roads aligned to that orientation field, and order those paths for collision-free deposition. The resulting toolpath had a 93.13% correlation with the optimized orientation field compared to the 54.99% correlation of the conventional toolpath; this additional alignment was expected to correlate with improved mechanical performance. Despite the optimization parameters being chosen such that the ordering process could be completed without requiring any roads to be deleted, discontinuities had to be introduced to the toolpath due to limitations in the deposition platform. As a result, the conventional toolpath achieved a 92.23% improved normalized maximum load when compared to the LL-MA toolpath (Figure 7.8). Although this does not validate the PRH, enhancing the deposition platform to eliminate the discontinuities in the toolpath is a focus of future work.
Chapter 8

Summary

8.1 Intellectual Merit

The PRH for this work was that aligning roads to 3D load paths through topology and toolpath optimization would increase the structural efficiency of printed parts by 150% relative to conventional printing strategies. In order to test this hypothesis, RQs focused on i) identifying the optimal geometry and orientation field for a specified load case, ii) planning roads aligned to that optimized orientation field, and iii) ordering those multi-axis roads for collision-free deposition were posed. These RQs serve to generalize multi-axis deposition for arbitrary geometries and orientation fields, where prior work in literature was limited to surface-based geometries and orientation fields.

8.1.1 Research Question 1

The load paths in the structure must be identified to facilitate road alignment. RQ 1 seeks to not only identify the load paths within a structure, but also the optimal geometry for the associated load case. While prior literature has explored this problem in planar contexts, a multi-axis context had not been explored.

<table>
<thead>
<tr>
<th>Research Question 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are i) the optimal geometry and ii) optimal material orientations for an arbitrary set of loading conditions for a given design space?</td>
</tr>
</tbody>
</table>

A TO problem formulation was presented to address Objective 1 where two sets of design variables were simultaneously optimized: one controlling the density distribution and another controlling an orientation field (Section 2.3). This orientation field was used to describe a discrete approximation for the load paths acting on the structure.
Objective 1

Identify the optimal topology and 3D orientation field corresponding to an arbitrary set of loading conditions.

Multiple parameterizations of the 3D orientation space exist, and the selected parameterization has a large impact on the computational efficiency of the algorithm and the quality of the final solution. Three parameterizations of the orientation space were explored in this work: 3D Euler angles, quaternions, and natural quaternions. These three parameterizations were compared against an existing planar TO algorithm that considered planar material orientation variation in terms of final solution compliance, iterations to convergence, and computation time.

Objective 2

Identify a computationally efficient orientation parameterization for the 3D material orientation design space.

The natural quaternion parameterization outperformed the other parameterizations in both the Wheel (Figure 2.14) and multi-load (Figure 2.17) load cases. This was due to the natural quaternion retaining the numerical stability of the quaternion parameterization while not requiring a unit length constraint to be applied for each element in the design space. In the Wheel load case, the natural quaternion parameterization resulted in a 38% improvement in final solution compliance relative to the planar orientation optimization technique.

8.1.2 Research Question 2

With the orientation field identified, roads must be planned such that they are aligned to the field. Existing literature has presented methods that require making assumptions of the orientation field based on the contours of the geometry, but they do not scale to 3D nor do they maintain the desired generality for arbitrary geometries and orientation fields. Therefore, RQ 2 focuses on generalizing the support and deposition path planning problems for multi-axis deposition.

Research Question 2

How does LL-MA deposition change the road propagation problem for an arbitrary geometry and orientation field?
While other multi-axis deposition strategies specifically minimize the required support material by selectively choosing build directions, the proposed method for aligning roads constrains the possible build directions for each road. It is therefore often necessary to incorporate dedicated support structure to enable successful fabrication, but the XY-planar method of propagating support is insufficient for planning multi-axis support structures.

**Objective 3**

Propagate suitable support structure for LL-MA structures with constrained and variable build directions.

The outlined method for multi-axis support propagation (Section 3.2) propagates support material according to each voxel’s specific build direction. This ensures each voxel is supported with an appropriate surface for material to be deposited onto. Additional considerations are made such that the interfacing layer of supporting roads are orthogonal to the supported model material. This allows model roads to contact multiple support roads rather than possibly slipping between gaps in parallel support roads. The result of the algorithm is a set of voxels with an associated orientation field.

The orientation fields and geometries associated with both the desired model and required support structure must then be propagated with roads. In order to actively respond to an arbitrary orientation field, a deposition path planning algorithm was presented that leverages existing streamline placement literature (Section 4.3). By iteratively populating the design space with seed points and numerically advecting those points through the orientation field, the resulting set of roads is highly aligned to the input orientation field.

**Objective 4**

Translate arbitrary geometries and corresponding orientation fields into roads.

The presented deposition path planning algorithm was then shown to be equivalent to conventional toolpath planning strategies in terms of volumetric coverage, each achieving approximately 75% coverage. While conventional strategies tended to under-deposit the geometry due to poor deposition packing ratios, the presented method over-deposits the structure, but process parameters could be modified to achieve under-deposition. In terms of alignment to an arbitrary orientation field, the presented method achieved a 98.9% correlation to a multi-load case orientation field while conventional strategies were only able to achieve approximately 70% alignment (Section 4.5). This corresponded to a mechanical property increase of 108.24% and 29.25% in tension and bending, respectively, for the specimens fabricated using the presented toolpath planning method.
8.1.3  Research Question 3

By nature, multi-axis roads change height along their length and have unique build directions, requiring specific consideration to prevent collisions during the printing process. In existing literature, this challenge is overcome by depositing surface-constrained roads in a layer-by-layer fashion, but this limits the potential for road alignment. In this work, the collision considerations between roads are explicitly determined and accommodated to improve process flexibility.

**Research Question 3**

How does LL-MA deposition change the problem of collision-free printing for an arbitrary set of unconstrained roads?

Precedence constraints between roads are identified by taking the Minkowski sum of a convex approximation of the deposition head and a given road. Each other road in the toolpath is then checked against this identified volume for collisions. In this way, an \( n \) by \( n \) collision matrix is defined, where \( n \) is the number of roads in the toolpath. This method assumes the structure is small relative to the size of the deposition head (i.e., collisions with the robotic platform are not considered), but the process should scale to larger parts with additional collision checks.

**Objective 5**

Establish precedence constraints for all of the LL-MA roads given a deposition head geometry.

With the collision matrix identified, coupled row-column transposes can be performed to identify a collision-free toolpath by creating an upper-triangular matrix. This does not consider sequentially depositing roads that share an end point though, which would degrade mechanical properties. Therefore, the algorithm presented in Section 5.3.2 considers i) precedence constraints, ii) continuity of roads, and when possible iii) minimizes the movement of the deposition head to improve print time.

**Objective 6**

Order the roads to satisfy the precedence constraints while promoting continuous depositions.
The collision detection and ordering algorithms were validated using a quarter-Wheel geometry (Section 5.4). Continuation methods were required to plan a collision-free toolpath including i) reorienting the build direction of certain roads and ii) deleting low-priority roads (weighted by length and location in terms of voxel density). Despite this, the resulting toolpath was successfully fabricated, as shown in Figure 6.4.

8.1.4 Research Question 4

With the LL-MA workflow established as a result of RQ 1-3, RQ 4 focuses on testing the PRH through mechanical evaluation of an optimized 3D load case.

Research Question 4

How does LL-MA deposition change the mechanical properties of optimized parts printed via the ME process?

The ordered toolpath, composed of poses for the deposition head, is analogous to a GCode file augmented with orientation information (in the form of a quaternion). This toolpath needs to be converted to a robot-interpretable toolpath for deposition. Methods for this translation and injecting collision-free travel movements were presented in Sections 6.3 and 6.2, respectively.

Objective 7

Convert the ordered set of roads into a list of joint trajectories and extrusion commands for the multi-axis deposition system to execute.

To test the PRH, an inverted Wheel load case (Figure 7.1) was optimized using the presented TO problem statement, and the resulting design space was processed by the LL-MA workflow. A conventional toolpath was generated for the same geometry, but ignoring the orientation field, using analogous process parameters. The resulting toolpaths were then fabricated on the identified multi-axis deposition platform customized for LL-MA (described in Chapter 6).

Objective 8

Fabricate and mechanically evaluate LL-MA geometries and compare them to geometrically similar parts fabricated using the same material and XY-planar layers.
Mechanical testing of the printed specimens demonstrated a 92.23% improvement in the normalized (by weight) maximum load in the conventional specimens relative to the multi-axis specimens. Although this contradicts the PRH, there were major deficiencies in the multi-axis deposition platform that led to this decrease in properties. Specifically, limitations in the deposition head collision volume and the kinematic chain introduced discontinuities in the toolpath. Although no roads were deleted during the ordering step, the convergent regions of the geometry (highlighted by the inset images in Figure 7.4) were unable to be printed continuously. Compounding this issue, the robot frequently hit joint limits when depositing features on the near side of the multi-axis specimens. This caused the robot to move away from the print in order to unwind itself, introducing additional discontinuities. These defects were focused in the features contained in the fracture surface shown in Figure 7.9a.

In contrast, the conventional specimens had at least one continuous layer through each of those critical sections. A more flexible system (e.g., one with a thinner deposition head and additional DoF) should be able to execute the multi-axis toolpath as intended, improving the mechanical properties relative to the conventional method as predicted by the PRH.

### 8.2 Publications

Publications directly associated to the presented work are listed in Table 8.1. Relationships between each paper and their correlating chapter and RQ are identified. The first three publications, although not directly related to any of the chapters in this work, are other works published by the author in the field of multi-axis deposition.

#### Poster Presentations


Table 8.1: Journal and conference publications directly relevant to the presented work with references to the associated chapter and RQ.

<table>
<thead>
<tr>
<th>Chapter</th>
<th>RQ</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>J. R. Kubalak, A. L. Wicks, C. B. Williams, “Aligning Material Extrusion Infill to Stress-based Orientation Fields for Improved Mechanical Performance” (prepared for submission)</td>
</tr>
<tr>
<td>3, 4, 5</td>
<td>2, 3</td>
<td>J. R. Kubalak, A. L. Wicks, C. B. Williams, “Workflow for 3D Deposition Alignment through Layer-less Multi-Axis Material Extrusion” (prepared for submission)</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>J. R. Kubalak, A. L. Wicks, C. B. Williams, “Fabricating Optimized Composite Structures via Layer-less Multi-Axis Material Extrusion” (planned)</td>
</tr>
</tbody>
</table>
8.2. Publications


Oral Presentations


Intellectual Property


Other Publications

Chapter 8. Summary


8.3 Contributions

8.3.1 Intellectual Merits

In creating the generalized multi-axis toolpath planning workflow, the following scientific contributions were made:

- A methodology for incorporating 3D material orientation considerations into TO by directly controlling the orientation field with design variables (Chapter 2).

- A strategy for propagating support structure suitable for multi-axis deposition when considering a multi-axis geometry with predetermined build directions (Chapter 3).

- A methodology for volumetrically propagating roads aligned to an arbitrary specified orientation field (Chapter 4).

- An understanding of the mechanical property improvements possible when considering aligned roads in a planar context (Chapter 4).

- A methodology for ordering multi-axis roads by explicitly considering the precedence constraints between each path (Chapter 5).

- A self-contained multi-axis deposition platform with synchronized extrusion and robotic movement (Chapter 6).
• Demonstration of fabricating an optimized geometry and orientation field in a multi-axis context (Chapter 7).

8.3.2 Broader Impacts

Composites AM is an emerging manufacturing and research area in industries that require competing strength and weight metrics (e.g., aerospace, automotive, prosthetics, athletic equipment). While conventional AM technologies have the desired geometric flexibility, the inherent process anisotropy does not allow printed parts to achieve the necessary strength metrics. Through the nexus of multi-axis robotics, AM technologies, and TO, the presented research enables true 3D fabrication and leverages that anisotropy to improve part performance. Going against conventional design for AM wisdom, more anisotropy in the material results in a greater performance improvement with the presented workflow.

Although this work focuses on improving the stiffness and weight of printed components, the workflow is formulated to be agnostic to the deposition process, material, deposition platform, and performance metrics. The orientation field representation of the desired material orientations enables other metrics (e.g., thermal dissipation, electrical conductance, mode shapes) to define the orientation field and non-mechanical anisotropy (e.g., conductivity) to be preferentially aligned with that orientation field. By combining objective functions, it should even be possible to simultaneously optimize for multiple metrics. This flexibility makes multi-axis deposition more accessible for further research and industrial applications.

Dissemination through Virtual Reality

The complexity of AM structures, particularly the 3D structures resulting from the presented work are challenging to visualize and inspect. This stems from the lack of appropriate tools; the designer must either inspect the geometry and toolpath in a standard computer aided design/manufacturing (CAD/CAM) package or they must print the geometry. While XY-planar toolpaths can be inspected layer-by-layer, it is difficult to visualize the wider (i.e., global) behaviors and effects of the toolpath. This problem is exacerbated in the presented multi-axis geometries as no delineated structures (e.g., surfaces) exist in the toolpath. Virtual reality (VR) technology offers a promising solution – the user can freely explore a customized, immersive 3D environment that pairs well with the 3D nature of multi-axis toolpaths. To this end, a VR environment has been created that enables the user to i) track topological and orientation field evolution throughout the TO process, ii) visualize and inspect deposition path propagation and ordering processes, and iii) visualize and validate the kinematic toolpath planning and final printing processes. Figure 8.1 shows screenshots from different phases of the process planning workflow in the VR environment, using the quarter-Wheel geometry.

Educational and diagnostic information is displayed to the user via panels on the walls. In an
Figure 8.1: VR environment showing the processing planning workflow for the quarter-Wheel geometry: (a) an intermediate TO iteration with load case shown, (b) deposition path planning during an intermediate iteration, and (c) printing the final toolpath with a simulated 6-DoF robot.

In an educational context, this information includes high-level overviews of the relevant processes (e.g., ME, TO, advection). The tool would also be useful in an industrial context, as the user has access to diagnostic information (e.g., solution compliance, average road length, the collision matrix and ordering results, print time, etc.).

The robotic deposition platform used throughout this work frequently appears at outreach events including the Virginia Tech Science Festival, ICAT Day, and the Smithsonian ACCelerate festival. This VR environment will be used to augment those exhibitions with a safer, more explorative, option for attendees. Attendees can learn about the entire process chain, rather than only experiencing the final printing process to get a wider view of the utility and science behind this otherwise complex manufacturing technology.

**Commercialization**

A provisional patent has been developed using the content and results of this research (US 63/030,542), and the underlying technology was used as the focus of a local I-Corps short course. During the three week course, 15 interviews were conducted with aircraft design and manufacturing engineers to identify a product-market fit for the multi-axis manufacturing technology. Following the promising results of the short course, the author anticipates continuing this effort with the national I-Corps program with initial target industries including non-load bearing secondary structures in private aircraft.
8.4 Future Work

The multi-axis mechanical testing results shown in Chapter 7 did not validate the PRH. It is hypothesized that the poor mechanical performance of the multi-axis specimens was due to deficiencies in the multi-axis deposition platform, but further experimentation is required to evaluate this hypothesis. As discussed in Section 7.2, the process parameters for the TO algorithm were chosen to produce a fully printable design but not necessarily one that was able to be printed in a fully continuous fashion. A more comprehensive exploration of the TO process parameters or a new load case may result in a more suitable geometry. This is not a robust solution to this problem though; to preserve the generality of the workflow, process and manufacturing considerations should be integrated back into the TO algorithm to produce an appropriate design for the specific deposition platform. These considerations, as well as other points of future work, are outlined in Figure 8.2 and detailed further in the following sections.

Shape Optimization. The TO algorithm presented in Chapter 2 optimized the structure and orientation field for stiffness rather than strength directly. This is the typical formulation for TO problems, as stress-based optimization introduces a large number of constraints which increase computational requirements [87]. Instead, the near-final topology is determined through TO and a follow up SO, which introduces stress concerns, adjusts the boundaries of the TO geometry [58]. While this was not considered in this work, this additional step would be critical for end-use application.

Smoothing the exterior boundaries of the geometry would also have impacts in the road
propagation step of the toolpath planning process. Due to the discretized nature of the voxel-based geometries, roads are often short and discontinuous near the exterior surface of the geometry (particularly for coarse mesh sizes). The advection process could be relaxed, allowing roads to advect across those exterior boundaries slightly to improve connectivity, but this could negatively impact dimensional accuracy and functional relationship between the input orientation field and resulting set of roads. By modifying the shape through SO, the exterior surface would be smoother, allowing for longer and more continuous roads without affecting the input-output functional relationship.

**Road Propagation.** Extensions to VDPG (Chapter 4), including other seeding strategies, could be explored (e.g., farthest point seeding [114]), as they may produce more continuous streamlines. In the same vein, it may not always be advantageous to select the positive dot product when determining the velocity of the next step (process described in Figure 4.2). There is a tendency for the algorithm to choose similar paths for nearby roads, rather than diversifying the roads along different truss members. It may therefore be advantageous to advect each seed point along both potential velocities at each step to fully explore the possible options. Once all potential paths have been identified, a single, most appropriate path could be selected either through the longest or most different path to ensure adequate coverage in the final toolpath.

**Extended Elastic Matrix.** The TO algorithm presented in Chapter 2 considered the compressive and tensile strengths of the material to be equivalent. It has been shown, for homogeneous ABS, that compressive strength can be up to five times greater than tensile strength [127]. The build direction independence of compressive, torsional, and flexural properties has also not been evaluated. In a previous study by the author [26], tensile properties were shown to be agnostic to the build direction of the specimen, allowing for robust extrapolation of planar mechanical performance to multi-axis deposition. This has not been evaluated for compressive, bending, or torsional loads; further experiments should explore these properties in a multi-axis context to i) demonstrate agnostic mechanical properties (like the tensile tests in [26]) or ii) construct a preliminary model of the effects of the build direction.

The eight-node brick element used to model the structure also introduces errors into the analysis. Specifically, a linear shape function was used to relate nodal displacement and strain. This is not sufficient for modeling bending-dominated load cases (e.g., the distributed bending load in Figure 4.7b); a higher order shape function is required [129]. Additionally, the element formulation used in this work only allows for compliant modes of deformation which over-predicts the element’s stiffness [128]. Future TO formulations should both extend the mechanical property representation in the elastic matrix formulation and use an element that accurately models the loading regimes expected in the structure.
8.4. Future Work

Topology Optimization. Considering the TO algorithm presented in Chapter 2, the number of design variables required to describe each element in the design space is not at a minimum. For a 3D orientation, only three variables are necessary, but due to the usage of natural quaternions, four are used. This increases computational time, and for large problems this additional time can become significant.

Although directly reducing the number of orientation parameters is not possible, it may be possible to integrate the density variable into the mapping of the natural quaternion. Specifically, the natural quaternion mapping allows for quaternions of less than unit length, which scales the material matrix, artificially weakening the element. This is inherently overcome by the optimization process, where those weakened elements are naturally driven to the state of minimum compliance (i.e., a quaternion of unit length). This feature could be leveraged to integrate the density though, whereby the length of the natural quaternion also denotes the density, reducing the total number of design variables required for each element to four (i.e., the minimum representation).

Road Ordering. The current method of ordering the roads is computationally expensive (requiring approximately 90% of the total computation time) and produces better results on subsequent iterations. The current method also results in the removal of roads that should be printable (as evidenced in the second ordering in Figure 5.3). Literature has demonstrated other methods for ordering tasks (e.g., through the Chinese Postman Problem [166] and the Traveling Salesman Problem (TSP) [167]), which could be more suitable for this application. TSP is a well researched graph theory problem that searches for an optimal path that contains each vertex in the graph [168]. This problem has also been generalized such that the optimal path contains each vertex in some subset of the graph, but some of the vertices can be left unvisited [169]. Imeson and Smith then extended the generalized TSP formulation to include constraints (SAT-TSP) that the optimal path must also satisfy [170]. The SAT-TSP expression of the problem has been used for robot task planning [167]. While expressing the presented toolpath planning problem in terms of SAT-TSP is not straightforward, it is expected that the resulting toolpaths would result in reduced deposition head motion relative to toolpaths developed using matrix decomposition.

Alternatively, cyclic precedence constraints could be determined through a graph theory approach [151], enabling efficient addressing. A graph of the continuity between roads could also be incorporated with the precedence constraint graph to improve road continuity. In essence, this method would focus the continuation methods (Section 5.3.2) such that the minimum number of roads are affected.

Intermediate Density Printing. Although the intention in Section 4.5 was to create a non-Michell truss geometry, the geometry was largely a truss-based solution except for the fixed regions of the multi-load case geometry. This is due to the biases in the TO algorithm itself; the penalization of the design space drives the solution to a Michell truss
structure [171]. It is well known that the application of constraints cannot improve the final solution fitness (assuming convergence to the global optimum) [58], therefore these binary solution spaces are almost certainly not optimal. This was recently shown by Sigmund et al.; solutions consisting of intermediate (grey) densities exhibited improved solution fitness when compared to binary design spaces [172]. While fabricating intermediate densities with traditional toolpath planning strategies is challenging, the process parameters of VDPG can be changed on an voxel-by-voxel basis. As such, it may be possible to plan and fabricate toolpaths with intermediate densities, further improving the overall structural efficiency.

**Kinematic Constraint Modeling.** Robotic systems impose complex physical constraints on manufacturability. Collisions between the robot and i) the environment (e.g., the build platform) and ii) the print in-progress must be determined and avoided to ensure a manufacturable result (Chapter 5). A constraint preventing the presence of steep build directions in the final solution was presented in Section 2.5.1, but this constraint formulation does not prevent collisions with the part itself. Literature has presented frameworks for integrating AM manufacturing constraints into TO problem statements including support material minimization [173] and imposing minimum feature sizes [174], but deposition head access constraints (i.e., constraints to prevent collisions between the deposition head and the printed part) have not been explored in literature. Constraints that exclude undercut and internal void features in traditional manufacturing processes [175] filter out unmanufacturable features from the optimized design space, but are not directly adaptable to multi-axis ME. Unlike milling processes, the multi-axis deposition head must have access to the entire internal geometry to fabricate the part; determining which features are printable cannot be determined by looking at feature location and shape relative to the surface of the part.

While the part-deposition head collision concerns evolve over the course of the print, they can be identified and addressed in a similar fashion to static environment collision concerns; collisions can be predicted by determining the volume required by the deposition head and ensuring it is free of obstructions. Although a full kinematic analysis is technically possible during each TO iteration, it would be more computationally expensive than the TO itself. To determine printability in a computationally efficient manner, an integrated mathematical approach to representing these collisions is required.

**Toolpath Considerations Modeling.** The homogenized TO design space must be populated with roads. This violates the homogenization assumption, as the roads are discrete entities; each deposition occupies one or more elements, and each element contains portions of one or more depositions. Unfortunately, optimal packing density of road shapes (i.e., cylinders) is relatively low even with a uniform orientation [123]. With randomly oriented cylinders, this packing ratio drops even lower [176]. Although the roads are not randomly oriented, the packing ratio will be somewhere between these extremes in each element; the mechanical properties assigned to each element by the TO algorithm must correctly model
and predict this behavior in order for the transition to not degrade structural optimality.

**Support Structure.** A large amount of support is generated due to the projection method used in Chapter 3. The support structure shown in Figure 3.2 constitutes approximately 62% of the deposited material. By tuning the assigned build directions for the support structure, it should be possible to reduce the volume of the generated support by leveraging critical deposition angles. Additionally, support structure could extend to and from other geometric features rather than being driven to the bed. Alternatively, a secondary TO could be executed to generate the support structure. While existing literature has explored creating minimal support structure (e.g., [177]), the formulations have only considered a single build direction. With multi-axis deposition, it should be possible to further minimize the amount of support structure by selectively varying the build directions. Implementing these techniques would reduce both print time and the computation time required for road propagation and toolpath ordering.

The support structure is challenging to remove, as discussed in Section 7.3, due to strong bonding between the support and model material. This can lead to some of the thin features in the model (characteristic of Michell truss structures) being damaged during removal. It would therefore be beneficial to adopt infill and contour patterns used in XY-planar support structure that separate the model and support material with an air-gapped contour at each layer. Some extension of this methodology, that does not rely on layer-like structures, should aid in feature survivability.

**Travel Movements.** The travel movement generation method described in Section 6.2 used an iterative offsetting method to create collision-free travel movements between discontinuous roads. While the method did guarantee a collision-free result, it often resulted in unnecessarily conservative and complex motions. Specifically, the travel movements were offset from the geometry conservatively and the quaternion associated with the movement (i.e., the deposition head orientation) often required complex wrist motions to achieve.

Generating effective and efficient travel movements is challenging, but algorithms have been presented in literature for these purposes. Rapidly exploring random trees (RRT) explores the configuration space from some initial position to a target, checking for collisions along the way [178]. RRT produces a collision-free and (relative to a linear path) kinematically smooth robot trajectory but does not guarantee a minimum path length. RRT* introduces the opportunity for reconfiguring the tree to reduce the path length and does guarantee a (probabilistically) optimal, in terms of path length, result [179].

**Large-Scale Printing.** The main goal of this work was to create a generalized process planning workflow for multi-axis deposition. To test the generality of the workflow, other robotic deposition platforms should be used to execute toolpaths. Specifically, the large-
scale pellet-fed platform shown in Figure 8.3 offers two opportunities: i) demonstration on a separate large-scale platform and ii) demonstration of a different deposition process. The system has more stringent process constraints than the demonstrated small-scale system in terms of the achievable deposition head orientations (i.e., the pellet fed system cannot be oriented too steeply with respect to gravity before extrusion fails).

![Figure 8.3: Large-scale pellet-fed robotic extrusion platform (photo credit: Ben Woods)](image)

**Hardware Advancements.** The utility of the multi-axis printing platform (Chapter 6) could be expanded in the form of i) integrating additional DoF to improve the reachable work envelope and ii) incorporating additional functionality and processes, and iii) expanding the material selection. Specifically, a two-axis tilt-turn table (Figure 8.4) has been designed and fabricated in collaboration with Tadek Kosmal and Kieran Beaumont. The tilt-turn serves to add mobility to the build platform, allowing the part to reorient relative to gravity and the robotic platform itself. A deposition head, specifically designed for multi-axis deposition is in development in collaboration with Kelsey Rogers. The head is designed to have a smaller collision volume (e.g., Figure 5.1) to improve the range of accessible orientations and reduce printability concerns. To hybridize the system and improve the surface finish and dimensional accuracy of printed parts, work has also been started (in collaboration with Tadek Kosmal and Kieran Beaumont) towards a milling head for the robotic platform. Additionally, in collaboration with Om Bhavsar, design work has begun on a composite tape-laying head, suitable for multi-axis deposition. These tapes are highly anisotropic and exhibit mechanical properties magnitudes larger than those achieved by commercial (short fiber) composite filaments. In combination with the presented workflow, these composite materials could offer another step-change in the mechanical properties possible via AM processes.
8.4. Future Work

Figure 8.4: Tilt-turn table for adding a 7th and 8th axis to the robotic deposition platform (photo credit: Tadek Kosmal and Kieran Beaumont)
Bibliography


Appendix A

Problem Statements for Specific Parameterizations

In this section, the general TO problem statement used in this work (given in Equation 2.8), is specified in terms of each of the 3D orientation parameterizations.

The rotation matrix \( R_i \) is a function of the orientation design variable vector \( (Q_i) \) associated with element \( i \) using the equations presented in Section 2.2. \( R_i \) is used to calculate the strain transformation matrix \( T_i \), which modifies the element’s elastic matrix as shown in Equation 2.9. The equation for \( T_i \) is shown in Equation A.2 in terms of a general \( R_i \) as defined in Equation A.1 (from [88]).

\[
R_i = \begin{bmatrix}
l_1 & m_1 & n_1 \\
l_2 & m_2 & n_2 \\
l_3 & m_3 & n_3
\end{bmatrix}
\]  
(A.1)

\[
T_i = \begin{bmatrix}
l_1^2 & m_1^2 & n_1^2 & l_1m_1 & m_1n_1 & n_1l_1 \\
l_2^2 & m_2^2 & n_2^2 & l_2m_2 & m_2n_2 & n_2l_2 \\
l_3^2 & m_3^2 & n_3^2 & l_3m_3 & m_3n_3 & n_3l_3 \\
2l_1l_2 & 2m_1m_2 & 2n_1n_2 & l_1m_2 + l_2m_1 & m_1n_2 + m_2n_1 & n_1l_2 + n_2l_1 \\
2l_2l_3 & 2m_2m_3 & 2n_2n_3 & l_2m_3 + l_3m_2 & m_2n_3 + m_3n_2 & n_2l_3 + n_3l_2 \\
2l_3l_1 & 2m_3m_1 & 2n_3n_1 & l_3m_1 + l_1m_3 & m_3n_1 + m_1n_3 & n_3l_2 + n_1l_3
\end{bmatrix}
\]  
(A.2)

Following are the specific forms of the design variable vector \((x)\), the vector of orientation design variables associated with an element \(i\) \((Q_i)\), and the full problem statement. For the sake of brevity, the specific forms of the gradient equations (given in general form in Equations 2.10 and 2.11) are only shown for the Euler angle parameterization, but all parameterizations follow the same structure.
A.1 Euler Angles

\[ x = \{ \rho^T, \theta^T, \phi^T, \psi^T \}^T \]  
(A.3)

\[ Q_i = \{ \theta_i, \phi_i, \psi_i \}^T \]  
(A.4)

The problem statement (Equation A.5) remains largely the same as the CFAO problem statement. The main difference is the additional side constraints imposed by the additional rotation angle variables. To prevent a cyclic representation, each rotation angle is held within the bounds \([-\pi/2, \pi/2]\). These bounds are equivalent values in terms of material orientation; the algorithm is allowed to move across the boundary but the value of the variable remains bounded.

\[
\min_{x} : c(x) = \sum_{k=1}^{N_{lc}} U_k^T K(x) U_k \\
\text{subject to:} \\
\frac{V(\rho)}{V_0} \leq f \\
: K(x) U_k = F_k \\
: 0 < \rho_{\text{min}} \leq \rho \leq 1 \\
: -\pi/2 \leq \theta \leq \pi/2 \\
: -\pi/2 \leq \phi \leq \pi/2 \\
: -\pi/2 \leq \psi \leq \pi/2 \\
\]  
(A.5)

\[
\frac{\partial c}{\partial \rho_i} = -\eta \rho_i^{\eta-1} \sum_{k=1}^{N_{lc}} u_{k,i}^T \int \int \int_{\Omega_i} (B^T T_i^T(Q_i) E_0 * T_i(Q_i) B) \partial \Omega_i u_{k,i} \\
\]  
(A.6)

\[
\frac{\partial c}{\partial \theta_i} = -\rho_i \sum_{k=1}^{N_{lc}} u_{k,i}^T \int \int \int_{\Omega_i} (B^T (\frac{\partial T_i(Q_i)}{\partial \theta_i}) E_0 T_i(Q_i) + T_i(Q_i)^T E_0 \frac{\partial T_i(Q_i)}{\partial \theta_i}) B) \partial \Omega_i u_{k,i} \\
\]  
(A.7)

\[
\frac{\partial c}{\partial \phi_i} = -\rho_i \sum_{k=1}^{N_{lc}} u_{k,i}^T \int \int \int_{\Omega_i} (B^T (\frac{\partial T_i(Q_i)}{\partial \phi_i}) E_0 T_i(Q_i) + T_i(Q_i)^T E_0 \frac{\partial T_i(Q_i)}{\partial \phi_i}) B) \partial \Omega_i u_{k,i} \\
\]  
(A.8)
A.2 Quaternions

\[
\frac{\partial c}{\partial \psi_i} = -\rho_i^T \sum_{k=1}^{N_{lc}} u_{k,i}^T \iint_{\Omega_i} \left( B^T \left( \frac{\partial T_i(Q_i)}{\partial \psi_i} \right)^T E_0 T_i(Q_i) + T_i(Q_i)^T E_0 \frac{\partial T_i(Q_i)}{\partial \psi_i} B \right) \partial \Omega_i u_{k,i} \quad (A.9)
\]

A.2 Quaternions

\[
x = \left\{ \rho^T, \ q_w^T, \ q_x^T, \ q_y^T, \ q_z^T \right\}^T \quad (A.10)
\]

\[
Q_i = \left\{ q_{w,i}, \ q_{x,i}, \ q_{y,i}, \ q_{z,i} \right\}^T \quad (A.11)
\]

The quaternion parameterization necessitates a unit length constraint placed on each element in the design space. This is reflected in Equation A.12.

\[
\begin{align*}
\min_x : c(x) &= \sum_{k=1}^{N_{lc}} U_k^T K(x) U_k \\
\text{subject to : } &\frac{V(\rho)}{V_0} \leq f \\
&K(x) U_k = F_k \\
&\|Q_i\| = 1 \forall i = 1, 2, 3...N_e \\
&0 < \rho_{\text{min}} \leq \rho \leq 1 \\
&-1 \leq q_w \leq 1 \\
&-1 \leq q_x \leq 1 \\
&-1 \leq q_y \leq 1 \\
&-1 \leq q_z \leq 1
\end{align*} \quad (A.12)
\]

A.3 Natural Quaternions

\[
x = \left\{ \rho^T, \ w^T, \ x^T, \ y^T, \ z^T \right\}^T \quad (A.13)
\]

\[
Q_i = \left\{ w_i, \ x_i, \ y_i, \ z_i \right\}^T \quad (A.14)
\]
Chapter A. Problem Statements for Specific Parameterizations

\[
\min_{x} \, c(x) = \sum_{k=1}^{N_{lc}} U_k^T K(x) U_k \\
\text{subject to:} \quad \frac{V(\rho)}{V_0} \leq f \\
\quad K(x) U_k = F_k \\
\quad 0 < \rho_{\text{min}} \leq \rho \leq 1 \\
\quad -1 \leq w \leq 1 \\
\quad -1 \leq x \leq 1 \\
\quad -1 \leq y \leq 1 \\
\quad -1 \leq z \leq 1
\]
Appendix B

Orientation Field Alignment to Principal Stresses

Although the TO algorithm presented in Chapter 2 uses design variables to directly optimize the orientation field of material orientations, it is possible to couple the orientation field to the nodal displacement of each element (e.g., [69, 70]). As discussed in Section 2.4.7, additional design variables are computationally expensive; therefore, this section explores a method that couples the field of material orientations to the principal stresses within each element. The same load cases used in Chapter 2 (i.e., an MBB beam, Wheel, and tensile/bending multi-load case) are optimized with this coupled method.

The mesh convergence for each load case is shown in Figure B.1 for both the natural quaternion parameterization and the principal stress method. There is no appreciable difference between the methods for the MBB beam load case, but for the Wheel and combined load case, the explicit design variable parameterization achieved more performant designs across all of the mesh sizes. This difference was 2.9% and 6.0% at the most refined mesh size for the Wheel and combined load case, respectively.

The cause of this discrepancy is currently unexplored, but is hypothesized to be due to the checkerboarding in the orientation field (shown in Figure B.2). An additional point of note is the larger degree of complexity in the principal stress geometry as compared to the natural quaternion result (Figure 2.15d). The algorithm parameters (e.g., the Heaviside density filter...
radius) were kept constant between the parameterizations, so the cause of this complexity is unknown. It is hypothesized that the orientation field oscillates between iterations due to changes in the displacement field, causing additional (suboptimal) features to appear in the density field.

Figure B.2: Combined load case results using the principal stress method for the most refined mesh size; results exhibit checkerboarding in the orientation field.

A computation time comparison is shown for each load case, mesh size, and orientation parameterization combination in Figure B.3. A linear fit of each parameterization’s computation times is shown for the sake of simplicity. As seen, there is a clear improvement in computation time requirements due to the reduced number of design variables. Compared to the natural quaternion parameterization, the principal stress method requires approximately 60% of the computation time.

Figure B.3: Computation time comparisons between each parameterization, including the principal stress method.

While there were clear computation time improvements with the principal stress method, the quality of the final solution fitness did decrease. More importantly, the coupling of the density and orientation fields decreases the opportunity to constrain the design space in response to manufacturing constraints (as discussed in Section 8.4). It also may affect the generality of the algorithm, as other performance criteria (e.g., mode shapes) may not have a principal stress analog.
Appendix C

Discussion of Stress and Modulus Measurements

This work presents mechanical property data in the context of maximum load normalized by the mass of the printed structure. While there is literature precedent for this type of quantitative evaluation (e.g., [28]), a more compelling and traditional comparison would be to use stress and modulus measurements. As discussed in Section 4.5.3, selection of a cross-section for proper normalization of the data was challenging due to the non-uniformity of the geometry (as opposed to, for instance, a standard tensile bar). For posterity, measurements of the central cross-sections (i.e., the three truss members crossing the center plane of the specimen) were taken and are used as the normalization factors in this section. A summary of the cross-sectional measurements is provided in Figure C.1. Direct measurement of the fracture surface was neglected as i) the fracture surface was difficult to identify as the specimens shattered on failure and ii) the truss members under load changed throughout each test. The resulting stress-strain curves and mechanical property measurements are discussed here to support the use of the normalized maximum load measurement used throughout this work.

Sample stress-strain curves are shown in Figure C.2. The dashed lines denote two different measurements of Young’s modulus: i) the blue line is the linear fit from the first point of load that contains the largest number of data points while maintaining $R^2 > 0.99$ and ii) the orange line represents the tangent modulus as determined using the method described in ASTM E111 [1]. As shown, the two methods produce significantly different measurements of Young’s modulus and as a result, the yield strength. This discrepancy is due to the irregular shape of the specimen; there is not a consistent cross-section being loaded throughout the test. As load is increased, truss members are loaded and reach their elastic limits at different times, leading to difficulties in the estimation of Young’s modulus and therefore yield strength.
Figure C.1: Average cross-sectional area measurements of the specimens produced using each toolpath planning strategy. The cross-section is taken across the central XY-plane (i.e., the sum of the three truss members that cross that plane).

Figure C.2: Stress-strain curves for an example (a) uniform tensile test, (b) contour tensile test, and (c) VDPG bending test. The blue dashed line represents a linear fit with an $R^2 > 0.99$ from the first point of load, and the orange dashed line represents the tangent modulus as determined using the method described in ASTM E111 [1].