UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Tranchage courbe pour la fabrication additive

THÈSE

présentée et soutenue publiquement le 1 december 2022

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Jimmy ETIENNE

Composition du jury

Présidents :	Isabelle DEBLED-RENNESSON	Université de Lorraine, Loria
Rapporteurs :	Stefanie HAHMANN	Université Grenoble Alpes Laboratoire Jean Kuntzmann
	Marco ATTENE	Italian National Research Council, Institute of Applied Mathematics and Information Technologies (IMATI)
Examinateurs :	Isabelle DEBLED-RENNESSON	Université de Lorraine, Loria
	Bernd BICKEL	Institute of Science and Technology Austria
Directeurs :	Sylvain LEFEBVRE	Université de Lorraine, Loria, Inria
	Cédric ZANNI	Université de Lorraine, Loria, Inria

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Résumé

La fabrication additive est un sujet de recherche actif qui a connu de nombreuses améliorations ces dernières années. L'un des objectifs de la fabrication additive est le prototypage avec l'économie de matières premières et de temps comme axe de recherche essentiel. L'impression 3D présente également de nombreux défis qui peuvent entraver la fabrication correcte d'un objet. L'un des défi évident, est l'évidement de pièce, permettant à la fois d'économiser de la matière et dáccellerer les temps d'impression, comme il y a moins de matière à déposer. Cependant, le fait d'économiser de la matière peut entraîner la chute d'une autre partie de l'objet en raison d'un manque de support.

De manière générale, l'imprimabilité d'un objet reste un problème difficile à évaluer en raison des innombrables possibilités de forme que peut avoir un modèle 3D. Des propriétés telles que la solidité et l'intégrité structurelle peuvent alors devenir la source de problèmes lorsque l'impression 3D n'est pas optimisée pour la forme. Il faut donc trouver des méthodes pour prévenir les défauts liés à l'évidement ou à la forme de l'objet elle même. Des structures de support internes et des remplissages épars ont été proposés de différentes manières pour imprimer correctement les surfaces en surplomb et minimiser la quantité de matériel utilisée lors de l'impression. Néanmoins, la réduction de la densité du remplissage d'une impression 3D peut entraîner une réduction de la robustesse. Ce problème peut être partiellement résolu en optimisant la structure du remplissage en y ajoutant des contraintes supplémentaires. Par exemple, nous pouvons utiliser le résultat d'une optimisation topologique ou forcer un dépôt d'impression entièrement continu pour maximiser la structure du remplissage et l'autoadhésion. Chaque chemin de dépôt de matière revet donc une certaine importance, car il impacte l'imprimabilité globale, mais aussi les autres dchemins de dépot a une échelle plus locale. Cela signifie qu'à l'intérieur d'une couche, même entièrement remplie, le choix des chemins et de leurs orientations est

essentiel. Une approche plus générale consisterait à générer de multiples courbes à l'intérieur d'une couche qui suivent aussi fidèlement que possible les trajectoires données par un utilisateur ou un quelconque processus d'optimisation. Cependant, l'impact de l'orientation de la trajectoire n'est pas seulement limité au plan. Une autre question en suspens concerne le schéma d'impression dans la fabrication additive. La plupart des procédés de fabrication additive façonnent des objets en empilant des couches planes de matériau solidifié. En conséquence, les pièces produites présentent un effet dit d'escalier, qui résulte de l'échantillonnage de surfaces inclinées par des plans horizontaux de taille fixe. Cela a un impact négatif sur l'état de surface et la précision d'une pièce, et doit être minimisé. La méthode la plus courament employée et la créations de couches les plus fines possibles, soit sur toute la pièce, soit réparties uniquement sur les zones impactées. En lieu et place de la méthode couche par couche, une meilleure approche pourrait consister à remplir un volume avec des courbes imprimées en 3D. Celles-ci pourraient servir différents objectifs tels que l'amélioration de la qualité de la surface, l'application de contraintes mécaniques ou la réalisation d'objectifs esthétiques.

Cette thèse tente de surmonter certains problèmes intrinsèquement liés à l'impression 3D qui apparaissent lorsque l'on veut améliorer la distribution de la matière pour respecter des contraintes structurelles ou pour améliorer la qualité visuelle des objets imprimés. Une partie de ces problèmes est liée au schema d'impression couche par couche, où chaqune de ces couches est plane. C'est pour cela que nous nous sommes intéressés au dépôt courbe de filament, permettant de sortir du schéma classique, et, autant que possible, améliorer la qualité structurelle ou visuelle des objets imprimés.

Le premier chapitre décrit les technologies de fabrication additive en général avant de se concentrer principalement sur la déposition de filament fondu (FDM) et l'impression hors du plan. La fabrication additive regroupant énormément de technologies, nous listons les principaux procédés avant de rentrer plus en détail dans la fabrication par dépôt de filament. Nous expliquons ensuite brièvement le processus, de la modélisation assistée par ordinateur jusqu'à l'étape d'impression. Après avoir énuméré les différents problèmes qui affectent cette technique et quelques moyens de les résoudre,

on trouve une description plus approfondie des techniques de remplissage. Le remplissage de l'objet peut être plein ou creux selon l'usage que l'on veut faire de l'objet final. Le remplissage plein, comme son nom l'indique, a pour but de remplir totalement la pièce imprimée. Pour cela, les méthodes classiques telles que le zigzag ou le contour parallèle sont généralement employées, mais ont un biais, soit par le choix de l'orientation des zigzags, soit par la forme de la pièce. Il existe donc d'autres méthodes de remplissage dense qui offrent un meilleur contrôle à l'utilisateur. Quand on ajoute des objectifs supplémentaires, comme la réduction du temps d'impression, du poids, ou de la quantité de matière utilisée, le remplissage creux est à privilégier. Il existe différents types de remplissage creux, par exemple les extrusions 2D, qui sont juste l'extrusion d'un motif sur la hauteur; les remplissages qui évoluent dans l'espace comme le remplissage cubique ou gyroïde; des remplissages qui cherchent à évider la pièce au maximum, etc.

Nous abordons ensuite les techniques d'impression hors plan, de ses balbutiements aux algorithmes complets. Pour pouvoir imprimé hors du plan, les machines doivent être conçues ou modifiées pour avoir un dégagement suffisant autour de la buse d'impression. Certaines buses sont même conçues spécialement pour l'impression courbe, en affichant une tête d'impression très pointue et avec un très faible méplat. La modification des imprimantes 3D dites classiques, permet de gagner un peu de liberté sur l'axe Z, ce qui a permis de travailler sur différents points. Le premier, et le plus recherché par la communauté de passionné, et le lissage de la surface. En effet, lorsque les surfaces sont quasiment horizontales, un effet très délétère pour la qualité de surface survient: l'effet d'escalier. Cet artéfact visuel réduit drastiquement la qualité visuelle de l'objet, et aurait pu être évité si l'impression n'avait pas été effectuée plan horizontal après plan horizontal. Tout comme pour les motifs de remplissages, d'autres objectifs peuvent être poursuivis, comme la réduction du nombre de supports, faire un dépôt continu de matière, suivre un champ de direction, ou tout simplement du dépôt libre pour des œuvres graphiques.

Le deuxième chapitre explore plus en profondeur le remplissage de couches planes et propose deux techniques pour améliorer l'orientation des chemins de dépôt. Pour simplifier le problème, nous ajoutons artificiellement

une contrainte sur les couches qui restent planes, mais nous nous éloignons des lignes droites et des techniques simples de détourage.

Dans la première partie du chapitre sur les remplissages dans le plan, nous introduisons une méthode pour produire des courbes dont l'espacement et l'orientation peuvent être contrôlés par des champs de densité et d'orientation donnés en paramètres. Cette méthode est alors utilisée pour créer des motifs de remplissage de densité variable, mais aussi pour des applications d'infographie afin de rendre rapidement des effets de texture tels que des mèches de cheveux et des streamlines. Ce travail a été initialement développé et inclus dans IceSL sous le nom de "remplissage progressif", offrant aux utilisateurs la faculté de créer des remplissages à densité variable. Les variations pouvant être effectuées au sein d'une couche comme au travers des couches. L'intérêt principal de ce motif de remplissage est qu'il permet une gradation de la densité de matière tout en évitant les variations abruptes des méthodes par subdivision classiques. La plupart des approches par subdivision divisant chaque chemin en deux lors du doublement de la densité, ceux-ci créent alors des front de subdivision très marqués. Notre méthode quant à elle permet de choisir le ratio de subdivision, dispersant les lieux où les chemins se divisent.

Dans la deuxième partie sur les remplissages dans le plan, nous adaptons la méthode susmentionnée pour des courbes uniformément espacées sur un plan via un bruit de phasor. Cette méthode est en revanche utilisée pour permettre un contrôle total de l'orientation du dépôt imprimé en 3D. En permettant le contrôle de la direction des chemins de dépôt, cette méthode peut facilement être combinée à, par exemple, une optimisation topologique, afin d'optimiser la direction des différents dépôts. Le choix de l'orientation permet aussi de jouer avec des effets de lumières.

Le troisième chapitre détaille la principale contribution de notre travail. Il traite de l'impression 3D légèrement incurvée et fournit l'un des premiers algorithmes de tranchage courbe pour améliorer la qualité de surface des impressions 3D sur des machines standard à 3 axes. Pour imprimer des chemins courbes, nous tirons parti de la liberté de l'axe Z des imprimantes pour permettre à l'imprimé de sortir légèrement du plan, ce qui nous permet

de suivre de petites pentes. Nous avons exploité cette capacité dáller hors du plan pour accroitre la liberté de déposition et de réduire considérablement l'erreur de reproduction.

La première partie de ce chapitre présente une technique basée sur la déformation spatiale contrainte et permet de réduire l'erreur volumique pour des surfaces presque planes. Les surfaces quasiment planes présentent la plus grosse erreur volumique, notre objectif est donc de génerer des chemins de dépostions qui suivent la surface de l'objet au lieu du plan de coupe. Pour ce faire, nous calculons une déformation de l'espace, dans lequel nous tentons d'aplanir et dáligner les surfaces afin de pouvoir utiliser les trancheurs classiques. Une fois l'objet tranché dans l'espace déformé, nous ramenons les chemins obtenus dans l'espace d'origine afin de l'imprimer. Pour calculer cette déformation, nous utilisons un optimiseur que nous parametrons avec pour contrainte la fabricabilité, assurant qu'il n'y ait pas de collision, et en objectif de minimiser l'effet d'escalier. Les résultats de cette technique sont plutôt excellents, avec une réduction allant jusqu'à 90% lorsque les surfaces peuvent être bien optimisées.

Dans la seconde partie, nous avons choisi d'étendre le remplissage dense de Phasor à la 3D et donner la priorité au contrôle des trajectoires. L'objectif ici est donc d'ordonner les trajectoires pour garantir l'absence de collision entre le dispositif d'impression et les pièces imprimées. Le principal apport de cette partie est un algorithme de découpe et d'ordonnancement des tranches, mais il ne s'agit ici que d'un travail préliminaire.

Mots-clés: Impression courbe, impression 3D, FDM

Abstract

Most additive manufacturing processes fabricate objects by stacking planar layers of solidified material. As a result, produced parts exhibit a so-called staircase effect, which results from sampling slanted surfaces with horizontal planes. This negatively impacts the surface finish and accuracy of a part. While thinner slices reduce this effect, it remains visible in areas where the input shape surfaces almost align with the layers. This horizontal slicing scheme also impacts the resilience of the printed part as layers cannot be aligned to obtain the maximum strength. As with layers, the orientation of trajectories within a slice is often constrained and cannot be freely controlled.

In this thesis, we exploit the ability of some additive manufacturing processes to deposit material slightly out of the plane to overcome these limitations. We mainly focus on extrusion-based technologies, particularly Fused Filament Fabrication technology, since most printers in this category can deposit along slightly curved paths under deposition slope and thickness constraints. Our algorithms are split into two categories, the ones that produce freely oriented trajectories inside a layer and the ones that curve the layers themselves.

My first contribution focuses on deposition trajectories inside a layer, allowing the users to control their orientation. This led to two novel infill patterns aiming at two different objectives. The first is a sparse infill that follows a direction field and density field, while the second is a dense, oriented staggered infill pattern with minimal porosity.

My second contribution focuses on printing with curved layers, exploring two different approaches. The first one operates directly on the layers, making them either follow the natural slope of the input surface or, on the contrary, intersect the surfaces at a steeper angle, thereby improving the sampling quality. We demonstrate that this approach enforces all fabrication constraints, including the guarantee of generating collision-free toolpaths. The second method builds atop the staggered infill introduced before, generating trajectories with free orientation throughout the part's volume.

Keywords: Curved printing, 3D printing, FDM, FFF

Remerciements

Je tiens en premier lieu à remercier mon premier directeur de thèse, Sylvain LEFEBVRE, pour tout le temps passé ensemble, parfois à écrire, souvent à débugger, mais surtout à réfléchir. Merci pour tout ce que tu m'as appris, et de m'avoir permis de découvrir le monde de la recherche. Je te remercie de m'avoir pris en stage et d'avoir fait tout ton possible pour que je puisse faire une thèse.
Je remercie aussi Cédric ZANNI, mon second directeur de thèse, de m'avoir accompagné et aidé dans la rédaction des articles et de cette thèse. J'aurais vraiment aimé travailler plus avec toi sur des sujets que nous avons en commun, mais il y a déjà tellement à faire. Si je devais changer quelque chose dans l'organisation de ma thèse, ça serait d'avoir passé plus de temps sur les squelettes et axes médians avec toi.

Je remercie le CNRS et le LORIA pour le financement et l'hébergement au sein de ce magnifique laboratoire.
Évidement, ce travail n'aurait pas pu se faire sans mes collègues.
Je remercie Noémie VENNIN, pour sa bonne humeur, son entrain, et surtout pour son assurance en anglais. Dommage que tu sois partie trop tôt (pour travailler ailleurs).
Je remercie également Pierre BEDELL et Pierre-Alexandre HUGRON pour l'aide apportée lors de cette thèse, le support technique et l'impression des résultats.
A Salim PERCHY, un gran agradecimiento, unas notas de guitarra y unos pasos de salsa por todo el tiempo compartido en la oficina.
Je remercie encore tous les autres membres de l'équipe MFX, ils le méritent bien.
Je remercie certains de mes enseignants, qui sont devenus pendant un temps mes collègues, avec une place spéciale pour Dmitri SOKOLOV et Nicolas RAY. Merci de m'avoir pris en initiation à la recherche, merci de m'avoir traumatisé aux pauses café, et non, je n'ai pas de blagues en tête, pas même des carambars.

Pour le soutien moral, les barbecues et les mojitos, je souhaite remercier mes amis, collègues et colocataires. Tout spécialement Yassine EL KHADIRI, Lucien RENAUD, Tatianna ZUNIGA, Clélie AMIOT, Théo EISENBARTH, Marion COCCHI, Maxime SCHOEPPS, Solène GORSZCZYK ... La liste, si tant est que je réussisse à n'oublier personne, serait bien trop longue, mais pour tout le temps passé ensemble, un gros merci.

Et enfin, la dernière partie et pas des moindres, ma famille, sans qui je ne serais pas là où j'en suis. En tout premier, Nicolas GAUVILLE, qui arrive à me supporter presque quotidiennement depuis déjà quelques années. Et pour le record de temps à me supporter, je ne peux que remercier mes parents, ma sœur et mon frère. Et pour avoir cru en moi inconditionnellement et toujours avoir pris ma défense, tata Luce, tata Ziquette, je vous remercie de tout mon cœur.

Contents

I General Introduction **1**

II Related work **9**

Introduction **13**

1 Additive manufacturing technologies **15**

1.1 Powder Bed Fusion 16

1.2 Material Jetting 16

1.3 Binder Jetting 17

1.4 Material Extrusion 17

1.5 Directed Energy Deposition 18

1.6 Sheet Lamination 18

1.7 Vat Polymerization 18

2 Slicing pipeline **21**

2.1 Computer-Aided Design 22

2.1.1 Modeling and shape representation 22

2.1.2 Tesselation 23

2.1.3 Polygon mesh requirements 23

2.2 Computer-Aided *Manufacturing* 23

2.2.1 Slicing step 24

2.2.2 Toolpath generation 26

3 Fused Filament Fabrication **27**

3.1 Fused Deposition Modeling printers 27

3.1.1 Anatomy of a 3D printer 28

3.1.2 Standard 3-axis machines 31

3.1.3 Multi-material and multi-color printing . . . 34
3.1.4 Four and more axis machines . . . 36
3.2 Constraints, defects and overcoming them . . . 37
3.2.1 Stringing defect . . . 37
3.2.2 Staircase defect . . . 37
3.2.3 Overhangs and islands . . . 38
3.2.4 Delamination . . . 41
3.2.5 Shrinkage and warping . . . 41
3.2.6 Seams . . . 42
3.2.7 Printing time . . . 42

4 Infills **45**
4.1 Dense infill . . . 45
4.2 Sparse infill . . . 47
4.2.1 2.5D infills . . . 48
4.2.2 3D infills . . . 48

5 Out-of-plane printing **55**
5.1 Machines and modifications . . . 55
5.2 Conventionnal printers . . . 58
5.2.1 Smoothing the surface . . . 58
5.2.2 Printing lattice structures . . . 58
5.2.3 Following a field . . . 59
5.2.4 Spiralize, helical or continuous . . . 59
5.2.5 Artistic work . . . 60
5.3 Experimental approaches and partially curved layers . . . 61
5.3.1 Partially curved layers . . . 61
5.4 Fully curved algorithms . . . 61
5.4.1 Curved printing pipeline . . . 62
5.4.2 Shell-like prints . . . 62
5.4.3 Supportless printing . . . 63
5.4.4 Wiremeshes . . . 65

III Adaptive infills **69**

Introduction **73**

6 Procedural band patterns **75**

6.1 Method . . . 76

6.1.1 Overview . . . 76

6.1.2 Procedural bands . . . 76

6.2 Controls and parameters . . . 79

6.3 Pseudo-code . . . 80

6.4 Results . . . 83

6.4.1 Bands and curves . . . 83

6.4.2 Animated patterns . . . 83

6.4.3 Infill patterns for 3D printing. . . . 84

6.5 Limitations . . . 88

6.6 Conclusion . . . 88

7 A brick in the wall: Staggered orientable infills for additive manufacturing **91**

7.1 Background on phasor noise . . . 92

7.2 Our approach . . . 93

7.2.1 Defining trajectories . . . 93

7.2.2 Periodic variable fields . . . 95

7.2.3 Graph extraction . . . 95

7.2.4 Clipping and ordering trajectories . . . 96

7.2.5 Staggered bead layout . . . 97

7.3 Results . . . 97

7.3.1 Comparison to curve hierarchies . . . 98

7.3.2 Singularities, staggered layout . . . 98

7.4 Discussion and conclusion . . . 104

Conclusion **107**

IV Curved printing **109**

Introduction **113**

8 CurviSlicer: Slightly curved slicing for 3-axis printers **115**

8.1 Overview . . . 116
8.1.1 Fabrication constraints . . . 116
8.1.2 Improving surfaces . . . 117
8.1.3 Choosing whether to flatten surfaces . . . 118
8.2 Algorithm . . . 120
8.2.1 Main algorithm . . . 120
8.2.2 Fabrication constraints . . . 121
8.2.3 Objective function . . . 122
8.2.4 Relaxing flatness constraints . . . 124
8.2.5 Applying the mapping and its inverse . . . 125
8.2.6 Controlling the number of layers . . . 125
8.3 Results . . . 127
8.3.1 Printed models . . . 127
8.3.2 Influence of optimization parameters . . . 128
8.3.3 Quality . . . 128
8.3.4 Discussion . . . 128
8.4 Discussion and future work . . . 137

9 Another brick in the wall: Orientable trajectories for additive manufacturing **139**

9.1 Multi-axis print constraints . . . 139
9.2 Our approach . . . 140
9.2.1 Periodic variable fields . . . 140
9.2.2 Defining trajectories . . . 140
9.2.3 Graph extraction from a phasor field . . . 141
9.2.4 Layer extraction . . . 143
9.2.5 Path ordering . . . 143
9.2.6 Avoiding collisions . . . 146
9.3 Conclusion . . . 146

Discussion and future work **149**
1 Layer unstacking 149
2 Layers from Procedural Band Pattern 149
3 Layers based on a mapping 149

V Conclusion and perspectives **151**

Bibliography **155**

Acronyms

ABS Acrylonitrile Butadiene Styrene.

AM Additive Manufacturing.

CAD Computer-Aided Design.

CAM Computer-Aided Manufacturing.

CLAD Construction Laser Additive Directe.

CLFDM Curved Layer Fused Deposition Modeling.

CLFFF Curved Layer Fused Filament Fabrication.

CLIP Continuous Liquid Interface Production.

CNC Computer Numerical Control.

CPU Central Processing Unit.

CSG Constructive Solid Geometry.

CT Computed Tomography.

DED Directed Energy Deposition.

DLP Digital Light Processing.

DMLM Direct Metal Laser Melting.

DMLS Direct Metal Laser Sintering.

DoF Degrees of Freedom.

EBM Electron Beam Melting.

FDM Fused Deposition Modeling.

FFF Fused Filament Fabrication.

GLSL OpenGL Shading Language.

GPU Graphics Processing Unit.

IDEX Independent Dual Extrusion System.

LOM Laminated Object Manufacturing.

MMU Multi-Material Unit.

NURBS Non-Uniform Rational B-Spline.

PBF Powder Bed Fusion.

PC PolyCarbonate.

PEEK PolyEther Ether Ketone.

PEI PolyEtherImide.

PLA PolyLactic Acid.

PVA PolyVinyle Alcohol.

QP Quadratic Programming.

SHS Selective Heat Sintering.

SLA stereolithography.

SLS Selective Laser Sintering.

STL Standard Triangle Language, Standard Tessellation Language or STereoLithography file format.

TPU Thermoplastic PolyUrethane.

TSP Travelling Salesman Problem.

UAM Ultrasonic Additive Manufacturing.

UM2 Ultimaker 2.

UV UltraViolet.

VR Virtual Reality.

List of Figures

General Introduction **1**

1 Example of 3D printed objects in cities (a bridge and a house). 3
2 Example of technical prints, a rocket tank and a hip implant. 4
3 Plate printed using Procedural Band Pattern, exhibiting a gradation of material density 5
4 Gears with a phasor infill with path orientations choosen to have the longest paths. 6
5 Comparison of adaptative slicing and curved slicing. 6

Related work **9**

Additive manufacturing technologies **15**

6 Puzzle, where each layer has to be put on a specific order to solve it. . . . 15
7 Power bed fusion technology. 16
8 Material jetting technology. 16
9 Binder jetting technology. 17
10 Material extrusion technology. 17
11 VAT polymerization technologies. 19

Slicing pipeline **21**

12 Slicing pipeline, from conception to instruction generation and then reification. 21
13 Multiple shape representations of a sphere. 22
14 CSG operations 23
15 Over or underxtrusion at turns in FDM 25

Fused Filament Fabrication **27**

16 Common parts of an FDM printer 27
17 Anatomy of the material extruder 29
18 Different sort of printers, a Cartesian, a Polar and a Delta. 31
19 Different cartesian machines. 32
20 A conveyor belt printer with tilted nozzle. 33
21 Different color prints 34
22 Multicolor print methods, using dithering or varying height and transluency. 35
23 Picture of a diamond nozzle able to mix three filaments. 36
24 Stringing defect during travels. 37

25 Staircase defect at every layer. . . . 37
26 Print with overhanging features without support. . . . 38
27 Print with overhanging features cut to be printed without supports. . . . 41

Infills **45**
28 Different dense infill patterns. . . . 46
29 Dense Hamiltonian cycle infill with evolving properties. . . . 47
30 Different sparse infill patterns with 10% density. . . . 49
31 Two homogeneous infill in the main axes. The Cubic infill is a closed-cell pattern, which can trap unsolidified material (e.g. powders or resines) while the Gyroid infill is an open-cell pattern. . . . 50
32 Iterative carving results examples. . . . 51
33 Folding finger printed with a Voronoi foam. . . . 52
34 Star-shaped structure with gradually evolving distance. . . . 53

Out-of-plane printing **55**
35 Simple comparison of curved and planar printing. . . . 55
36 Different machine architectures and modifications. . . . 57
37 Comparison between a classical nozzle and an interference-free nozzle. . . . 58
38 Comparison between standard and smoothed surface generation. . . . 59
39 Example of aesthetic 'curved' prints. . . . 60
40 Chair designed by Zaha Hadid Architects with Nagami where we can see the curved layers. . . . 63
41 Stanford bunny model fabricated with different build directions. . . . 64
42 Support-free volume printing VS standard print with supports 65
43 Bottle printed with WirePrint. . . . 66

Adaptive infills **69**
44 A screwdriver handle with enhanced grip. . . . 73

Procedural band patterns **75**
45 Procedural Band Patterns teaser showing different results. . . . 75
46 Finer and coarser band for a specific density. . . . 77
47 Interpolation between different densities. . . . 77
48 Computation of band IDs after interpolation. . . . 78
49 Alignement removal by random shifts of the bands. . . . 79
50 Result using complex orientation and density fields. . . . 83
51 Printed part exhibiting progressive changes in density to improve strength. 86
52 Printed part exhibiting progressive changes in density. . . . 87
53 Rendered GCode with density varying from 50% to 5%. . . . 87
54 Various effects generated with Procedural Band Patterns. . . . 89

A brick in the wall: Staggered orientable infills for additive manufacturing **91**
55 Overview of paths and printed result with a dense infill. 91
56 Representation of the graph extraction process. 94
57 Extraction of a valid vertex using the marching square algorithm. 95
58 Trajectories shifting at each layer. 96
59 Comparison of the density between a subdivision technique and ours. . . 99
60 CuteOcto model sliced with a radial direction field, constant target beamwidth, and staggered beam layout. 100
61 First slices of the CuteOcto model with constant and varying target bead width. 100
62 A sectional view of the beads within the CuteOcto model volume, under a radial direction field. 101
63 Deposition width controlled by an additional field. 102
64 Controlled surface roughness to create a grip-like surface. 102
65 Direction field can be used to texture horizontal area by using the fact that filament deposition direction is clearly visible on 3D printed models. . . 102
66 Gecko object exhibiting a branching structure. 103
67 Example of a controlled field to fill a mechanical part. 103

Curved printing **109**

CurviSlicer: Slightly curved slicing for 3-axis printers **115**
68 Overview of Curvislicer pipeline 117
69 Sideview of an object sliced with uniform slicing. 118
70 Sideview of an object sliced with curved slicing. 119
71 Moving the vertices of a tringle rotates its normal. 124
72 Comparison of volumic error between uniform slicing and our method. . . 130
73 A frog printed with adaptive slicing and curved slicing. 131
74 Small frog printed with layers at the top smoothly following the curvature. 131
75 Curved layers height smoothly adapting to follow the object's shape. . . . 132
76 Good repartition of layers around the hood of the car, reducing the staircase deffect. 133
77 Optimization of kitten model for different angle constraints. 133
78 Optimization of the frog model for different min/max thickness ratios. . . 134
79 Optimization of the frog model with varying layer heights. 134
80 Comparison of uniform slicing, optimal adaptive slicing and curved slicing for five different models. 134
81 Different toolpath orientations for the top layers in curved slicing result in different surface finish. 135

Another brick in the wall: Orientable trajectories for additive manufacturing **139**
82 A comparison between the fully-traversal layer scheme and partial layering. 140
83 GCode generated with our technique. 147

Conclusion and perspectives **151**

Part I

General Introduction

Introduction

In its broadest sense, manufacturing involves transforming or modifying raw materials to produce new objects using a technical process. Industrial manufacture of parts is traditionally done by subtraction of material or molding.

For instance, the first step in molding is to create a mold, followed by pouring material into the mold, e.g., through injection, centrifugation, or other processes. Finally, the mold is removed. This process is well suited for mass production. Depending on the method, the mold can be reused, as with die-casting, but with sand casting processes, molds are destroyed, and a new mold has to be fabricated.

In *subtractive* manufacturing, the process starts from a block of raw material, and then excess material is removed. The main techniques used are cutting, boring, drilling, and grinding. A computer usually controls these processes, commonly called a Computer Numerical Control (CNC) machine.

The fabrication usually starts with a virtual 3D object that is loaded onto another software meant to generate machine instructions. This often requires human supervision to choose parameters specific to the manufacturing process. The resulting machine instructions are then fed to a machine that performs the according instructions (e.g., cutting, drilling and sanding). This last stage requires little to no human supervision, thus making it suitable for mass production.

These manufacturing processes have significant cost overheads when prototyping. It is not uncommon for engineers and design teams to iterate over various prototypes to gather feedback and validate design choices in order to meet user specifications.

Figure 1: Example of 3D printed objects in cities. ***Left:*** *3D printed bridge in Amsterdam designed by Joris Laarman Lab and printed at MX3D. © PersianDutchNetwork, CC BY-SA 4.0* ***Right:*** *A 3D-printed concrete house © Mense-Korte GbR*

Additive manufacturing

In contrast to subtractive processes, Additive Manufacturing (AM) constructs an object by depositing material to approximate its shape.

To achieve the aforementioned, the creation of an object is divided in 'layers' that, when joined together, form up the resulting object. These layers are built in a specific direction (e.g., from top to bottom) to discretize the shape and develop a manufacturing strategy.

Regardless of the approach's simplicity in Additive Manufacturing, its practical applications are numerous. Compared to subtractive manufacturing or molding, Additive Manufacturing, speeds up prototyping in the long run.

Additive Manufacturing also allows for producing complex object shapes (e.g., hollowed objects) without requiring additional steps like assembly.

Apart from prototyping, AM is also used for final products. A particular technology of AM that is very widespread is called 3D printing. It is popular across several professions (e.g., artists, engineers, designers) and has an affordable entry-point cost. Many complex objects have been 3D printed with success (see Figures 1 and 2).

Figure 2: Example of technical prints. ***Left:*** *Rocket tank being printed with a robotic arm.* ***Right:*** *A 3D-printed hip implant with a porous trabecular structure, which allows for the patient's bone tissue to integrate with the part. © Limacorporate*

Objective

Additive Manufacturing (AM) is an active research topic and has seen many improvements in recent years. As stated before, one of the objectives of AM is prototyping; thus, saving raw materials and time is an essential axis of research. 3D printing also brings about some challenges that might hinder the correct manufacture of an object. For example, hollowing the print to save on material might cause some other part of the object to fall due to lack of support.

Printability of an object remains an issue due to the countless possibilities a shape can take in a 3D model. Properties such as sturdiness and structural integrity can become the

source of problems when the 3D print is not optimized for the shape. Inner support structures and sparse infills have been proposed in different ways to correctly print overhanging surfaces and minimize printing material. Notwithstanding, reducing the density of the infill of a 3D print could result in reduced sturdiness. This is addressed by optimizing the infill structure under additional constrains. For example, we could use a topological optimization or use a fully continuous printing deposition to maximize self adhesion. This means that inside a layer, even a fully-filled one, choosing the paths and their orientations are essential. A more general approach would be to generate multiple curves within a layer that follow as closely as possible trajectories given by a user or any optimization process. However, the impact of the path direction is not only limited to the plan. Another upstanding issue concerns the layer-by-layer printing scheme in AM. This approach cannot accurately reconstruct the surface due to the quantization of the 3D shape in its height axis and the fixed size of each layer. Instead of the layer-by-layer method, a possible better approach could be to fill a volume with 3D printed curves. These could serve different purposes such as improving surface quality, enforcing mechanical constraints or fulfilling aesthetic goals. It is the object of this thesis to study these printed curves.

Contributions

This thesis attempts to overcome some problems intrinsic to 3D printing that appear when we want to improve the distribution of material and stresses inside a part or improving the visual quality of the printed objects.

The first chapter describes additive manufacturing technologies in general before focusing mainly on Fused Deposition Modeling (FDM). After listing the different problems that affect this technique and some ways to solve them, there is a more in-depth description of infills techniques. We also discuss out-of-the-plane printing techniques, from its infancy to complete algorithms.

The second chapter explores more in-depth planar layer infilling and proposes two techniques to improve the orientation of the deposition paths. As a constraint, layers remain flat, but we move away from straight lines and simple contouring techniques.

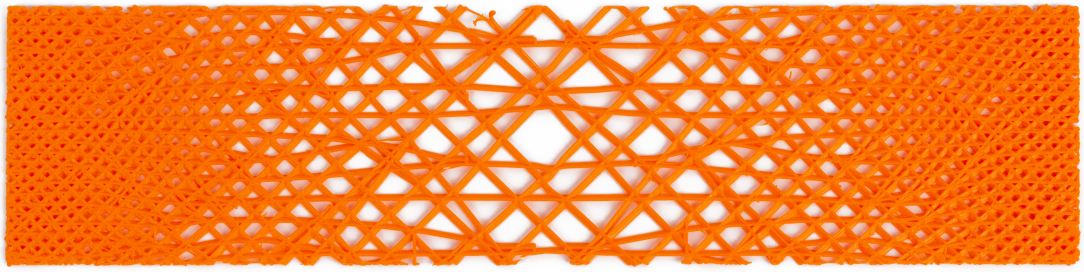


Figure 3: Plate printed using procedural band patterns [Etienne and Lefebvre, 2020], exhibiting a gradation of material density.

In the first part, we introduce a method to produce curves over a plane where their spacing and orientation can be controlled by parameters of density and orientation fields. This is used for creating infill patterns of variable density (Figure 3) but also for computer graphics applications for fast rendering texturing effects such as hair strands and streamlines.

In the second part, we adapt the aforementioned method for uniformly spaced curves over a plane via a *phasor noise*. This is in contrast used to allow full control over the orientation of 3D printed deposition (Figure 4).

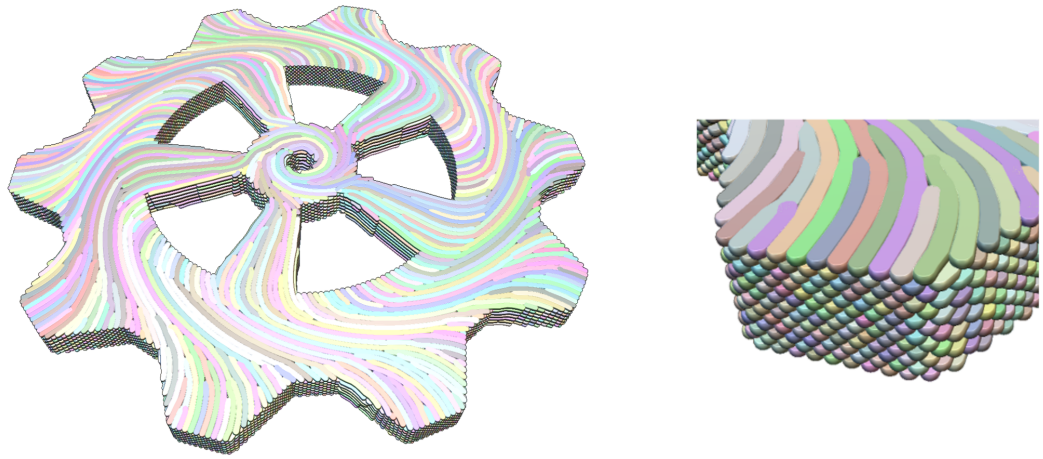


Figure 4: Gears with a phasor infill with path orientations choosen to have the longest paths [Tricard et al., 2021].

The third chapter details the main contribution of our work. It deals with *slightly* curved 3D printing and provides one of the first curved slicing algorithms to improve surface quality of 3D prints on standard 3-axis machines. We take advantage of the leveraging of the Z-axis freedom of the printers to allow the printed to slightly go out of the plane (example Figure 5). This allows us to follow small slopes and considerably reduce reproduction error. The technique is based on constrained space deformation and it can reduce volumic error up to 90% for almost flat surfaces. We also describe how to extend the Phasor dense infill to 3D and order the paths to ensure no collision between the printing device and printed parts.

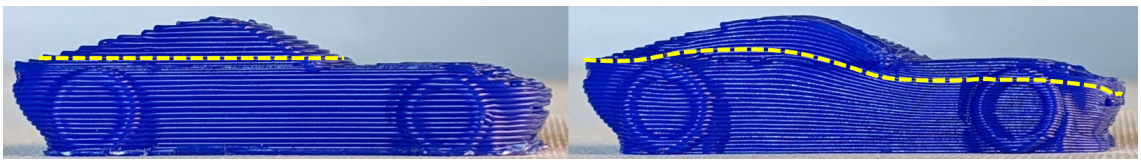


Figure 5: Comparison of adaptative slicing and curved slicing. ***Left:*** *car printed with flat layers.* ***Right:*** *car printed with curved layers [Etienne et al., 2019].*

Publications and softwares

Publications

- **Procedural band patterns**
  J Etienne; S Lefebvre
  I3D '20: Symposium on Interactive 3D Graphics and Games,
  May 2020, Article No.: 1, p. 1-7,
  https://doi.org/10.1145/3384382.3384522

- **A brick in the wall: Staggered orientable infills for additive manufacturing**
  T Tricard; J Etienne; C Zanni; S Lefebvre SCF '21: Symposium on Computational Fabrication,
  October 2021, Article No.: 7, p. 1–8,
  https://doi.org/10.1145/3485114.3485117

- **CurviSlicer: Slightly curved slicing for 3-axis printers**
  J Etienne; N Ray; D Panozzo; S Hornus; CCL Wang; J Martínez; S McMains; M Alexa; B Wyvill; S Lefebvre
  ACM Transactions on Graphics
  Volume 38, Issue 4, July 2019, Article No.: 81, p. 1–11,
  https://doi.org/10.1145/3306346.3323022

Accesible code and softwares

- **Procedural band patterns**
  The shaders used in procedural band patterns are available in Shadertoy, a website allowing the visualization and sharing of GLSL shaders.
  https://www.shadertoy.com/user/I3D20_23

- **CurviSlicer: Slightly curved slicing for 3-axis printers**
  The complete code of this paper is available on GitHub. There are two branches; the main branch is for most users, as it is compatible with free optimization software (OSQP). The second branch is the code run for the paper's results.
  https://github.com/mfx-inria/curvislicer

- **Contributions to IceSL**
  Implementation of procedural band patterns technique for additive manufacturing is available in IceSL, a slicing program, as "progressive infill." To improve surface quality for curved slicing, I also implemented the "ironing" feature in IceSL, doing additional passes on the top surfaces to iron out the deposition paths and some other infills.
  https://icesl.loria.fr/

Part II

Related work

Contents

Introduction **13**

1 Additive manufacturing technologies **15**
1.1 Powder Bed Fusion . . . 16
1.2 Material Jetting . . . 16
1.3 Binder Jetting . . . 17
1.4 Material Extrusion . . . 17
1.5 Directed Energy Deposition . . . 18
1.6 Sheet Lamination . . . 18
1.7 Vat Polymerization . . . 18

2 Slicing pipeline **21**
2.1 Computer-Aided Design . . . 22
2.1.1 Modeling and shape representation . . . 22
2.1.2 Tesselation . . . 23
2.1.3 Polygon mesh requirements . . . 23
2.2 Computer-Aided *Manufacturing* . . . 23
2.2.1 Slicing step . . . 24
2.2.2 Toolpath generation . . . 26

3 Fused Filament Fabrication **27**
3.1 Fused Deposition Modeling printers . . . 27
3.1.1 Anatomy of a 3D printer . . . 28
3.1.2 Standard 3-axis machines . . . 31
3.1.3 Multi-material and multi-color printing . . . 34
3.1.4 Four and more axis machines . . . 36
3.2 Constraints, defects and overcoming them . . . 37
3.2.1 Stringing defect . . . 37
3.2.2 Staircase defect . . . 37
3.2.3 Overhangs and islands . . . 38
3.2.4 Delamination . . . 41
3.2.5 Shrinkage and warping . . . 41
3.2.6 Seams . . . 42
3.2.7 Printing time . . . 42

4 Infills **45**
4.1 Dense infill . . . 45
4.2 Sparse infill . . . 47

4.2.1 2.5D infills . . . 48
4.2.2 3D infills . . . 48

5 Out-of-plane printing **55**
5.1 Machines and modifications . . . 55
5.2 Conventionnal printers . . . 58
5.2.1 Smoothing the surface . . . 58
5.2.2 Printing lattice structures . . . 58
5.2.3 Following a field . . . 59
5.2.4 Spiralize, helical or continuous . . . 59
5.2.5 Artistic work . . . 60
5.3 Experimental approaches and partially curved layers . . . 61
5.3.1 Partially curved layers . . . 61
5.4 Fully curved algorithms . . . 61
5.4.1 Curved printing pipeline . . . 62
5.4.2 Shell-like prints . . . 62
5.4.3 Supportless printing . . . 63
5.4.4 Wiremeshes . . . 65

" *The thing about quotes on the internet is that you can not confirm their validity.* "

Abraham Lincoln

Introduction

Additive Manufacturing has a vast field of research, which is impossible to cover in a single thesis. There are various technologies with mixed processes. Some processes are available only for some technologies, and sometimes relying on the base material used. This state of the art will first cover Additive Manufacturing and the different technologies available on the market. What is interesting in this part is to have a good overview of what is done, which facilitates identifying with which technologies our research can be used. Then we will outline the process from a 3D model to a physical object to better visualize the different steps of modeling, slicing, or printing. The third part is dedicated to 3D printing by filament deposition. This thesis uses this type of printer, so it is worth knowing the techniques and possible issues related to this technology. Then in the fourth part comes the part detailing filling patterns. We will also find a whole chapter dedicated to it, with the presentation of two filling techniques we have developed. And finally, the last part and the main subject of this thesis, out-of-plane printing. We find a mixture of hardware and various techniques that have emphasized the challenges and intricacies of curved printing.

Chapter 1

Additive manufacturing technologies

Additive Manufacturing (AM) is a process for creating 3D objects by adding or solidifying material. During the process, layers of material are stacked on top of each other, creating, layer by layer, the desired object (Figure 6). Additive Manufacturing can make use of many technologies with different tradeoffs. Each AM technology has constraints related to the printing technology, the material used, and the expected functions (aesthetic, mechanical, functional). A wide range of fields makes use of 3D printing for part manufacturing: aerospace, aeronautics, architecture, art, automotive industry, dentistry, fashion, food, jewelry, medicine, pharmacy, robotics, and the toy industry [Brischetto et al., 2016; Rebong et al., 2018; Chen et al., 2016; Moraru et al., 2018; Hutmacher et al., 2001].

From a software point of view, the additive manufacturing processes can be split into broad categories: vector-based and raster-based. The vector-based techniques generally extrude or solidify raw material by moving a tool following continuous paths. In contrast, the raster-based methods generally use images (pixels or voxels), allowing the solidification of an entire layer at once. In the following section, I will present the different categories of technologies and specify whether they are raster-based or vector-based, as all the work in this thesis mainly focuses on vector-based technologies.

Figure 6: Yoda puzzle, where each layer has to be put on a specific order to solve it. (Educa-16501)

1.1 Powder Bed Fusion

Powder Bed Fusion (PBF) [Sun et al., 2017] (Figure 7) is a vector-based technology used in a wide range of AM processes, including Direct Metal Laser Sintering (DMLS) [Childs et al., 2005], Selective Laser Sintering (SLS) [Deckard, 1988; Williams and Deckard, 1998; Kruth et al., 2005], Selective Heat Sintering (SHS), Electron Beam Melting (EBM) [Murr et al., 2012] and Direct Metal Laser Melting (DMLM). These systems use lasers, electron beams, or thermal print heads to melt or bond layers of material in a three-dimensional space. The materials commonly used are powders of steel, stainless steel, chrome cobalt, titanium, and aluminum. Today, manufacturers offer machines with powerful laser technology capable of printing from 10 µm thick layers or less. The manufacturing process begins by spreading a thin layer of powder. A beam then fuses the powder at the locations defined by the computer. When the layer is done, the bed is lowered and another layer of powder is applied. The process is repeated until the object is entirely done. At the end of the process, excess material (unused powder) needs to be removed. One of the significant drawbacks of these techniques is that material might end up being imprisoned inside the part if it contains closed pockets.

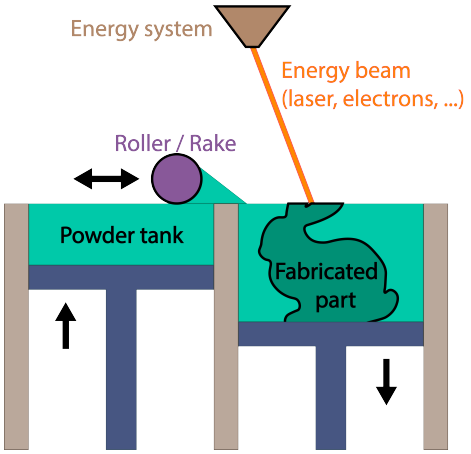


Figure 7: Power bed fusion technology.

1.2 Material Jetting

Material jetting (Figure 8) is a raster-based technology. Much like the head of a 2D inkjet printer, it adds material by moving back and forth the print head [Gao and Sonin, 1994]. However, unlike a standard printer, it also moves in a third axis to print 3D objects[Yeong et al., 2005]. A print head deposits a droplet of material; then, it hardens as it cools or is cured by ultraviolet light, creating a layer from the object. One significant advantage of this technology is the ability to print high-quality colored objects and multi-material objects[Yang et al., 2017].

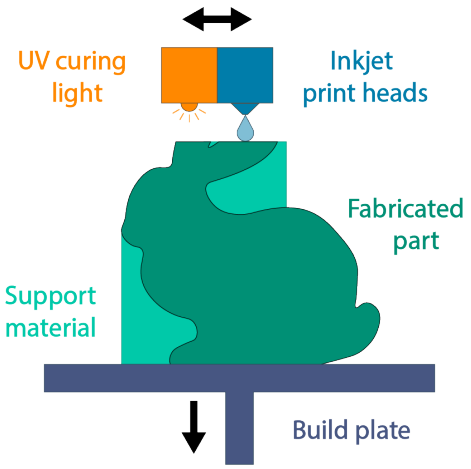


Figure 8: Material jetting technology.

1.3 Binder Jetting

The binder jetting process (Figure 9) is quite similar to material jetting [Ziaee and Crane, 2019]. An automated roller distributes a thin layer of powder (polymers, plaster, or metal) on the manufacturing plate. Then the print head applies a liquid binder to the powder, creating a layer. The printing platform lowers itself slightly to allow the addition of a new layer of powder. The process is repeated until the object is created. As with SLS, binder jet printing does not require support as the unfused powder itself takes care of this. However, this technology requires post-processing steps. Excess powder must be removed by vacuum, and the object must be dedusted with compressed air. Depending on the materials used, the printed part must be placed in an oven for baking or sintered.

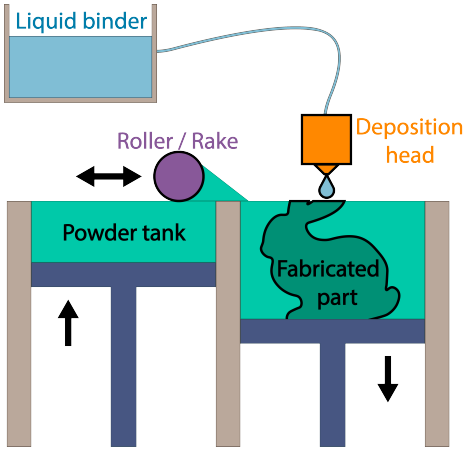


Figure 9: Binder jetting technology.

1.4 Material Extrusion

Material extrusion (Figure 10), a vector-based technology, is one of the most well-known and most widespread additive manufacturing processes. It focuses mainly on extruding a thermoplastic filament following deposition paths, but other materials like ceramics, concrete, or silicons can also be used. Fused Deposition Modeling (FDM)[Crump, 1991] is a trademark registered by Stratasys Inc in the late 1980s (patents expired in 2009). The equivalent term is Fused Filament Fabrication (FFF).

The material is extruded through a nozzle to print a layer by filling the section of the object to be fabricated; then, the nozzle moves vertically to repeat the process to the next layer. The most common materials used in FDM are ABS [Ahn et al., 2002], PLA [Drummer et al., 2012], PolyCarbonate (PC), and polypropylene [Carneiro et al., 2015]. However, it is possible to find filaments filled with wood particles, short fibers, and other polymers like PEI [Schöppner and KTP, 2011] or PEEK [Wu et al., 2015] with high mechanical, chemical, thermal, and medical-grade properties. FDM is the most widespread and accessible 3D printing technology on the market. It is the main focus of this thesis.

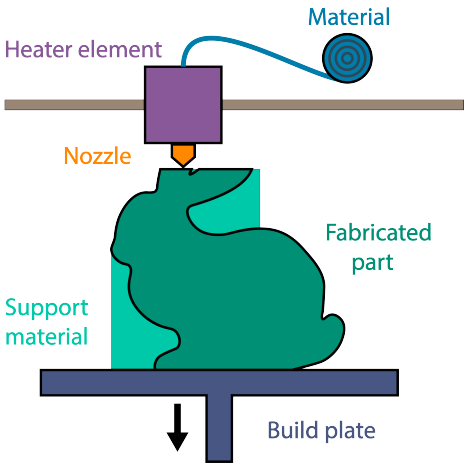


Figure 10: Material extrusion technology.

1.5 Directed Energy Deposition

The process of Directed Energy Deposition (DED) [Gibson et al., 2015] is similar to material extrusion [Mazumder et al., 1997]. An electron beam gun or laser is mounted on a four or five-axis arm [Milewski et al., 1998] instead of a standard nozzle, e.g., Construction Laser Additive Directe (CLAD) nozzle. It can be used with various materials: polymers, ceramics, metals fed as either wire, filament feedstock, or powder. It is pretty similar to material extrusion with the same advantages and drawbacks except for the price.

1.6 Sheet Lamination

The main sheet lamination methods are Laminated Object Manufacturing (LOM) [Mueller and Kochan, 1999] and Ultrasonic Additive Manufacturing (UAM) [White, 2003]. LOMalternates layers of paper and adhesive, while UAM weld thin metal sheets through ultrasonic welding, including titanium, stainless steel, and aluminum. Laminated Object Manufacturing excels at creating aesthetic objects. For instance, Felix Semper, an artist, build unique moving sculptures only using paper and glue.

1.7 Vat Polymerization

Vat polymerization uses a liquid photopolymer resin. This resin will harden when exposed to UltraViolet (UV) light. The object is generally printed downward as the resin stays in the vat during the process (Figure 11).

Light curing in a resin bath using laser scanning, such as stereolithography (SLA), is a vector-based technology. It was invented in 1984 in the USA and France simultaneously. In July 1984, Alain Le Mehaute, Olivier de Witte, and Jean-Claude André filed a patent describing the first successful stereolithography process André et al. [1984]. It was accepted in January 1986; however, this application was abandoned. In August 1984, three weeks after the French patent was filed, another patent based on a stereolithography process was filed by Chuck Hull(Charles W. Hull) [Hull, 1986] and was accepted in March 1986. While Laser 3D failed to market the technique as a service, not as a machine, Chuck Hull founded 3D Systems, one of the most important manufacturers of Additive Manufacturing systems.

Digital Light Processing (DLP) [Hornbeck, 1996] is a raster-based technology that allows building an object much more quickly by decreasing the time of manufacture by replacing laser scanning with image projection. This image projection solidifies a whole layer at once, compared to SLA, where the laser needs to scan the surface.

Unlike stereolithography, DLP, and most other processes, Continuous Liquid Interface Production (CLIP) of 3D objects [Tumbleston et al., 2015] is continuous and allows faster prints. The Continuous Liquid Interface Production is achieved with an oxygen-permeable window below the UltraViolet image projection plane, which creates a "dead zone" (persistent liquid interface) and removes the need of lifting the object above the window to let the resin levels stabilize.

Computed Axial Lithography is a recent method by [Kelly et al., 2019]. Much like

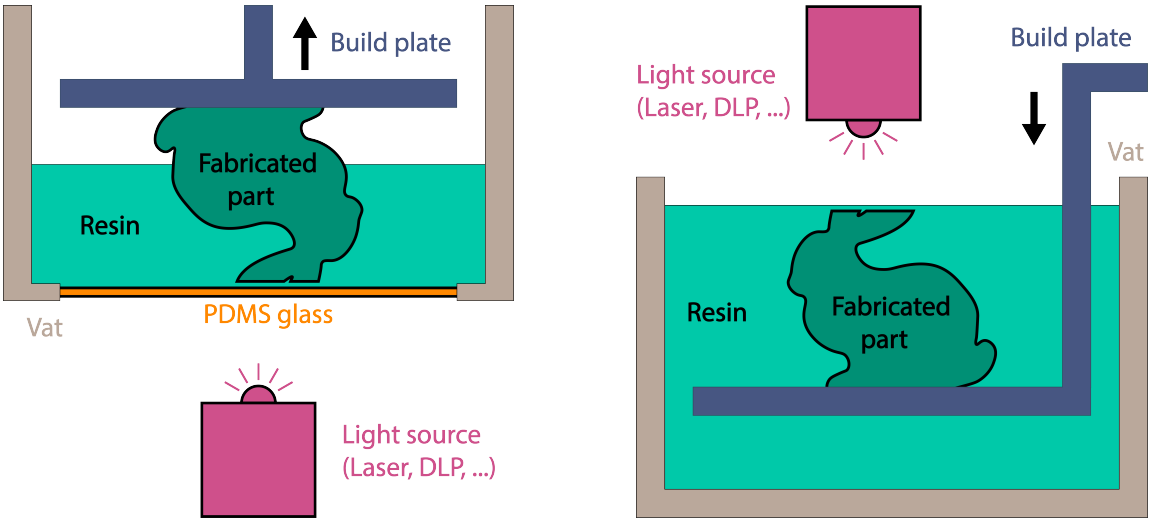


Figure 11: Main VAT polymerization technologies. On the left, the bed rises out of the VAT tank while on the right, it deepens inside. On the right Vat technology, the printing height is bounded by the VAT height.

CLIP, an image is projected. However, the whole tank receives light instead of projecting on a thin layer. This process is inspired by Computed Tomography (CT), which is used in radiotherapy to cure an area. Instead of accumulating radiations on precise areas, it accumulates a sufficient dose of light, hardening the resin.

In this thesis, I will only focus on vector-based technologies, and all my experiments will be run with material extrusion technologies.

	Material					Categories		Support and overhangs	Problematic closed cells
	powder	liquid, resine	pellet	filament	sheet	vector	raster		
Powder bed fusion	plastic, metal					X			unfused powder
Material jetting		plastic, wax, metal					X	always required (dense structure)	
Binder jetting	gypsum sand	(binder)					X		unfused powder
Material extrusion		clay, silicone	plastic, composite	plastic, composite		X		45 deg (any structure)	
Directed energy deposition	metal		plastic	plastic, metal		X		45 deg (any structure)	
Sheet lamination		(binder)			paper, composite		X		
Vat polymerization		plastic, silicone				SLA	DLP CLIP	always required (any structure)	not empty closed-cells if the build plate goes down

Table 1.1: Comparison of the different AM technologies.

Chapter 2

Slicing pipeline

For each technology, a dedicated software has to generate the set of layers, either as paths or images, representing the different steps to print the 3D object. In all cases, before any operation, a 3D object needs to be created. When modeled, it is transformed into machining instructions by a dedicated software called slicer, and then be printed. Livesu et al. [2017] did a complete review detailing every aspect of this slicing pipeline. The objective of this pipeline is to step by step print an object based on its original design (Figure 12). Starting from a 3D model, we want to generate machine instructions to create the desired object. The first step is to create a 3D representation of the object inside a Computer-Aided Design (CAD) software. Once modelization is complete, we use another software, called Computer-Aided Manufacturing (CAM), that takes a 3D model to generate machining instructions. The targeted hardware will then execute these machine instructions to manufacture tangible objects.

This section will follow the slicing pipeline with first an overview of Computer-Aided Design and then an overview of Computer-Aided Manufacturing. The last part of the slicing pipeline, the effective printing, will not be seen, as material deposition or solidification highly depends on the technology used.

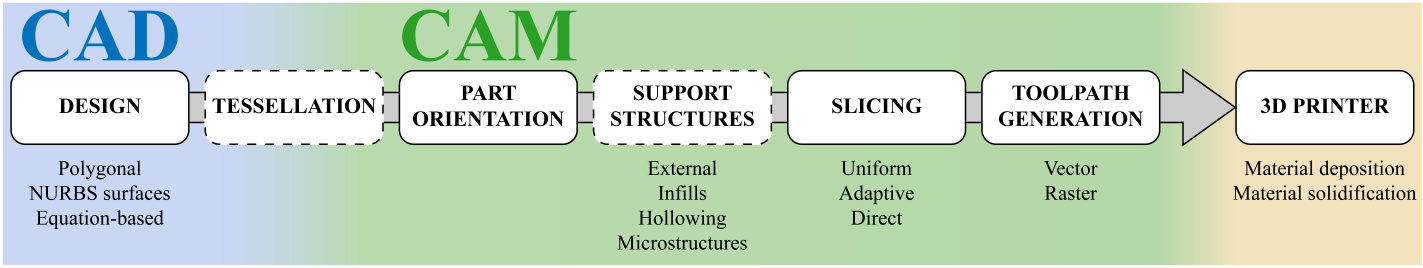


Figure 12: Slicing pipeline, from conception to instruction generation and then reification.

2.1 Computer-Aided Design

Computer-Aided Design (CAD) is an umbrella term for all software and geometric modeling techniques used to design, optimize and simulate parts on a computer. The design process can be done in different ways depending on the desired objectives. For engineering parts, the designer defines quotes like extrusion lengths, hole diameters, bevels, and all kind of other shapes and dimensions. However, in artistic models, the designer generally proceeds with more manual techniques. The manual modeling process can be done similarly to plastic arts, such as sculpting (creasing and carving the surface), creating, moving, and linking points to make edges and surfaces.

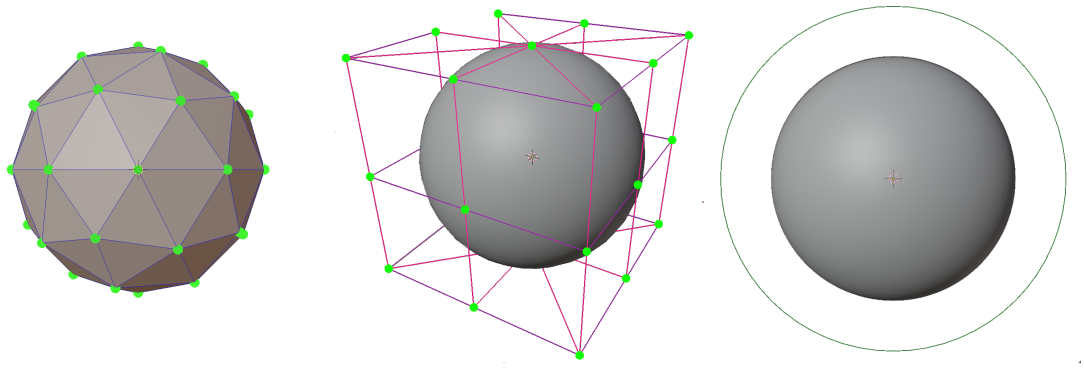
2.1.1 Modeling and shape representation

Modelization with points in space connected through edges and faces is called polygonal modeling (Figure 13a). It is an approach for modeling objects by representing or approximating their surfaces using polygon meshes. A polygon mesh is an object represented by its surfaces, and each surface is composed of three or more vertices. With polygonal meshes, all the points and surfaces must be stored explicitly.

A different way to model 3D objects is through mathematical representation. We can use mathematical models instead of directly representing the surface with vertices. The commonly used representation uses Non-Uniform Rational B-Spline (NURBS), see Figure 13b, where some control points define the surface. This method gives high-quality surfaces with better reproduction of curves and lowers storage costs than polygonal meshes.

Another way to represent surfaces is to use equation-based modeling otherwise known as implicit surfaces or functional representation (Figure 13c). They can be composed with boolean operations to generate more complex shapes. This combination of shapes is called Constructive Solid Geometry (CSG) Figure 14.

Constructive solid geometry allows the creation of complex surfaces or objects using Boolean operators, such as Union, Intersection, and Difference. While often used with implicit surfaces, it can also be performed on polygonal meshes.



(a) (b) (c)

Figure 13: Multiple shape representations of a sphere. a) Polygonal modeling with vertices in green and edges in blue. b) NURBS representation with control points in green. c) Implicit representation, surface defined with: $x^2 + y^2 + z^2 - 4 = 0$.

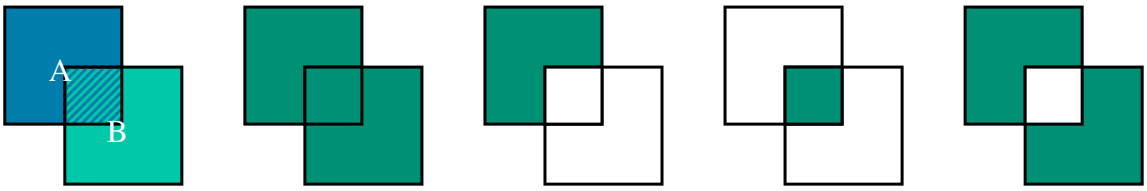


Figure 14: CSG Operations (A operator B). From left to right: Union, Difference, Intersection, XOR.

2.1.2 Tesselation

The chosen standard representation of a 3D model in the additive manufacturing industry is a tessellation with triangular surfaces to ensure interoperability between different CAD and CAM packages. In this representation, non-polygonal meshes have to remesh the surface with triangles, introducing a surface approximation.

3D Systems introduced a file format dedicated to stereolithography CAD software, the STL file format. It is now widely used for rapid prototyping, 3D printing, and computer-aided manufacturing. Compared to other 3D object file formats, STL files describe only the surface geometry of an object without any representation of color, texture, or any other attributes.

2.1.3 Polygon mesh requirements

In order to be usable with CAM softwares, the polygon meshes must enforce some constraints. The mesh cannot contains degenerated triangles (2 or 3 vertex on the same point in space), facets should not overlap and the mesh must be watertight. Watertightness usually describe manifold meshes consisting of one closed surface that do not contain holes. Please, refer to Attene et al. [2013] for a complete survey of repairing methods.

2.2 Computer-Aided *Manufacturing*

Moving from CAD to CAM is equivalent to switching from design to fabrication. It is necessary to use tessellated objects to interoperate from software most of the time. The Autodesk tool suite and IceSL did not have this requirement, as both are complete CAD and CAM packages. The former shares information between its modules through dedicated file formats, while the latter embeds CAD and CAM in the same interface. Once the CAD file is created, it must be transformed into machine instructions in the CAM phase. This part is done in two steps for layered additive manufacturing: slicing and toolpath generation. The first step is to slice (cut into layers) the 3D model and ensure the printability of the part. In the second step, the toolpath generation must convert these slices into movement sequences the tool must execute. The whole process is generally called slicing, too. This homonymy can lead to a non-harmful misinterpretation as the slicing step of the process is the central part.

2.2.1 Slicing step

The slicing process requires additional structures and data to print successfully. Due to the layer-by-layer nature of the slicing and the constant build direction, the part orientation define which part will or will not be hoverhanging. After choosing the orientation, as some parts may not be supported with previously deposited material, the slicer needs to add support structures. Depending on the technology used, additional structures can target other properties than fabricability constraints, like inner structures also called infills that impact the physical properties. The designer may use those other structures for more particular properties, such as a lightweight object, a highly resistant print, or a fast print for a prototype. In order to be compliant with the designer's requirements, the slicer will have to generate trajectories for the outer shell and the inner parts.
For raster-based technologies, such as DLP, the slicer has to generate a set of images corresponding to each layer, and in vector-based slicing, a set of paths corresponding to all the moves the printhead will do.
In FDM, a vector-based technology, material deposition along the toolpath poses some challenges. The path has to be designed so that material deposition is as consistent as possible, avoiding under and overflow during deposition. The same problem arises when printing paths are close to each other, where the overlap between deposition paths must be carefully controlled.

Continuity

Keeping the amount of deposited material constant along the path is crucial in FDM and other material deposition processes. Since it is a viscous fluid, any requested flow variation will not be instantaneous. Based on this, the deposited material at the path endpoints is challenging to control. Printer firmwares optimizes the acceleration of the motors [Klipper3D, 2022] and can be tweaked to achieve the best results. Recent works have focused on overcoming this problem by reducing the number of paths needed to cover the slice area as much as possible to reduce discontinuities in the deposition process [Zhao, Gu, Huang, Garcia, Chen, Tu, Benes, Zhang, Cohen-Or and Chen, 2016; Jin, He, Fu, Zhang and Du, 2017; Dehaeck et al., 2018; Papacharalampopoulos et al., 2018; Bedel et al., 2022].

Geometry

We can extend the discontinuity problem to the path's geometry, where low curvature paths are preferred to sharp turns [Jin et al., 2014; Comminal et al., 2019]. As every change in speed implies a change in flow, every sharp turn may lead to under or overextrusion of material.

To better see the effect, let's consider the nozzle depositing a circle of material. The deposition paths can be decomposed into a sum of simple paths, and an extrusion of a circle along this paths as the shape of a capsule. When summing this capsules, we can easily see the areas where overfilling and gaps will appear (see Figure 15).

Of course, in practice, the defect will not be such a disaster. The printer firmware is aware of this issue and will reduce it. In addition, the material is not solidified at this stage, so it will spread a little, partially filling the gaps and reducing the overflow. Some infills also compensate for this problem, see for example variable width contouring section 4.1.

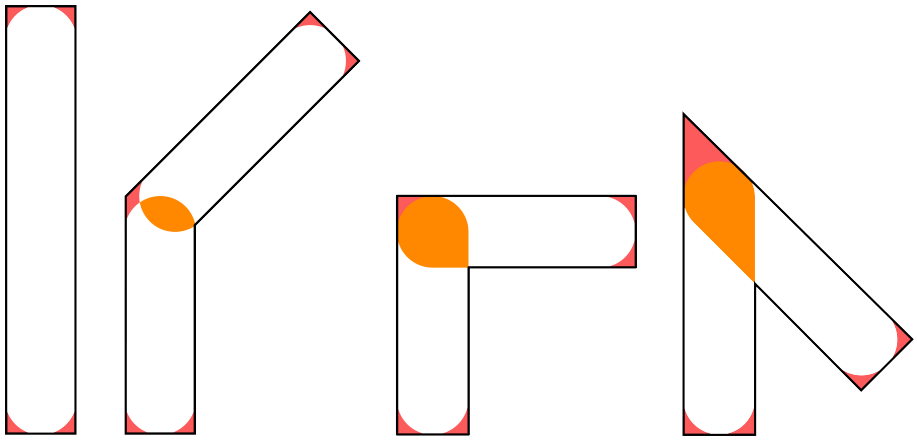


Figure 15: Sharp turns, leading to over-extrusion and gaps. in orange: over-extrusion, in red: under-extrusion or gaps.

Performances

The printed objects' size ranges from a few centimeters to tens of meters, with a resolution starting from the tenth of millimeters to centimeters. In order to process arbitrarily complex slices, the easiest way to reduce the size of the problem is to split it into smaller and more manageable chunks. The more evident chunks are slices themselves. When it's not sufficient, approaches such as a partitioning strategy can be used; The slice is split into smaller areas rather than covering the whole area at once, and the results are merged. The easiest way to do this is to split the domain into homogenous square tiles .

Besides the method used to decompose the domain and the particular paths used to fill each subproblem, another essential factor in the definition of a good toolpath is the reduction of the travel time. The travel time is the time necessary to move the nozzle from the end of a path to the beginning of the subsequent one. Disconnected toolpaths distances can be minimized to reduce the travel times by optimizing the path process order (the nozzle travel path from the end of a toolpath to the beginning of the next one), and the direction of each printing path (from start to end, or the other way round).

The path planning optimization problem can be reduced to the NP-complete Travelling Salesman Problem (TSP). Many methods have been proposed to find an approximate solution, using greedy algorithms, genetic algorithms [Wah et al., 2002; Gong and Zhang, 2015], the Christofides algorithm for TSP [Christofides, 1976; An et al., 2015; Fok et al., 2016; Ganganath et al., 2016] or other heuristics [Castelino et al., 2003; Ghaiebi and Solimanpur, 2007; Xiaomao et al., 2011]. Some better solutions are given for more specific ordering problems, such as contour parallel infill [Xiong et al., 2019], lattice structures using the solution of the chinese postman problem [Dreifus et al., 2017] or non-connex parts [Jiang, 2020].

Patterns

In addition to the computational time, internal volumes of parts are a significant factor in time and material consumption. Several infill patterns have been proposed to save time, reduce material consumption or improve mechanical properties. See Chapter 4, "Infills" [p.45] in related work for an in-depth discussion of infill patterns.

2.2.2 Toolpath generation

Once the slicing is done, all the paths generated are exported under a format that the machine can process. For raster-based technologies, it is a set of images, with an additional file describing the printing size, layer height, and other pieces of information. For vector-based technologies, the CNC machining instructions' set (already able to follow paths) is extended with additional instructions to manipulate the printer.

Raster-based toolpath. Most raster-based technologies use boustrophedon-like movements [Golomb and Selfridge, 1970] to cover all the surface of the image, which means that the tool will go back and forth like a paper printer. The most affordable technology, DLP, does not use any toolpaths; it directly projects the image onto a screen to light up the resin, solidifying it. The techniques directly using images are faster to print objects but at the cost of a loss in precision as the minimum feature size will depend on the screen resolution.

Vector-based toolpath. In vector-based technologies, the tool is moved "freely" in space. The toolpaths generated during slicing are writen in a specific file format: GCode, which is an extended version of the CNC's file format that adds temperature, extrusion speed, extrusion volume and many other instructions specific to the type of printers.

Chapter 3

Fused Filament Fabrication

Fused Deposition Modeling (or Fused Filament Fabrication) is the most widespread additive manufacturing technology and the least expensive one. While generally using 3-axis motions, laboratories and industries often use four and more-axis machines to improve characteristics such as stiffness or visual quality. Even though this technology is extensively used nowadays and thus feels mature, it still suffers from many constraints and defects actively being researched. That is why this thesis mainly focuses on this technology. The first section will extensively describe the printing process and the printers themselves. The second one will explore the common defects and how to manage or mitigate them.

3.1 Fused Deposition Modeling printers

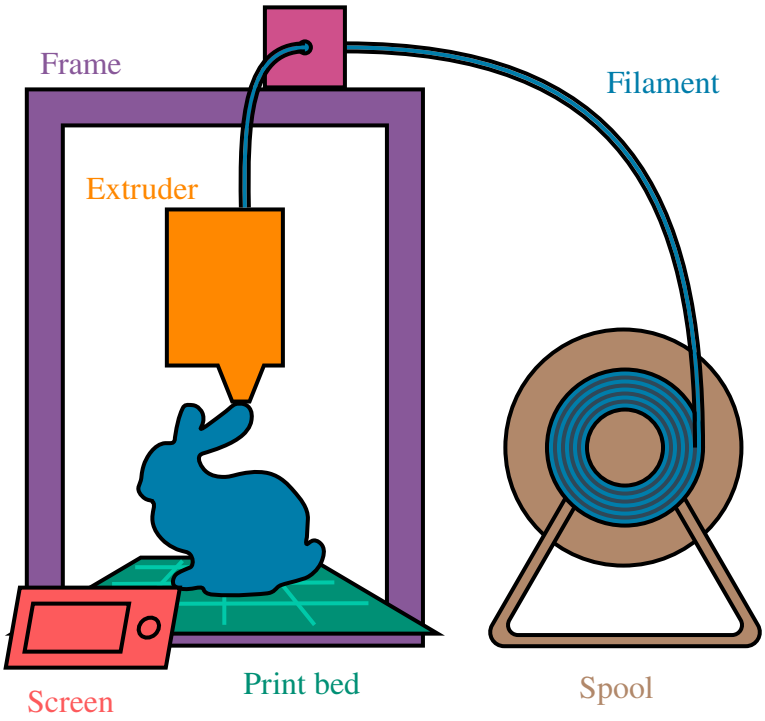


Figure 16: Common parts of an FDM printer

3.1.1 Anatomy of a 3D printer

The following paragraphs give a quick guide on the anatomy of the most common desktop 3D printer, focusing on mechanical and electrical components.

The frame

The frame is the structure keeping all the printer's parts together. Its shape depends mainly on which part of the printer each motor will drive (subsection 3.1.2, "Standard 3-axis machines" [p.31]).

The build plateform

The build platform, commonly named the print bed, is the surface over which the part will be printed.

Most build platforms are heated, below the glass transition temperature of the printed material, to reduce the stress induced by the material cooling. By limiting the cooling stresses, the part adhesion is much better, and waiting for a complete cool down of the build plate allows easy removal. Another way to reduce cooling stresses is using an enclosure to keep a reasonable temperature and avoid drafts. There are many different types of bed surfaces. Most printers will come with some sort of all-purpose surface, like BuildTak or PEI sheet. However, depending on the material printed, different surfaces will have different results.

Bed Leveling From the beginning, bed leveling has been a part of 3D printing that can make or break a user's experience. Additional structures, such as *rafts*, can be used to mitigate the risk of a failed print. A *raft* is a sacrificial structure that compensates for an unleveled bed to obtain a more flat printing surface. Initially, rafts were the best way to ensure that your 3D printer had a level surface to work on but are used much less today because of the variety of ways to level a bed reliably.

There are three main categories when it comes to bed leveling: software leveling, hardware leveling, and manual leveling. Most printers have some form of manual leveling, meaning that the height of the printer bed can be adjusted, usually with screws at three or four points on the bed. However, three screws make it easier than four; after all, three points form a perfect plane. What can happen with four screws is that you can have three screws in one plane and the fourth screw slightly offset so that it bends the printing surface along the diagonal. It is possible to level a bed with four screws, but it is a little more complex than with only three screws. Nevertheless, four contact points ensure better stability with oversized or heavy beds. Sometimes, the printer's bed leveling cannot be adjusted manually. In such conditions, the manufacturer generally adds a way to specify the suitable height of the bed on multiple points of it. Then, at the execution time, it will consider the modification done by the user. While the previous methods allow to truly level the bed or compensate for the deformations of the bed, there are also automatic methods. The machine can check the bed's position using probes to estimate its deformation and compensate for lousy leveling and distortion. This way, the print quality is kept constant without needing to fine-tune by hand.

The material extruder

The extruders are available in two formats, direct extrusion and remote extrusion. Direct extrusion (or direct-drive) offers the closest extrusion design to the hotend, and remote extrusion allows placement of the extrusion's motor outside the carriage.

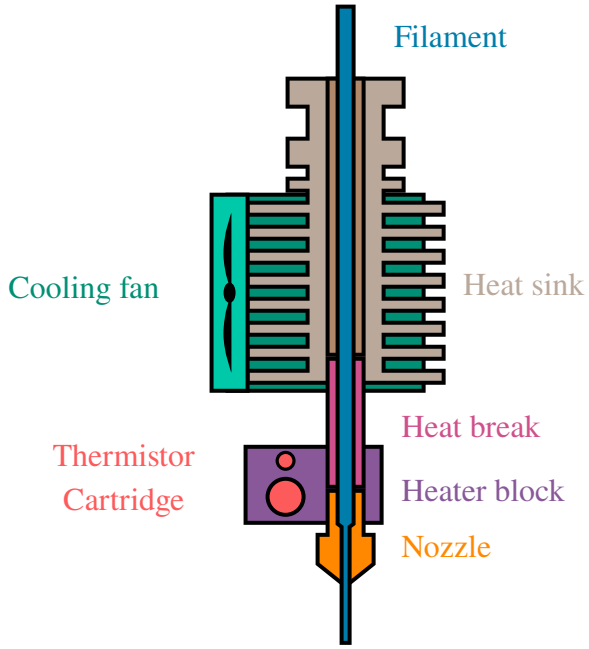


Figure 17: Anatomy of the material extruder

The direct extrusion method provides a good quality of filament drive as it limits friction to its minimum. As the motor is embedded in the print head, as close as possible to the heating block, it naturally reduces extrusion problems, like the filament's shrinkage. The direct drive's real advantage is its excellent compatibility with a wide variety of 3D filaments. Indeed, the direct-drive technology is adapted to print flexible filaments and more abrasive materials. However, more powerful motors and axes can be used to compensate for this additional weight to limit the vibrations.

Remote extrusion is achieved via a bowden tube, hence bowden extrusion. Unlike direct drive extrusion, the extrusion motor is fixed to the 3D printer chassis, and a PTFE tube (bowden tube) connects it to the heating block. The extrusion motor then pushes the filament into the bowden tube directly connected to the top of the heating block. The advantage of this technology is the weight loss of the print head, which makes the movements smoother, quieter, and, therefore, more precise. Nowadays, the extruders and the motor's drivers compensate for the problems which affected the bowden tube a few years ago. Nevertheless, offset extrusion limits the printing of technical materials. Flexible materials tend to move inside the bowden tube, leading to a precision loss. Those too abrasive may quickly deteriorate the bowden or catch on it, amplifying the risk of printing errors.

	direct extrusion	**remote extrusion**
extruder weight	extra vibrations	
printing speed	inertia	
material retraction		precision
standard material		
soft material		precision
flexible material	risk of failure	precision / failure
technical material		deterioration
maintenance	all in one block	

Table 3.1: Comparison between direct and remote extrusion.

Cooling system

In FDM printing, two critical areas may require to be cooled down.

The first one is the extruder; otherwise, the heating block will heat the whole extruder. The heat must stay on the hotend to fuse the plastic only on this part. This heat transmission is blocked by connecting the hotend and the coldend with a smaller piece called a heatbreak, which reduces the heat transmission. Additional cooling is generally achieved via a fan blowing through a heatsink to dissipate the heat.

The second area is the material itself, which has to cool down to be solid again to build on top of the previously deposited layer. In order to make the material cool fast enough, a cooling system (generally mounted on the extruder) is added to the printer. Most of the time, an air cooling system is used, but for specific cases where the material has to cool down almost instantly, other fluids can be sprayed directly on the newly deposited material.

Motherboard

The motherboard (or controller board) is the brain of the printer. It takes the commands by reading external storage or directly by a computer and orchestrates their execution. The motherboard contains a microcontroller and all the circuitry needed to run the motors, read the different sensors, and talk to a computer.

Sometimes, a Raspberry Pi is added as a middleware between the printer and the computer, offering an enriched interface to control the print process and better display the current state of the machine and the print.

Stepper drivers

These chips are responsible for the operation of stepper motors. They automatically activate the motor coils, causing them to move in increments. The stepper driver also controls the amount of electrical current that powers the motor. The higher the power, the stronger the motors will be, but at the expense of heat (which shortens the lifespan). By balancing the power delivered to each coil, the driver can divide the steps into additional micro-stepping increments allowing for more precise motor control than is usually possible.

3.1.2 Standard 3-axis machines

3-axis machines come in various shapes; some take less space relative to their build space, and others can achieve higher precisions or speeds. In Figure 18, all printers have approximately the same build space, and we can see that the delta printer is much bigger than the other two. Nevertheless, that does not preclude delta printers from being used more than polar ones.

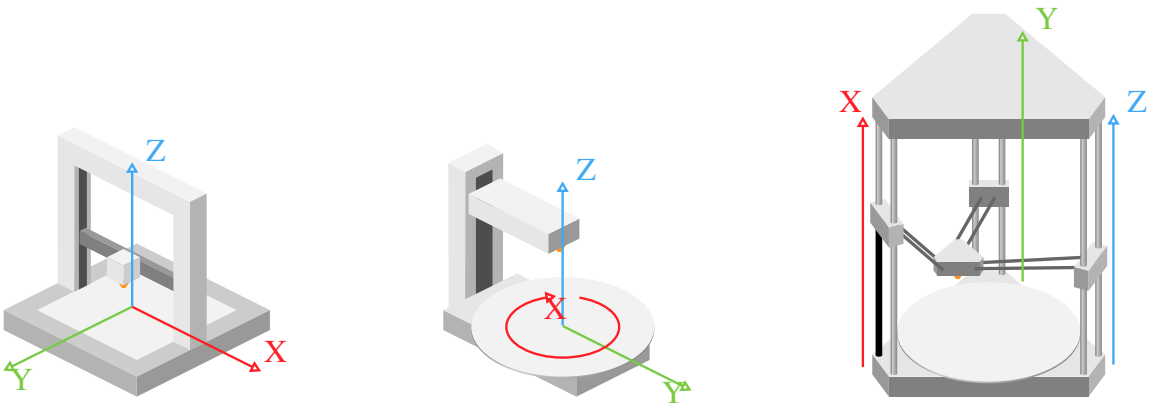


Figure 18: From left to right: Cartesian printer, Polar printer, and Delta printer. The motor axes are represented with X, Y, and Z arrows unrelated to the GCode coordinates.

Cartesian printers They are named so because of the Cartesian coordinate system they use. This coordinate system consists of three orthogonal axes - the X, Y, and Z - used to determine the printhead's position regarding the printed part. This type of printer is the most common on the market with some of the most well-known brands, such as Makerbot (with the first model called Replicator based on the open design RepRap project), Prusa, and Ultimaker. The main advantage of these solutions is that they are usually relatively inexpensive and are even sold as kits.

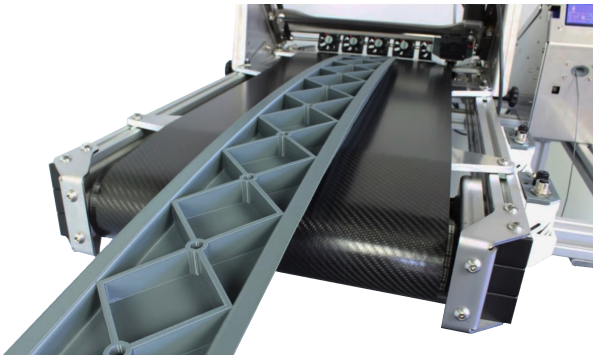
The print bed and printheads can move in specific ways:

- The bed moves horizontally along one axis (X or Y), and the extruder moves vertically and along the remaining axis. It is referred to as Prusa style since they are the most well-known manufacturer that democratized this type of platform.
- The bed moves vertically (Z), and the extruder moves in two dimensions along the horizontal (X and Y) axes. This type of printer exists in different fashions such as crossed-gantry, CoreXY and H-Bot. Ultimaker and Makerbot use them.
- The bed moves vertically and horizontally along one of the two axes while the extruder moves on the second one.
- The bed moves along the XY axes, and the extruder moves vertically.
- The bed does not move, and the extruder moves in three dimensions.

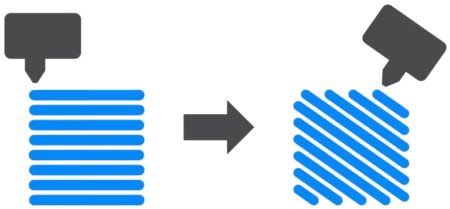
(a) Prusa MK2s, a i3 style printer *(b) Ultimaker3, a crossed-gantry style printer*

Figure 19: Different cartesian machines, on the left a Prusa printer and on the right an Ultimaker one. An evident difference between those two systems is the direct and remote extrusion. However, another difference lies is the Z axis, holding the print bed for the Ultimaker and holding the X-axis in addition to the extrusion motor for the Prusa.

Quasi-infinite bed size can be achieved in one direction using a treadmill or a conveyor belt as a one-axis print bed Figure 20. This infinite bed can be used to print large print or (sort of) mass producing by continuously dropping finished objects into a container. To print with a conveyor belt, the nozzle must be tilted, which affects the overhang angle (see section 3.2.3), as the detection angle will also be tilted. A distinct version of Cura dedicated to conveyor belt printers, called Blackbelt Cura, was created. To be used with any slicer, a post-process script was written by *wjsteele* and is available on Thingiverse (Tilted Conveyor FDM Printer Post Processor, thing:2358314).



(a) Conveyor belt printer from ©Blackbelt 3D



(b) ***Left:*** *Standard FDM extrusion* ***Right:*** *Belt 3D tilted extrusion*

Figure 20: A conveyor belt printer with tilted nozzle.

Polar printers The difference with Cartesian machines is that they use polar coordinates for 3D printing. This coordinate system describes points on a circular grid instead of a square, replacing X, Y, and Z axes by angle and length. This technical choice means that the bed rotates and moves simultaneously. These printers are ideal for cylindrical objects because of the rotating print surface, allowing for a constant speed over the motor. Another advantage of polar solutions is that the printer can easily have a larger print volume, as the tool head only has to move half the bed size horizontally, and the dimension of the Z-axis defines the height of the print. However, the main problem that plagues polar machines is their precision. Movement using polar coordinates is intrinsically more precise near the center than around the edges as they are defined with angles and rotated in a circle.

Delta Delta printers are named so because they move using a triangular system of coordinates. The delta printer's extruder head is suspended mid-air by three arms separated at equal angles. The suspended head imposes a lightweight extruder. These arms move up and down to control the extruder's position by moving the print head closer or further away. One of the main advantages of this technology is that on a single slice, all arms work together to achieve a single movement. This configuration offers many advantages as multiple motors work for every movement and the lighter head reduces inertia:

- better accelerations and speeds
- quasi-instant changes of direction are possible
- higher print speed and accuracy

All those advantages come from the lower load on each motor.

3.1.3 Multi-material and multi-color printing

Extrusion printers can print several materials and colors by changing the filament. Using multiple filaments on a print makes it possible to achieve multi-color printing. The hobbyist community has a strong interest in filament color printing with several successfully funded community projects [Sammut, 2016; Labelle, 2016; Chan, 2017] or tools [LithophaneMaker, 2022]. One of the main interests is that the printed part does not need to be painted afterward, reducing the post-processing time and complexity. Color prints can also be used for visualization, for example, to see the stresses on a part with color gradients (Figure 21). The main idea of color printing with filament is to use filaments of different colors to trick the eye. This can be achieved in several ways, depending on the printer's hardware.

Single-extrusion heads A first approach is to switch filaments during printing. It can be done manually but requires the user to swap filament regularly. The Palette project [Mosaic, 2021] automates this process with an additional device that cuts and fuses multiple filaments into a single, continuous, multi-color filament. Another automated approach is to rely on a switching extruder [Prusa, 2016; Sammut, 2016], also called Multi-Material Unit (MMU), where a mechanical system allows for quick selection between different filaments. When a material is pushed into the melting chamber, it mixes with the previous one. During this transition, the output material takes an intermediate and unstable color. The typical way to eliminate the mixed material is to use a *purge tower*, a "bin" outside the print bed. The *purge tower* is a tower of waste material used to clean the melting chamber and is generally printed alongside the print and disposed of afterward. Furthermore, recently, a hotend swapper [MateRiel, 2022] that can be beneficial with Palette or MMU changes the hotend automatically at every color change.

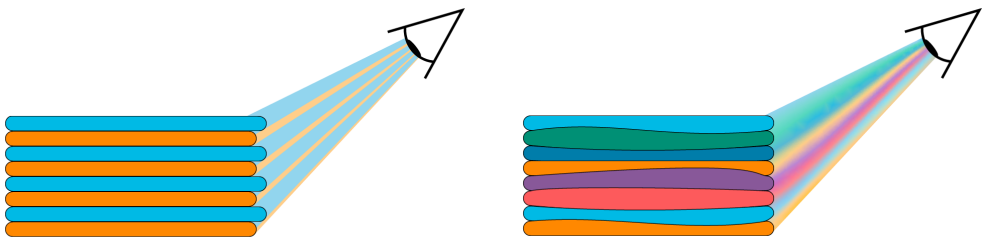
(a) Colored print *(b) Colored print to visualize stress distribution*

Figure 21: Different color prints [Song et al., 2019].

Multi-extrusion heads Another approach is using a printer with multiple hotends, typically mounted side-by-side on the same carriage. Multi-extrusion systems allow using eclectic filaments to print with different colors or materials. However, multiple hotends printers tend to have material oozing, but this issue can be tackled through path planning and part orientation [Hergel and Lefebvre, 2014].

Combining two extruders into a single printhead may initially sound inefficient, but it is the most popular and low-cost method. This dual nozzle system's tandem approach allows for good versatility and comparatively quick dual extrusion prints, as less time is spent switching filaments. As all the heads are all mounted on the same carriage and move simultaneously, they can only do one print at a time. Nevertheless, it exists Independent Dual Extrusion System (IDEX) printers. The IDEX printers, which entail two print heads that operate independently, allow, besides the possibility of blending colors and materials, to simultaneously print two items through duplication or mirror print modes.

Color mixing The aforementioned methods select one filament at a time and are not designed to produce mixtures; the color palette is limited to the number of filaments used. Interweaving filaments of different colors along the outer perimeter of a print can create a perception of gradients. By changing which filament is the most visible, a gradient effect similar to dithering is achieved [Reiner et al., 2014] (Figure 22a). However, the surface is covered by a high-frequency weaving pattern, and the results exhibit color deviations along slanted surfaces. Kuipers et al. [2018] revisited this idea by taking into account the surface slope and creating the dithering by pushing more or less flow on layers, which makes them protrude more or less on side surfaces. With these dithering methods, the perceived color is slightly impacted by the view angle as the colors rely on occlusion.



(a) *(b)*

Figure 22: ***Left:*** *dithering method, where the colors protrude more or less to give a gradient perception. Here the blue is more perceived by the eye than the orange.* ***Right:*** *layering method with varying heights of mixed colors with perception improved by the translucency.*

Figure 23: Picture of a diamond nozzle able to mix three filaments. It works exactly the same as a standard extrusion nozzle, except in the melting zone, where there is a mixing caveat where all filament meet.

Multi-material mixing The multi-material mixing nozzle, also called "diamond" because of its shape, allows mixing from 3 to 5 filaments (Figure 23). All filaments are inserted in one nozzle and are independently controlled with their own extruder. The small mixing chambers allow the push of multiple filaments at once, mixing the colors.

While the aforementionned methods have different results depending on the viewing angle, Song et al. [2019] used translucent filament, visually blending the colors. In addition to the mixing nozzle, a micro layering, or sub-layering, extends the color palette of the three filaments. By stacking thin layers, it is possible to trick the eye that will blend the colors together, like an LCD screen (Figure 22b), but instead of the light intensity, the variation is on the layer height giving slightly curved strata.

3.1.4 Four and more axis machines

The vast majority of four or more axis machines are robotic arms. Except for those in laboratories, most publicly available robotic arms are used in architecture, where walls, made with special concrete, are built [Gosselin et al., 2016]. These robots print like a standard FDM printer, while those used in research may go out of the plane to minimize FDM printing defects. The defects will be explained in detail in the next section, and all the approaches using more than three Degrees of Freedom in Chapter 5, “Out-of-plane printing”. My work is halfway between standard FDM printers and custom-made solutions.

3.2 Constraints, defects and overcoming them

In FDM, there are different kinds of defects, those inherent to the materials' nature and the ones inherent to the method itself. These defects can alter the part's visual quality, physical properties, or both.

3.2.1 Stringing defect

The stringing defect is generally a visual defect that occurs during travel movement. As the extruded material is kept liquid inside the nozzle, a small leakage can occur during travel creating thin material strings between the parts. While not problematic when using one material as they can be quickly melted with a heat gun, it is not the case for multi-material or multi-color prints where the presence of the wrong polymer is not desirable. An easy way to mitigate this is to retract the filament inside the nozzle, limiting material dripping. However, color printers may have more issues, as retracting with a mixing nozzle may not be possible.

Figure 24: Stringing defect during travels.

3.2.2 Staircase defect

Due to the layer-by-layer construction, the discretized nature of the layered structure will exhibit an approximation defect called the staircase defect. This aliasing impairs the printed part visually at each slanted surface. Furthermore, the smaller the angles of the slopes are, the larger the artifacts. In some applications, the influence of the staircase defect can be detrimental to the structure itself. A containment problem arises depending on how the approximation is made. By choosing a slice plane to compute, there will be either a surplus of material, inflating the 3D model, a lack of material inside, or both [Pandey et al., 2003]. Reducing this defect is a significant focus of this thesis.

Figure 25: Staircase defect at every layer.

Adapt layer height

Most Additive Manufacturing processes are not bound to a unique layer height. The thickness is an important parameter that will directly affect the part. Thicker paths provide better bonding strength with the previous layer, resulting in good mechanical parts but rough accuracy, while thinner paths provide less bonding strength but better geometrical accuracy. In order to reduce the staircase defect, it is better to print the thinnest layers

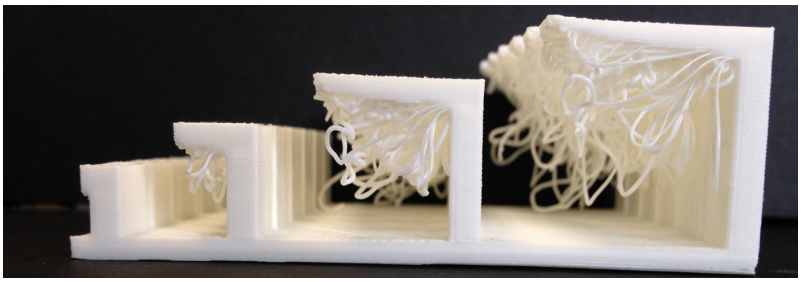


Figure 26: Print with overhanging features without support.

possible. Nevertheless, this comes with a significant drawback: time consumption. A first approach to mitigate this drawback consists of choosing the thickness using the local surface slope and ensuring that every flat area slice is well aligned with the model [Dolenc and Mäkelä, 1994]. The error evaluation of the slicing plane with respect to the local surface slope is called the cusp height. It is the more generally used metric to estimate the error. Most of the works [Dolenc and Mäkelä, 1994; Kulkarni and Dutta, 1996; Sabourin et al., 1996; Tyberg and Bøhn, 1999; Hayasi and Asiabanpour, 2013] considered the slices to have rectangular edges when evaluating this error. Hope et al. [1997] better approximate the surface using trapezium, while Pandey et al. [2003] used parabolic edges. Ma et al. [2004] proposed an adaptive approach where the user can pre-define the cusp height tolerance and the layer thickness bounds. Later, Alexa et al. [2017] proposed a technique that optimizes the layer height for every number of layers from the minimum number of layers to the maximum. These approach minimizes the overall *volume error*, that is the volume incorrectly assigned in the sliced part [Tata et al., 1998; Masood et al., 2000]. Users can thus choose any number of layers they want and get the optimal discretization to minimize the staircase defect.

One drawback of standard adaptive slicing is that it maintains the same thickness everywhere within the layer. To address this limitation, some approaches divide a part into multiple regions, using different layer heights in each [Tyberg and Bøhn, 1998; Sabourin et al., 1997; Mani et al., 1999; Wang et al., 2015].

Follow the surface

Another way to reduce the staircase defect is to move the trajectories slightly in the building direction (Z-axis). By slightly moving out of the slicing plane, it is possible to follow the object's shape more precisely. Song et al. [2017] decided to shift the paths in the building direction to match the surface. The layers remain the same as standard slicing, but the surface accuracy is significantly improved. More information on this is in Chapter 5, "Out-of-plane printing" [p.55].

3.2.3 Overhangs and islands

Managing overhangs and islands is one of the most significant constraints in 3D printing. When printed, they are the parts of the 3D model that have no material previously deposited below. Since the melted material is more or less liquid when it exits the nozzle, the material

will fall due to gravity. In the best-case scenario, this flowing material leads to some bad-looking layers, but in most cases, it leads to a failed print (see Figure 26).

The term self-supporting has been used in different contexts. In architectural geometry, it refers to a structure (i.e., an arrangement of blocks such as bricks) in a static equilibrium configuration. In FDM, instead of bricks, we consider the paths, and an object is considered not self-supporting if a deposition path is not supported from beneath [Vouga et al., 2012; Liu et al., 2013]. More specifically, in this thesis, regions or paths are considered self-supporting if the difference between the geometry of the lower slice and the current slice does not exceed half the deposition width [Chalsani et al., 1995] (45 degrees angle).

However, it is albeit possible to print straight lines mid-air between two distant points. This technique, called a bridge, makes it possible to draw a filament from one point to another with a distance sometimes exceeding ten centimeters.

Overhang detection

The first step in handling this potential failure is to detect problematic areas. For detecting overhangs, there are two types of approaches. The most common approach works directly on the input mesh, where every down-facing surface with too steep of an angle may be considered overhanging [Kirschman et al., 1991; Chalsani et al., 1995; Huang, Ye, Wu, Guo and Mo, 2009]. The second type of approach evaluates the difference between two consecutive slices [Huang, Ye, Mo and Liu, 2009; Huang et al., 2013]. Chen et al. [2013] uses shadows, where a slice casts a shadow by offsetting the current slice on the next one with the material-slope threshold, and every unshadowed area will need support. The computation to obtain those areas can be done with geometric operation on the geometry itself or a binary representation using images. Binary images have the robustness of 2D solid modeling operations like offsets, boolean operations, and Minkowski sums[Maragos and Schafer, 1986]. To not handle the cases of overhanging features, Reiner and Lefebvre [2016] proposed an interactive modeling tool ensuring the shape being created is self-supporting.

Supports and scaffolding

To enforce self-supportability, additional structures can be added. The supports (or scaffoldings) can theoretically alleviate this problem, but generating these structures is not trivial. Depending on the object, adding pillars that hold the overhang may not be possible without touching (and possibly damaging) other parts. Furthermore, we also want to minimize the number of supports because, when removed, they may impact the visual quality of the print at the interface layer between the model and support. If the supports are too firmly bound with the printed object, it can break the part in the worst case.

There are different approaches to creating such support structures. The first one is a naive approach called dense pillars (or walls). The material is laid layer-by-layer from the printing bed straight to the points requiring support. This approach results in the use of much material that must be removed during post-processing. The second one, lattice structures, is based on cell geometry. The individual cells are replicated to fill the space under the model that needs support [Hussein et al., 2013]. The third one, called sparse pillars, is quite similar to dense pillars. However, the points that need support are

down-sampled, resulting in less material used to support the model. Small overhangs do not alter the visual or structural quality, as the viscosity of the material makes it able to go over small bridges. The last ones are an improvement over sparse and lattice structures, where more elaborate structures are built instead of straight walls. Scaffold and Tree-like Structures attempt to support points by growing a structure down from the overhanging points. These complex structures aim to minimize the amount of material used by grouping or decimating supports into tree-like structures or scaffolding, similar to those used in building construction. Vanek, Galicia and Benes [2014] demonstrated that their tree-like structure used less material than dense pillar support structures. The scaffolding structure takes advantage of a 3D printer's ability to print short bridges supported at its endpoints but not beneath. This ability allows support to not start on other printed parts, for example, if another part of the object was below an unsupported portion. Dumas et al. [2014] describes a method of generating such scaffolding support structure and demonstrates that the bridge structure is more efficient and stable than a tree-like structure.

Most of the time, the overhangs' visual aspect of the inner part is not a problem compared to more essential aspects like mechanical integrity or time consumption. Sometimes both can be improved with specific infills and optimization processes. The user will need specific internal supports if the infill is not sufficiently dense or if the objective is to minimize the amount of material needed to do lighter and faster prints. Internal supports will be more extensively explained in Chapter III, "Referenceschap:Infills". In order to reduce the visual artifacts created by these methods, a dissolvable support material such as PVA can be used on a dual-extrusion machine. The whole support structure or only interfaces between the supports and the object can be printed with dissolvable material.

Orientation

The orientation of a shape plays an important role in the surface quality [Livesu et al., 2017]. By changing the orientation of the printed object, it is possible to diminish the number of unsupported parts [Frank and Fadel, 1995; Allen and Dutta, 1995; Cheng et al., 1995*b*; Alexander et al., 1998; Majhi et al., 1999; Ezair et al., 2015; Zhang, Le, Panotopoulou, Whiting and Wang, 2015], or minimize the reproduction error due to layering [Thrimurthulu et al., 2004; Cheng et al., 1995*a*]. The orientation can also be optimized to reduce the cost of supports, in time or material [Ingole et al., 2011; Gurrala and Regalla, 2012; Di Angelo et al., 2020], improve the stiffness [Umetani and Schmidt, 2013; Raut et al., 2014] or the surface quality [Pandey et al., 2004; Zhang, Le, Panotopoulou, Whiting and Wang, 2015; Buj-Corral et al., 2019].

Part decomposition

When no support is desired, the initial virtual object can be split into several smaller parts that do not require support [Hu et al., 2014]. Research on part decomposition can have other objectives, such as optimizing the space to fit in another volume [Luo et al., 2012] or a box for packaging [Attene, 2015; Vanek, Galicia, Benes, Měch, Carr, Stava and Miller, 2014]. Most of the time, part decomposition is coupled with part orientation to minimize staircase defect [Hildebrand et al., 2013; Wang et al., 2016]. The segmentation and printing

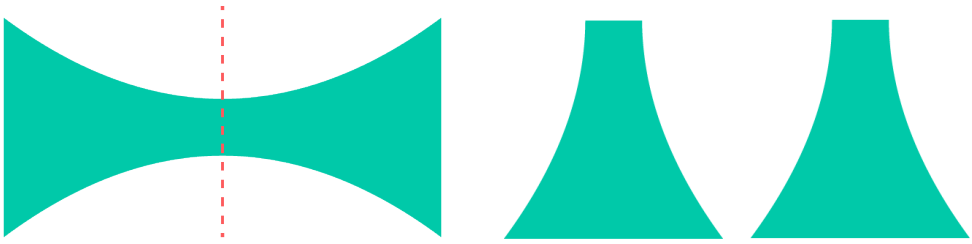


Figure 27: Print with overhanging features cut to be printed without supports.

orientation cannot be chosen only to minimize the defects, as the different cuts must be able to be joined once printed. Such methods can be combined with other manufacturing techniques like milling to get more accurate results [Herholz et al., 2015].

3.2.4 Delamination

Delamination is a structural defect that occurs when deposited material has already cooled down before adding another layer over it, leading to a lower bond between them. This defect can have a critical impact on the printed part's stiffness. Delamination is mainly related to materials and printing parameters (speed, temperature), and using an enclosure, heated or not, can reduce delamination for specific materials. The delamination defect can easily be observed with materials such as ABS or PEEK, where the bond between layers is very sensitive to temperature fluctuations and airflow. Using acoustic emission analysis, Barile et al. [2018] observed that delamination is reduced and toughness increases if extrusion temperature increases. They showed that acoustic emission analysis provides more predictable information than visual information about the material under test. However, as other defects are also related to temperature, there is a trade-off between layer bonding, stringing, and burr defects if the material does not cool down fast enough.

In Chapter "Adaptive infills" Chapter 7, "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91], we proposed a staggered pattern to improve the bonding between layers.

3.2.5 Shrinkage and warping

Warp deformation is an important metric to evaluate the quality of an FDM prototype. When the heated polymer is deposited, it starts to cool down. By doing so, its volume will diminish, implying a shrinking of the already deposited print paths, leading to a warping issue as uneven heat distribution creates internal stresses within the part [Yaman, 2018]. Such deformation may lead to a printing failure. Like delamination, the leading cause of this issue is material and parameter choice. By correctly adjusting the nozzle temperature and printing speed, the warping deformation can be reduced significantly [Alsoufi and Elsayed, 2017].

Wang et al. [2007]'s mathematical model provides a scientific tool for controlling and adjusting the deformation. Based on this model, optimizing the printed part to minimize

warping is possible. Alsoufi et al. [2019]'s experiments exhibit the differences between two materials (PLA and PLA+) and the importance of alternating the infill printing directions to distribute the internal stress. Dilberoglu et al. [2019] proposed an infill structure that reduces the shrinkage of internal holes with auxiliary line segments. Nevertheless, it does not improve the outer part.

3.2.6 Seams

Another visual defect, called seams or zippers, appears when the nozzle travels from the outer part to the inner part (or the other way around) or during layer change. This defect visually impairs the printed object by adding small holes on the surface (at least one seam per layer and connected area). A technique to remove the seams is never to print something else than the outer shell or have any layer change. Layer change can be avoided with a continuous increment over the building direction. This technique, spiralize or vase mode, is a common feature on available slicers (see subsection 5.2.4, "Spiralize, helical or continuous" [p.59]). Except for spiralizing, there is no way to eliminate the seams, but they can be partially hidden or set in a less visible area. A way to estimate the visibility of an area is to use *ambient occlusion*. *Ambient occlusion* is a shading technique that calculates how much a surface is exposed to ambient lighting. So the darker the surface, the less visible it is. By using this data Ray et al. [2010] and Song et al. [2017] obtained a substantial surface improvement. Sometimes, a concave area will be chosen for the seams, but the seams can still be visible from the main or intended point of view. IceSL software [IceSL, 2022] overcomes this by giving the user a way to specify the location of the seams.

Instead of hiding seams, Baumann and Roller [2017] purposely uses them to embed watermarks to ensure the traceability of an FDM manufactured object without the alteration of both the structure and geometry.

3.2.7 Printing time

The main limit in prototyping with FDM is to minimize the time between the design and the reification. There are different ways to reduce the time spent between the start of the designing phase and the prototype phase.

Fast fabrication

The machine's characteristics bound the printing time of an object. If we have to print a volume, the time spent will be close to the total volume divided by the amount printed by seconds. If we cannot change the machine, reducing the time is only achievable if we decrease the volume.

One way to diminish the volume is by hollowing out the object such that only the surface and the necessary filling remain. See the section dedicated to infills for more information.

Another way to decrease the volume is to replace some parts with existing objects. Mueller, Mohr, Guenther, Frohnhofen and Baudisch [2014] uses well-known plastic bricks to replace some parts of the surface so that less time is spent covering the surface.

Instead of replacing part of the surface, Mueller, Im, Gurevich, Teibrich, Pfisterer, Guimbretière and Baudisch [2014] went a step further and decided to remove material and print only part of the surface. The volume is drastically reduced by printing a wire mesh of the surface, resulting in lower printing times by orders of magnitude.

Interactive design

Instead of reducing the time spent on printing, the print can start as soon as possible. The standard process is to design the part, export and slice it, then load it into a 3D printer. However, what if the print process starts simultaneously with the design?

In 2010, Willis et al. [2010] introduced the notion of interactive fabrication. Since then, several approaches have been proposed for hands-on digital fabrication. Depending on the technology, some interactive processes will not tolerate changes afterward, like laser cutting or FDM, but machines that embed extrusion and milling can. An example of fabrication of functional objects using a laser cutter can be found in Construcatble [Mueller and Baudisch, 2015], where the user draws directly onto the board where it should be cut, and the machine operates afterward.

A first approach to 3D printing by Peng et al. [2015] enables non-experts to design 3D digital models from scratch. The designer uses a wax extruder to make a 3D object, like a 3D pen, and the computer transforms each movement into parametrized curve trajectories to obtain a smoother and cleaner version of the model. It was then enhanced with Virtual Reality (VR), where the wax extruder was replaced with VR controllers [Peng et al., 2018]. Weichel et al. [2015] allows the user to have a fast loop between manual shaping, digital milling, and extrusion of synthetic clay, giving fast feedback and control over the shape.

On The Fly[Peng et al., 2016] is an extension of Wiremesh print, leading FDM into interactive design. On The Fly and Wiremesh Print will be more extensively presented in Chapter 5, “Out-of-plane printing” [p.55].

All these interactive fabrication systems offer immediate, tangible feedback that can benefit the design process but are often slow because designers still have to instantiate the physical models themselves.

Chapter 4

Infills

Infilling traditionally consists in making objects printable and more sturdy by adding material inside the print. This filling can be done through the whole object or just in some part of it, respectively named dense and sparse infills.

While filling surfaces introduces some concerns for all technologies (see Chapter 1, "Additive manufacturing technologies" [p.15]), I will only focus on vector-based technologies.

One of the primary concerns of 3D printing is speed. Depositing material or just going from a point to another one takes time. As seen before, the slicing software has to optimize the order of the generated tool paths to minimize the travel movement's length (section 2.2.1 "Slicing Performances" [p.25]). Moreover, abrupt changes in direction are also a concern, as every sharp turn, like a half-turn, implies that the in-use motors must change direction, decelerating before accelerating again to compensate for inertia. This lead to the second concern, the structural soundness, as small holes or unexpected material deposition by leakage can appear at the paths' stops and starts or during travel movements.

To summarize, we must avoid travels, sharp turns, and gaps. The following sections overview the different dense and sparse infills implemented in most modern slicers.

4.1 Dense infill

Various methods have been introduced to generate dense infills covering a slice (Figure 28). The simplest way to cover an area is by crossing the surface in straight parallel lines with a boustrophedon-like [Golomb and Selfridge, 1970] printing scheme called *Linear infill* (or raster/zigzag pattern) [Held, 1991]. This pattern has to undergo many small travel movements and abrupt changes in direction, leading to a high number of stops and starts, increasing the print time. It is a simple pattern, but as seen before, stops and starts should be minimized. Moreover, printing the inner part with a pattern having abrupt changes near the object's surface, such as linear infill, may impact the outer surface, inducing a wavy appearance, as all the stops and starts are located there.
Concentric infill [Held et al., 1994; Yang et al., 2002], also called contour parallel, partially ignores this problems by following the object's surface. As the direction changes depend only on the object's surface, they may be smoothed, leading to fewer printing interruptions. However, the naive implementation of concentric infill leads to many holes between paths. Those gaps appeared as we tried to fill a layer with paths of constant width. In most cases, printing thinner or wider paths is possible by managing the quantity of extruded

material. An enhancement of the concentric pattern (to better cover the surface) is to use this freedom to control the deposition width and, thus, reduce the remaining gaps (*Variable width contouring*)[Kao and Prinz, 1998; Ding et al., 2016; Jin, Du and He, 2017; Xiong et al., 2019; Hornus et al., 2020; Kuipers et al., 2020].
Most modern slicers use a combination of concentric infill for the paths close to the surface (sometimes called shells) and a linear infill for the remaining parts [Ding et al., 2014]. Nevertheless, these methods are still generating a lot of stops and starts.

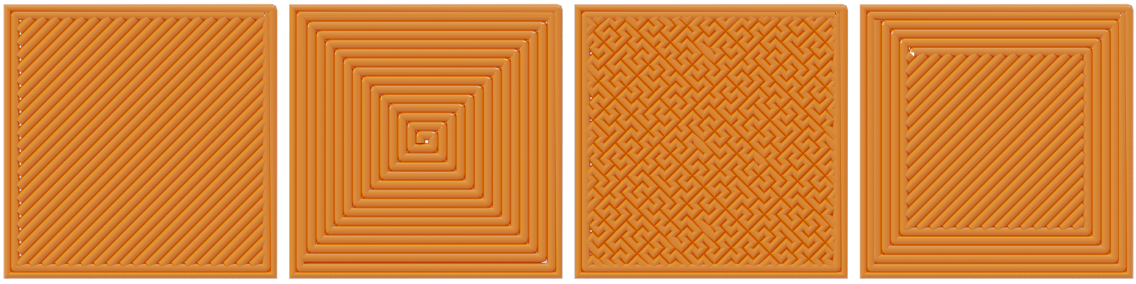


Figure 28: Different dense infill patterns, from left to right: linear, concentric, spiral, and combination of concentric and linear.

A better way to fill the slice while ensuring the continuity of the deposition would be to use curves to cover the space with evenly spaced lines. Peano's curves are a group of mathematical functions that fill the space, hence the more common name of the space-filling curve or, more generally, *Hilbert curve* in 3D printing. The commonly implemented space-filling curves are Peano [Peano, 1890], Hilbert (a variant of Peano) [Hilbert, 1891], and Moore's (loop variant of Hilbert) [Moore, 1900] curves. These curves are continuous and can be used to minimize the number of travel movements [Papacharalampopoulos et al., 2018]. However, a common drawback of the space-filling curves and contour-parallel patterns is the lack of direction bias, which aligns the infill paths from one layer to the subsequent one. This alignment may induce weak points, fragilizing the part as any defect in the infill may be propagated through the whole print. In addition, direction bias allows the generation of cross paths, preventing this alignment. Besides, none of the approaches above allows the user to change the orientation of the paths inside a slice.

A more generic approach to curve filling would be to draw evenly spaced curves that follow directions. Among the infills attempting to follow directions, more recent approaches give additional functionality to the object by considering the deposition path orientation. Doing so allows for greater control over mechanical strength, but it can also control how an object will deform under an external force or heat [An et al., 2018]. This control can be done by exploiting the anisotropic character of the filament deposit, which is more robust in the deposit's direction in the plane [Rybachuk et al., 2017] or all directions [Singamneni et al., 2012; Fang et al., 2020], and emphasizing it by including reinforcing fibers [Gray et al., 1998; Heidari-Rarani et al., 2019].

Evenly-spaced curves and streamlines are a long-standing problem in Computer Graphics and can be sorted into three families: approaches tracing streamlines in a vector field from starting points [Jobard and Lefer, 1997; Hertzmann and Zorin, 2000; Mebarki et al., 2005; Spencer et al., 2009], approaches based on global periodic parameterizations

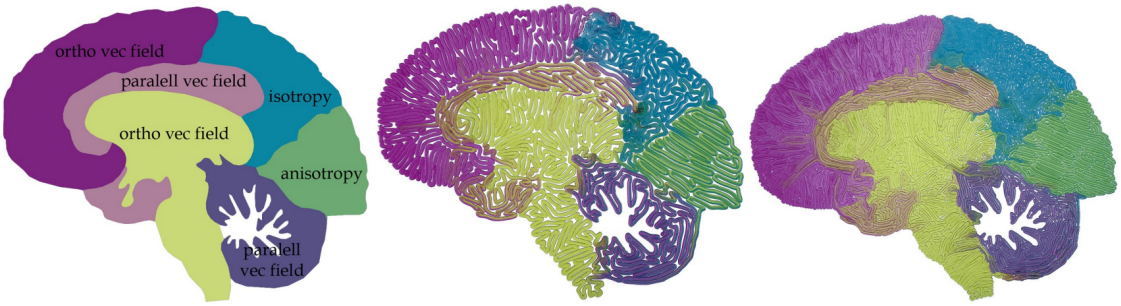


Figure 29: Dense Hamiltonian cycle infill with direction bias, isotropy, anisotropy, vector field alignment, and zoning [Bedel et al., 2022].

[Knöppel et al., 2015], and finally approaches splitting and merging covering curves, as pioneered by the seminal work of Elber and Cohen [1996]. One of my approaches most closely relates to this later family, particularly its image space variants [Groen et al., 2019], detailed in Chapter 6, "Procedural band patterns" [p.75]. Nevertheless, all those curves have a lot of singularities, T-junctions, and abrupt stops at the outline of the object.

Bedel et al. [2022] overcame this problem using *Hamiltonian cycles*, offering control over material deposition orientation following a strictly continuous path. The space-filling trajectories can be optimized for isotropy, anisotropy, vector field alignment, and zoning (see Figure 29). In Chapter 7, "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91], I will present another dense infill method with controllable path orientations.

However, printing a filled object is quickly expensive in time and materials as the volume increases to the cube of the object scale.

4.2 Sparse infill

Additive Manufacturing is often used for rapid prototyping or creating decorative objects. It is then interesting to have structures saving time or raw material. While saving time is only achievable for 'vector-based' technologies, saving raw material is a challenge for both 'vector' and 'rasterization' technologies that do not trap the material if a closed pocket is printed. Trapped material should be avoided, as unsolidified material can be harmful when released. Sparse infills are still a compelling area of research since they are available to the most widespread technologies.

Special care must be taken when hollowing an object to ensure that every deposition path will print correctly, ensuring it is supported by previously deposited material or a bridge can be successfully created.

Infills can be separated into two main categories, 2.5D and 3D. The 2.5D infills are vertical extrusions of patterns along the printing axis, while 3D infills evolve in the build direction.

4.2.1 2.5D infills

All dense patterns can be used as sparse infills by adding space between the paths. One of the main purposes of these 2.5D infills is to ensure the object will print correctly.

The simplest 2.5D infill is just extruding a linear infill with space between the parallel lines. By controlling the space (or density), the user can ensure that the object will print properly, managing inner overhangs. Nonetheless, other characteristics than supportability are interesting. For example, an object can be sturdier in multiple directions by crossing sets of lines such as rectilinear, triangle, and line patterns in Figure 30.

Among all the possible 2.5D infills, the most iconic pattern is the *Honeycomb infill*. According to Moradi et al. [2019], this pattern is an adequate internal fill pattern to manufacture low-weight PLA components with the ability to undergo slight deformation. The printed tensile specimens showed higher load and elongation before failure. However, sudden brittle fracture is usually observed in PLA specimens at yield strength and breaking load. For this infill, increasing layer thickness positively impacts the strength of specimens and the print time, at the cost of a slight augmentation of part weight and material consumption. Their results unveil that infill percentage is the second significant parameter, after layer thickness, influencing part characteristics. A comparison between rectilinear, triangle, line, and honeycomb infills (presented in Figure 30) gives honeycomb a slight advantage among energy-absorbing and peak load [Ma et al., 2020] but lower print times due to the high number of direction changes.

Other structures are available when the aim is more focused on mechanical properties. One of the best performing 2.5D structures uses an extruded Voronoi diagram but with an optimized density to handle loads [Liu et al., 2021]. This optimized anisotropic porous structure has revealed superior performance to any non-optimized or isotropic porous structure under the same mechanical conditions, outperforming the honeycomb pattern. This process of optimizing density can be used for different 2.5D mechanical parts without limitations on the geometrical shape. Additionally, many additive manufacturing processes can manufacture the generated porous structures.

Alvarez C et al. [2016]'s investigation on how the infill percentage affects the time and resistance concludes that, in general, infill values between 50% and 98% are not recommended as the gains in mechanical strength are ludicrous when taking into consideration the additional printing time required. Furthermore, Podroužek et al. [2019] demonstrated that 2.5D infill patterns extended into 3D via vertical extrusion only outperform 3D infills in the printing direction, which is too specific and does not reflect the actual use case. Moreover, while the 2.5D infills are self-supporting at constant density, they are not suitable for density gradation over the part, hence the interest in 3D patterns.

4.2.2 3D infills

Among the different 3D infills, some aim to compromise between stiffness and printing time, while some infills are more focused on material saving while ensuring self-supportability. Some infill structures also give some extra-mechanical properties to the part, such as forcing the deformations of a flexible material to be only in one direction.

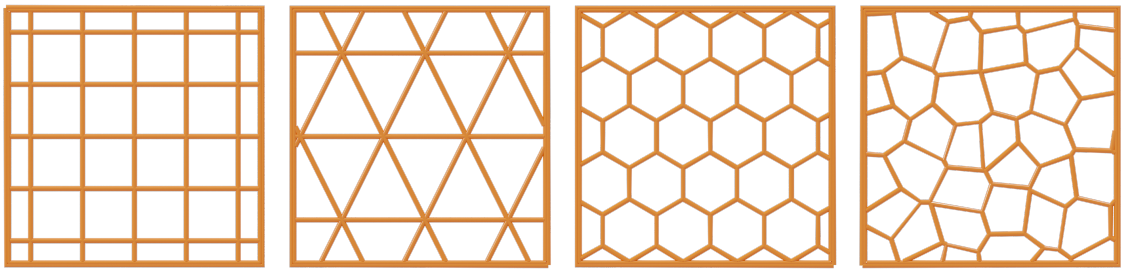


Figure 30: Different sparse infill patterns with 10% density, from left to right: rectilinear, triangle, honeycomb and Voronoi. Rendered in Blender using actual GCode from IceSL.

Simple geometric infills

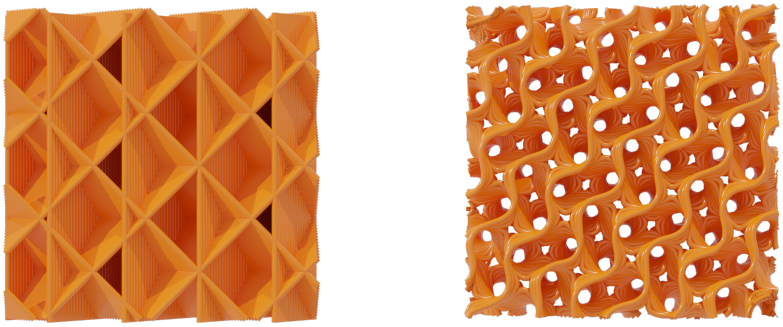
An infill must ensure self-supportability to be usable, but it better outperforms the others on print time, structural integrity, or other properties to be adopted and used. One desirable property can be to have a somewhat stiff part using a low amount of material without optimizing for specific criteria.

Cubic infill [Lefebvre, 2015] (Figure 31a) is swift, robust, and allows for fast printing by crossing sets of straight lines. It is well suited in all the printers configurations (Delta, CoreXY, H-Bot, and others), as the lines cross at 120 degrees, ensuring multiple motors participate in the movement for better accelerations. This method is further extended to a hierarchical scheme [Dinh et al., 2015], allowing faster print by using subdivisions like octrees to reduce the number of paths required while ensuring self-supportability. The subdivision pattern of Cubic infill and other hierarchical techniques [Bächer et al., 2014] is therefore abrupt, and the *T-shaped* junctions lead to a higher risk of breakage. To overcome this problem, an approach with splitting and merging curves [Elber and Cohen, 1996; Groen et al., 2019] allows for a better splitting pattern. Nevertheless, the splittings and mergings only occur on a front, where all the curves will split in two. I have partially addressed this problem in Chapter 6, "Procedural band patterns" [p.75], where the curve's splits follow a given split ratio.

These infills are *closed-cell*, meaning the unused material will be trapped inside closed pockets while printing with some resin-based and powder-based technologies. In these cases, using an *open-cell* structure or another infill structure will allow for circumventing the problem if the outside is not fully close (to evacuate the unsolidified material). The *Gyroid infill* (Figure 31b) is a pattern with no closed-cells and easily defined by the following equation [Schoen, 1970]:

$$\sin x \cos y + \sin y \cos z + \sin z \cos x = 0$$

In addition to its open scheme, this filling structure has a similar tensile resistance in X, Y and Z directions, thanks to its cubic symmetry. Moreover, the Gyroid's isotropy and structure are coveted in radiotherapy, where they can be used to print phantoms. A phantom is a mass of material that reacts similarly to human tissue to investigate the effect of radiation on human beings [DeWerd and Kissick, 2014]. Phantom materials can range from water to complex chemical mixtures that faithfully mimic the human body as it should interact with radiation. Madamesila et al. [2016] demonstrated the feasibility of controlling



(a) Cubic infill *(b) Gyroid infill*

Figure 31: Two homogeneous infill in the main axes. The Cubic infill is a closed-cell pattern, which can trap unsolidified material (e.g. powders or resines) while the Gyroid infill is an open-cell pattern.

the mathematical composition of Gyroids to produce a range of Hounsfield unit values (radiodensity measurements, frequently used in CT scans) using manufacturable printing parameters. Their results were consistent in every orientation, giving a predictable behavior due to the isotropic nature of the pattern.

While the previous infills' aim is more generic, like self-support or mechanical resistance, regardless of the shape or its usage, some infills aim to be printed using this knowledge to better target properties. Moreover, Gyroid and Cubic infills are paths generated and only depend on the input density, meaning that by changing the density across layers, some paths may not be self-supporting.

Minimizing material consumption

One recurrent goal is to have an excellent exterior while minimizing the quantity of material used or the print time. It should be observed that the printing time is, to some extent, correlated to the amount of material deposited.

A first approach to hollowing a part is solidifying the walls using *Ribbed support* structures. These ribbed structures are created by slightly thickening the object's outlines regularly. These material protrusions stiffen the various walls using relatively little material. A first attempt to optimize the location of those ribs was done to reduce the deflection of plates [Lam and Santhikumar, 2003], and then in shell-like objects [Li et al., 2017]. Those two techniques were used to improve the strength of shell-like objects, but such structures can also be used for a self-supporting purpose. To apply rib structures to hollow objects, Tricard, Claux and Lefebvre [2020] uses a greedy support generation algorithm to create compact hierarchies of rib-like walls. The walls are progressively eroded, straightened, and merged with the interior walls to have a structure as light as possible that guarantees that every point in a layer is supported. Except for some particular cases like small slopes or dreadfully located islands, this *modus operandi* has the lightest internal support structure.

Instead of creating structures inside the object, another way to reduce the unnecessary material is by removing it and progressively carving the inner volume. Hornus and Lefebvre [2017]'s *Iterative carving* approach minimizes the amount of material required

(a) Bridge structure *(b) Some results*

Figure 32: Iterative carving results examples [Hornus and Lefebvre, 2017]. On the right image, the transluency of the material allows to appreciate the location of the internal supports.

by iteratively removing chunks of material. This method is fast and memory-efficient, as it only has to do a sweep iterating over all the slices and only store the last two. A limitation of this approach is that it requires additional internal support due to the emergence of local minima and islands. They use internal bridges as supports but stated that bridges could also become long in large objects while supporting prominent features above them (Figure 32a). However, it does not seem problematic on their test prints (Figure 32b).

Sometimes, minimizing the printing time or the material used while complying with some constraints is desired. The constraints, often mechanical, will influence the deposition process. It is generally necessary to make an optimization pass to find a compromise between the material used and the desired behavior. This optimization can be done at different scales, directly on the generated structure or through an intermediate representation. This intermediate representation is often done with the user's initially optimized or painted fields.

Structural properties

Instead of a path infill, Wu, Wang, Zhang and Westermann [2016] developed a geometrical infill (*Rhombic infill*) that subdivides rhombic structures in a global optimization process, using more (and smaller) cells in the regions of high compliance. This optimization considers the manufacturability of the generated shapes, and the overhang constraints are seamlessly satisfied as the rhombic cells can automatically ensure manufacturing requirements on overhang-angle and wall-thickness. The rhombic cells are then carved out of the volume. Like cubic infill, this approach allows for faster prints using a subdivision strategy. Rhombic infill is designed to optimize for any given energy, such as sustaining a load or ensuring the self-balancing of the part. Self-balancing a part is also done by Prévost et al. [2013] by controlling the center of the mass. Moving the center of the mass is done by optimizing the internal distribution of material inside the object [Yamanaka et al., 2014] and can be optimized to compel for other static and dynamic behaviors, such as floating [Wang and Whiting, 2016], spinning [Bächer et al., 2014], or swinging [Zhao, Hong, Lin, Jin and Xu, 2016].

Wu et al. [2019] opted for a 3D approach by optimizing the orientation of lattice structures that conform to both the constraints' principal directions and the optimized shape boundaries. This method optimizes the shape (including its topology) and the distribution

of orthotropic lattices within the shape to maximize stiffness under application-specific external loads. Starting from the previously optimized lattice configuration (i.e., orientation, porosity, and anisotropy), it extracts a globally consistent lattice structure with field-aligned parameterization. Other approaches optimize truss [Wang et al., 2013; Zhang, Xia, Wang, Yang, Tu and Wang, 2015], bone-like Wu, Aage, Westermann and Sigmund [2017] or foam-like Lu et al. [2014] structures within the volume to obtain a rigid but lightweight 3D prints based on a topological optimization problem under local material density constraints.

Controlled deformations

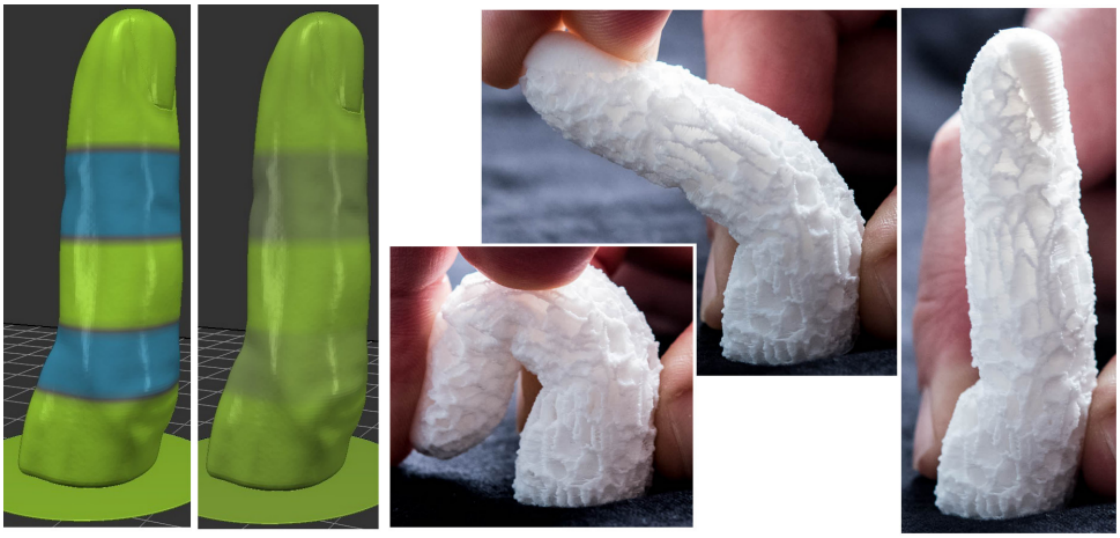


Figure 33: Finger with hand drawned control fields (anisotropy and density), folding at the joints [Martínez et al., 2018].

The diversity of base materials opens the field to numerous possibilities, and we can extend the properties of those materials with different kinds of infills. The properties can be optimized based on the microstructure characteristics or in combination with the underlying material. The stiffness in a direction can be straightened or loosened by simply orienting a sparse linear infill and using flexible materials such as TPU, which enables the bending of parts.

One approach to controlling the elasticity was made by generating an open-cell Voronoi structure [Martínez et al., 2016]. These *Procedural Voronoi Foam*-like objects were printed using a photopolymerization printer, and the cells were generated using a controllable density. The main advantages of their approach stem from the implicit formulation of stochastic microstructures, as the evaluation at slicing time has significant computational advantages. Rather than requiring a global optimization process, these microstructures are directly generated to exhibit a specified elastic behavior. The aperiodic and stochastic nature of the foams provides a simple and efficient way to grade the structures and conform to target elasticity fields [Roberts and Garboczi, 2002; Ashby, 2006; Luxner et al., 2007; Martínez et al., 2017]. Martínez et al. [2018] extended the Voronoi foam further with a

specific distance operator that enforces fabrication constraints. The printed parts exhibit a controlled behavior with TPU, making some areas prone to flexion and buckling (Figure 33).

While the previous methods use a stochastic distribution, Martínez et al. [2019] generated a structure with Voronoi-like cells but using a honeycomb grid distribution to define the center of the cells. They defined star-shaped metrics for each cell to design mechanical metamaterials based on these Voronoi diagrams. This cell structure supports interpolation between arbitrary metrics with a wide range of mechanical properties, including isotropic, tetragonal, orthotropic, and smoothly graded materials Figure 34. This concept of by-cell metrics was extended to 3D-shaped cells [Efremov et al., 2021].

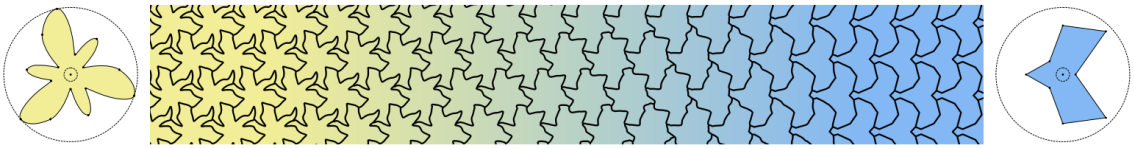


Figure 34: Star-shaped structure with gradually evolving distance [Martínez et al., 2019].

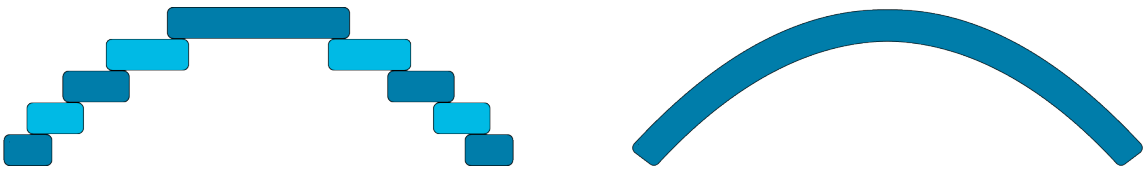
While the previous methods use cells to generate the paths, it can be desirable to directly control the printing paths' orientation. This oriented path extraction can be done using a phasor noise [Tricard et al., 2019], an orientable noise that can generate trajectories in space. This noise was then used to generate structures allowing a rigidity gradation for given orientations within an object [Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre, 2020]. This microstructure control allows the behavior to evolve between isotropic to highly anisotropic. In Chapter "Adaptive infills" Chapter 7, "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91], we extended this noise to an orientable dense infill.

Chapter 5

Out-of-plane printing

Conventional 3D printing has many drawbacks, such as surface roughness due to the staircase effect, scars and additional material consumption caused by support structures, while the possible delamination diminishes the robustness of some parts. These problems can be substantially reduced or eliminated using out-of-plane printing, commonly called curved printing (or Curved Layer Fused Deposition Modeling (CLFDM) / Curved Layer Fused Filament Fabrication (CLFFF)). A basic example of curved printing, Figure 35, shows how it can affect the structure of layers.

While slightly curved printing is possible with a conventional 3-axis printer, it is best to use a machine with at least five axes. The advantage of 5-axis machines comes from the possible orientations of the nozzle, wherewith 3-axis, the nozzle always remains perpendicular to the planar layer to be deposited. With 5-axis machines, the nozzle may remain tangent to the part's surface, and being tangent to the part's surface makes it possible to improve the accuracy of parts by following variations in curvature, therefore, avoiding the staircase effect. Another advantage of printing with a 5-axis machine is avoiding supports, as rotating the bed or the head removes the overhangs.



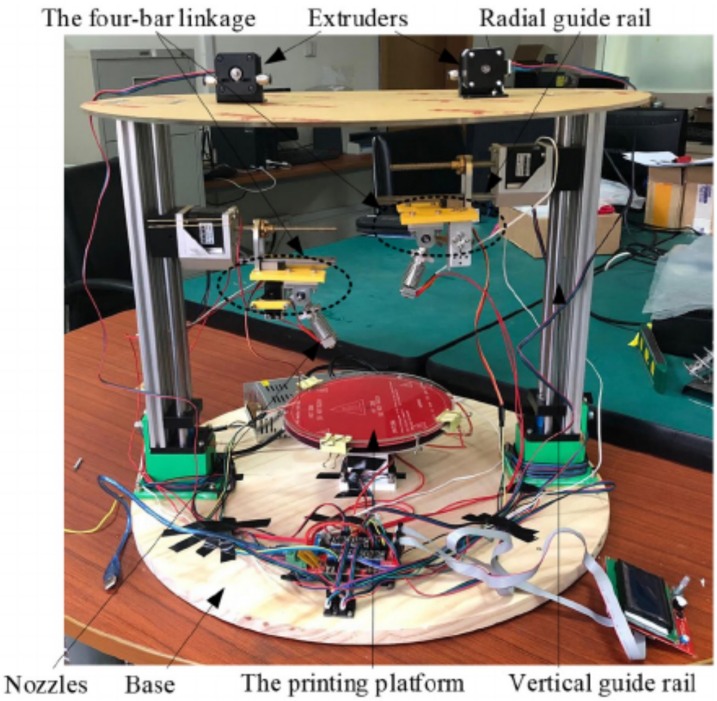
(a) Planar layers representation *(b) Curved layer representation.*

Figure 35: Simple comparison of curved and planar printing.

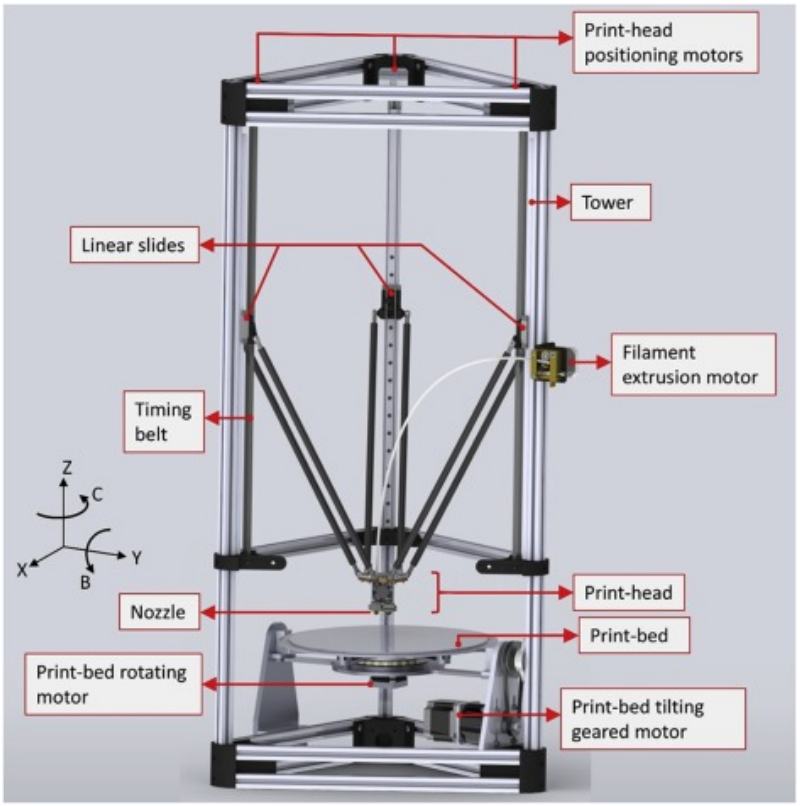
5.1 Machines and modifications

The mobility of FDM machines has an impact on the limitations of the geometry of the parts. Conventional machines have three Degrees of Freedom (DoF). Consequently, the extruder is usually restricted to remain in a vertical position. This fixed orientation restricts the curved profiles' slopes from being achieved via curved deposition using a machine with 3 DoF. This limitation has motivated several researchers to upgrade machines' mobility, even though Cartesian configurations can be employed for CLFDM up to a geometrical limit (between 10 and 30 degrees, depending on the hardware). However, the main limitation lies

in keeping the nozzle normal to the intended curvature, even if the material can be deposited correctly. In contrast, four- and five-axis milling machines solve this problem by changing the tool orientation to be normal to the fabrication surface. Modification of conventional machines has been achieved through different setups. The more economical way is to modify a standard printer to use additional DoF. Those modifications are generally done on a Cartesian or delta printer (see Figure 36). The majority of robots used for multi-degree of freedom printing are composed of 6 axes because they are directly accessible via a minor modification of industrial robots. However, there are custom made printers using 7, 8 or even 12 degrees of freedom.

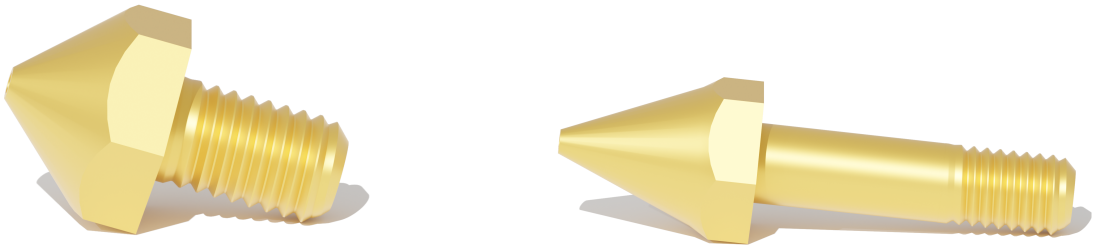


(a) 4-DoF polar printer with rotating nozzle [Zhao, Li, Shen, Jiang, Guo and Gao, 2020].



(b) 5-DoF printer, composed by a delta printer with a tilting and rotating bed [Isa and Lazoglu, 2019].

Figure 36: Different machine architectures and modifications.



(a) Ordinary nozzle *(b) Interference-free nozzle*

Figure 37: Comparison between a classical nozzle and an interference-free nozzle.

5.2 Conventionnal printers

As most printers are 3-axis machines, some algorithms using the ability to deposit material slightly out of the plane to reduce the problems mentioned earlier were developed.

5.2.1 Smoothing the surface

The staircase defect badly affects the object's surface with standard slicing. This effect becomes prominent on slanted surfaces or when using thicker layers. In order to print the part more accurately, the number of layers needs to be increased [Sabourin et al., 1996], leading to order of magnitude slower build times. In addition to part orientation, the best way to reduce the staircase defect is to extrude the material while closely following the surface. In 3-axis printing, it is possible to change the nozzle height within a layer slightly. Song et al. [2017] used this property to smooth the surface by shifting the paths within a layer to get closer to the surface mesh. As the layers remain the same as standard slicing, the print time stays in the same bounds, but the surface accuracy significantly improves. Ahlers et al. [2019] smoothed the surface across multiple layers by extracting the parts of the surface that can be printed without collision of the tool-end with previously deposited material. Then he generates a set of paths that follows the surface to print the object, resulting in curved layers (Figure 38).

5.2.2 Printing lattice structures

A specific case of surfaces that can be smoothed is curved lattice structures with a constant height.

The lattice arc structures are constructed with traversal arc rasters and orthogonal arcs [McCaw and Cuan-Urquizo, 2018; Cuan-Urquizo et al., 2019]. Those lattice structures were fabricated following a parametrization that places the arc out of the plane. The mechanical properties of several lattice-arc samples with different porosities were tested. In Cuan-Urquizo et al. [2021], it was confirmed that straight arcs stiffen the structure experimentally and computationally.

Lattice structures and other auxetic structures have a lot of valuable properties, such

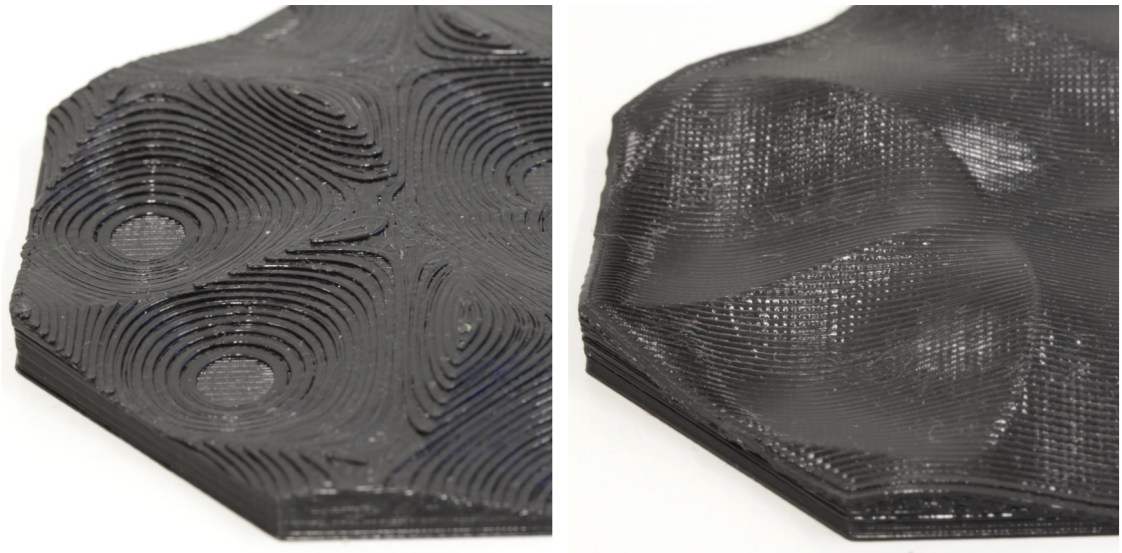


*Figure 38: Comparison between standard and smoothed surface generation ([Ahlers et al., 2019]). **Left:** we can see the layers and the staircase defects. **Right:** we can still see the tool head paths, but the surface finish is smoother.*

as minimizing bending stress or better distributing it Lim [2015]. Martínez et al. [2019] demonstrate this by creating new metamaterials with controllable Poisson's ratio, see Chapter 4, "Infills" [p.45]. The production of lattices through additive manufacturing has created many opportunities for new metamaterials that can be used for shock absorption, stress distribution, and reduction of vibrations and have direct applications in sportsSanami et al. [2014], aircraft[Budarapu et al., 2016], or biomedical engineering to control the porosity of lattices to promote tissue and bone growth Taboas et al. [2003]; Tan et al. [2017].

5.2.3 Following a field

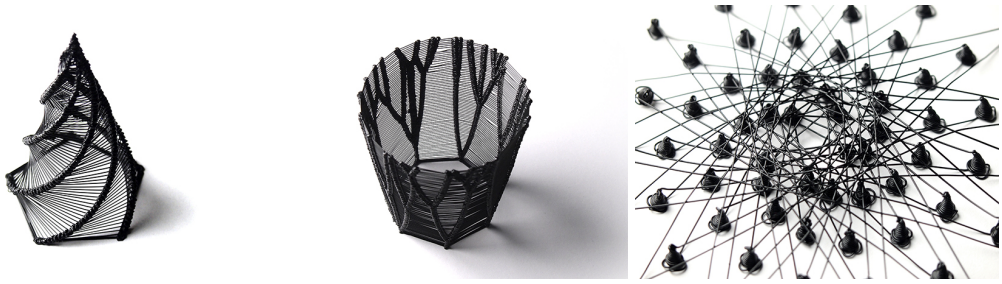
As seen with lattice arc structures, orienting the paths in space can improve the strength of a part in specific directions. Given a user-specified parameterization Ezair et al. [2018] generates curved toolpaths in a volume following this field. The curved layers are generated along with the parameterization's iso values. The layers that are too close are trimmed to ensure that the appropriate coverage is achieved. Then, all the layers are covered with curves to extract the toolpaths. As in Song et al. [2017], the 3-axis toolpaths' set is obtained by splitting and ordering the curves to avoid collisions during deposition.

5.2.4 Spiralize, helical or continuous

The spiralized, helical, or vase mode is the most popular aesthetic approach for out-of-plane printing. This printing technique allows a continuous deposition path where the nozzle constantly moves up slightly as it prints, with no more traditional layers [Yigit and Lazoglu, 2019]. It can be seen as a spring with a unique thread to make the whole shape and be used as a spring [He et al., 2019]. Almost any printer can print a model with just a single wall

thickness, reducing printing time and material. Like iterative carving's infill, vase mode generates lightweight objects. However, a significant difference lies in the impossibility of printing the top surfaces or steep slopes, as there are no internal or external support structures. It is better suited for objects with continuous cross-sections, without holes or gaps in their geometry, hence the name "vase mode." The continuous deposition removes all the defects, such as seams, and makes it more appealing to the eye. Diourté et al. [2021] makes it available to 5-DoF machines, Wire Arc Additive Manufacturing, which also are extrusion-based hardware.

5.2.5 Artistic work



(a) Samples of prints using bridging and stringing to create structures [Lia, 2014]

(b) "Blond" object from the Poilu collection.

(c) Albert Einstein "hairy" model designed by Print That Thing (https://cults3d.com/en/3d-printing/hairy-einstein).

Figure 39: Example of aesthetic 'curved' prints.

Lia et al. explored the parameter space of 3D printers to create a new kind of sculpture, using all sorts of things previously seen as defects and constraints into hypnotic outcomes [Lia, 2014]. These sculptures are mostly made of incomplete bridges and stringing (Figure 39a). By doing incomplete bridges, the material falls more or less, depending on the size of the bridge. The POILU collection explores the possibility of 3D printing to generate shapes

and to create and implant "hairs" [bold design, 2016]. It is now fairly common to have hairy prints (such as hairy lion and hairy Einstein, Figure 39c).

5.3 Experimental approaches and partially curved layers

Only a few methods focus on problems related explicitly to FDM, and the case of curved printing is not an exception, with most works exclusively focusing on experimental approaches. Many of them concentrate on exhibiting better physical properties by doing tests and comparisons between flat-layer specimens and curved-layer specimens. The first experiments took place in the nineties [Klosterman et al., 1999], with a growing interest in curved printing for mechanical properties. Except for "by-hand" experiments, the first out-of-the-plane algorithms targeted only shell-like parts or other constant layer height prints. Keating and Oxman [2013] introduced an FDM-based proof-of-concept printing, showing how exploiting all 6DoFs of a robotic arm can improve the 3D printing process. However, the demonstration is limited to simple shapes (e.g., cubes, torus, and cylindrical surfaces), and there are no details regarding toolpath generation.

One main advantage of out-of-plane printing is orienting deposition paths to improve the resistance to external forces. Huang and Singamneni [2015] with his Curved Layer Adaptive Slicing algorithm effectively showed that curved layers improve the mechanical properties of a 3D object. They also experimentally demonstrated that the thicker the layer in adaptive slicing, the more resistant to bending stress. It is also true with curved layers, where bending test results exhibit better compressive loads and flexural strengths with thicker curved layer specimens.

In addition, some specific infill structures can improve FDM printed object resistance. For example, auxetic materials show better resistance to bending stress than conventional lattice [Lim, 2015; Yang et al., 2015], and this is even more true when these structures are printed in a curved shape, which allows having much more resilient curved structures [McCaw and Cuan-Urquizo, 2018; Cuan-Urquizo et al., 2019].

5.3.1 Partially curved layers

To better approximate the surface, a variable layer height scheme [Wang and Chen, 2013; Chen et al., 2019; Zhao, Guo and Gao, 2020].

5.4 Fully curved algorithms

Only a few algorithms addressing some problems related to FDM exist and, more generally, in AM with curved printing.

I will first explain the slicing pipeline used for curved printing and then review the different types of objects and technics used to print. For now, existing algorithms try to

address one or multiple specific problems, such as shell-like prints, constant layer offset, supportless printing, or wire meshes.

5.4.1 Curved printing pipeline

Planning the layered process in classical AM can be formulated as a two-step process. The first step is to decompose the volume into layers. Then, a print path is generated to fabricate each of them.

In out-of-plane printing, the same process is mainly used for simplicity as extracting layers and then extracting paths inside those layers is more accessible than directly extracting freeform paths within the volume.

Layer extraction

First, the volume is split into a set of layers. Then, the generation of paths to be followed by the extruder is needed to cover the non-planar surfaces. Diverse approaches have been investigated depending on the objective. The process of filing surfaces in CLFDM demands continuity for different aspects, e.g., trajectory, orientation, and pose [Dai et al., 2018]. Unlike 3-axis FDM, where ensuring trajectory-continuity suffices, in multi-axis CLFDM, the orientation of the nozzle also requires continuity. Finally, continuity is also needed when printing with robotic arms, as it avoids deficient kinematic performance and singularities. All those parameters must be considered when generating the set of layers as they will condition the paths generated by the infills.

Infills for curved layers

Most approaches use a standard toolpath generation within a slice, but the infills are generally the same as in modern slicers. To be more specific, dense infills (section 4.1) are generally used. Zigzag infill is the most used infill either alone [Xie et al., 2020; Singamneni et al., 2012; Chen et al., 2019] or in combination with concentric infill [Isa and Lazoglu, 2019; Zhao et al., 2019; Feng et al., 2021]. Other infills are often used, such as space-filling curves[Zhao, Li, Shen, Jiang, Guo and Gao, 2020] or Fermat-spiral curves [Zhao, Gu, Huang, Garcia, Chen, Tu, Benes, Zhang, Cohen-Or and Chen, 2016; Dai et al., 2018].

5.4.2 Shell-like prints

Shell-like printing takes advantage of curved printing to improve the strength of the printed part and minimize staircase defects [Chakraborty et al., 2008]. For shell-like prints, Diegel et al. [2010] introduced a simple way to generate constant layer offset based on the 3d model normals and pushed it forward to use conductive polymers inside Diegel et al. [2011] and then biopolymers [Brooks et al., 2017]. Using polymers such as conductive ones may require a constant layer height, as the filament conductivity is directly related to its diameter, and variations in height can add bottlenecks.

Figure 40: Chair designed by Zaha Hadid Architects with Nagami where we can see the curved layers.

Thomas et al. [2016] presented another method to print shell-like parts using a Delta-style printer. The shell model is fabricated by depositing a curved layer on top of a proxy structure printed with planar layers. As a result, the surface of the 3D printed model does not exhibit the staircase defect. However, this approach is limited to models with relatively simple shapes (i.e., constant thickness height fields facing the z-axis) or interpolation between two surfaces (like the 3D printed chairs Figure 40). In addition, regions with a steep slope will generate a collision between deposited materials and the printer head. A most recent effort was paid to generate 3-axis motion toolpaths inside a given volume [Ezair et al., 2018], which also suffered from gouging.

Proxys structure or mandrel are often used to speed up the printing process [Thomas et al., 2016; Brooks et al., 2017; Fry et al., 2020]. These structures can be fixed in the case of batches or just be printed before. To reduce the print cost, Hongyao et al. [2018]; Ramírez-Gutiérrez et al. [2020] created a bed composed of movable pins, removing the process of printing such support structures.

5.4.3 Supportless printing

The drawback of having support structures while fabricating an object is an enormous limitation in extrusion-based techniques. First, adding support structures means the machine spends time building those supports. This additional time significantly increases the total build time of the part [Zhao, Li, Shen, Jiang, Guo and Gao, 2020]. Second, the support structures require extra material during the printing process, which increases build costs. Third, the part built with support structures requires additional post-processing to remove it. Finally, the surface finish is badly impacted at the interfaces between the object and the support structure. While support structures solve the problem of printing overhanging features, they also introduce additional costs and complications.

There are two ways in out-of-plane printing to remove the need for support. On the one hand, adjusting the part's orientation as it prints ensures there is always material already deposited below the extruder. On the other hand, solidify the material quickly enough to

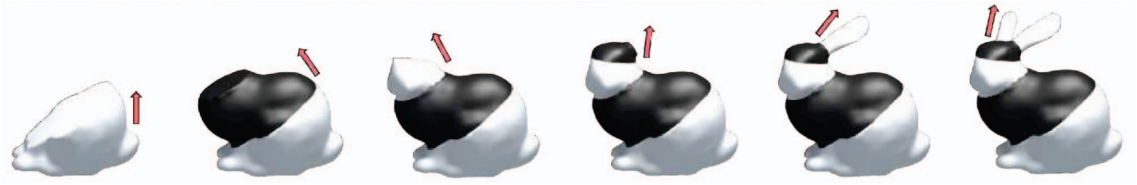


Figure 41: Different parts of the Stanford bunny model are fabricated in different directions (following the red arrows in the illustration), and filaments of different colors are used for making different parts by the RoboFDM system [Wu, Dai, Fang, Liu and Wang, 2017].

counter the influence of gravity (or make sure it is minimal).

Adjusting the orientation

Building among different orientations can be done by rotating the print bed continuously to have previously deposited material below the printing tool head. "RevoMaker" [Gao et al., 2015] tackled this problem with its cuboid print bed to adjust the orientation. This multi-orientation bed leads to basic multi-axis printing but is limited to a few printing directions starting from the bed.

The first algorithm's attempts to print with more orientations were not out-of-plane but somewhat similar to part decomposition, with different build directions for each part. However, instead of printing multiple parts and assembling them afterward, they were directly printed on top of other printed parts, first with DED [Lee and Jee, 2015] and then with FDM [Wu, Dai, Fang, Liu and Wang, 2017] (Figure 41).

Those previous algorithms cannot be considered out-of-plane printing, as all the paths are still generated using planes, so new algorithms using multi-axis motions appeared using the latitude added by these additional degrees of freedom. This additional freedom is used in all current methods tackling the generic problem of free form deposition, e.g., using the 6-DoF of a robotic arm [Dai et al., 2018]. While offering many advantages over traditional flat deposition, these systems require expensive hardware (6-DoF robotic arms or 5-axis motion platforms), limiting their applicability.

One of the first algorithms working for any input shape is support-free volume printing by multi-axis motion [Dai et al., 2018] (see Figure 42). It uses the additional freedom to remove the need for an external support structure altogether. This approach introduced a methodology to compute advancing fields for material accumulation by always performing material deposition along the surfaces of convex hulls. By using convex hulls, the printing process is guaranteed to remain collision-free. The algorithm to generate the toolpaths works as follows: First, it produces a scalar field representing the shape's growth during the fabrication process. After that, it extracts from the field a sequence of curved layers. Lastly, toolpaths are generated to cover each curved layer by incorporating the constraints from hardware systems. As many robot arm poses can print the same segments, an additional optimization pass minimizes the distances between the different positions and ensures no collision will occur.

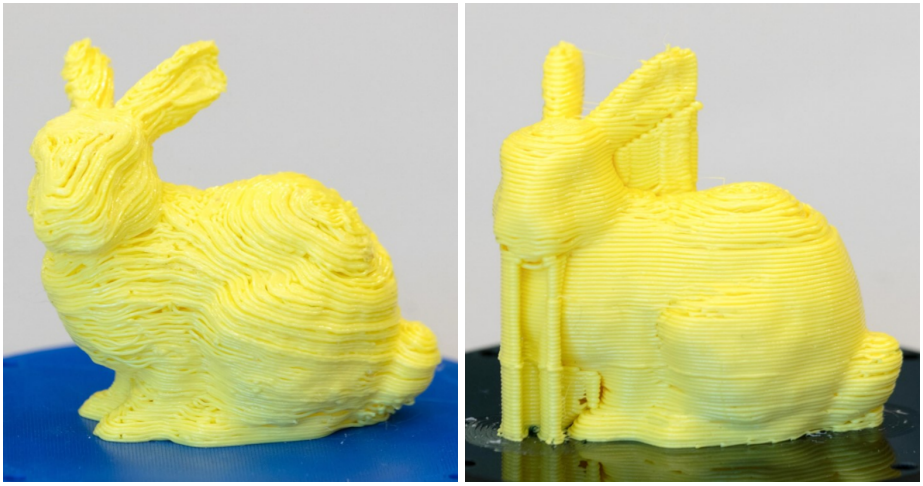


*Figure 42: Comparison between **Left:** Support-free volume printing by multi-axis motion, and **Right:** Standard 3-axis printing on the same setup. [Dai et al., 2018]*

Xu et al. [2019] introduces another way to remove the need for support, ensuring every printed path is supported from a previously deposited one, using a geodesic distance field to extract layers iso-contours in a valid order. The iso-contours establish a smooth transition from the bottom upwards, and being a scalar field gives a natural order of non-intersecting curved layers being generated. However, an iso-contour is only a closed 3D curve, not a layer. The first step in generating a layer is to flatten this contour. Once flattened, the contour is filled with triangles, and, finally, they map back the triangular mesh to a 3D surface based on the least square fitting.

I also used this idea of flattening and mapping back to the original curved shape in CurviSlicer (Chapter 8), but inside the volume using tetrahedrons.

Mid-air printing

The main property that makes support required is that the extruded material is in a viscous state and is impacted by gravity before it hardens. A way to remove the need for support is to remove the impact of gravity without impacting the structural properties, but printing in space Werkheiser et al. [2014] Prater et al. [2016] may be expensive. In that case, hardening the filament faster is a more feasible way to prevent leakage. Special extruders have been designed and assembled to spray water or blow enough air on the printed part to cool down the material before it starts to fall. As a direct applicative approach for mid-air printing, there are wireframe meshes.

5.4.4 Wiremeshes

Instead of printing the entire object volume layer by layer, wireframe meshes are a set of wires extracted from the surface, directly printed in 3D space. This technique performs in-

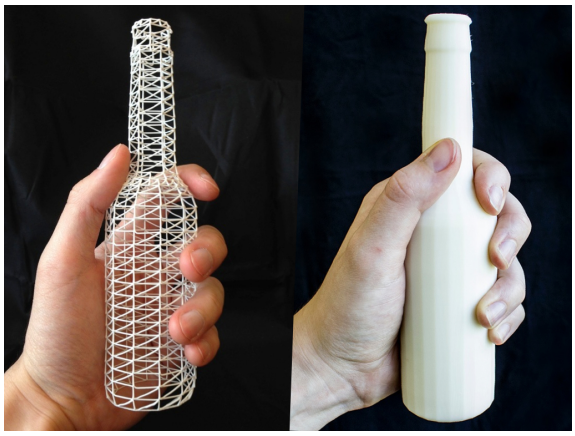


Figure 43: 3D prints of a bottle using WirePrint [Mueller, Im, Gurevich, Teibrich, Pfisterer, Guimbretière and Baudisch, 2014]. This wireframe print (left) is 8 times faster than a standard 3D print (right) on the same 3D printer.

credibly well on FDM printers, as it can print mid-air to create bridges between printed parts.

While standard printing can take hours to days to finish, wire meshes print in low fidelity, requiring less time and less material than standard printing. Wire meshes allow the users to design objects and have a quick overview of the result. It is well suited to have an overview and test the ergonomic fit of a design, where the object can be in-hand in a matter of minutes instead of hours.

Wireprint

The first published approach of wire meshes by Mueller, Im, Gurevich, Teibrich, Pfisterer, Guimbretière and Baudisch [2014] aim to do low-fidelity parts that print faster to do more design iteration before printing the high-fidelity part (example Figure 43). A good analogy is seeing these low-fidelity steps, like sketching and paper prototyping, prioritizing speed over functionality. This trade-off pays off in the early phases of design because it encourages the quick exploration of several versions before committing further resources, eventually leading to a better design.

The extraction of the wire mesh works as follows:
First, the mesh is sliced by a set of parallel planes. The height of each slice is chosen to keep as many details as possible from the 3D model. Then, the contours of each slice are extracted as the only thing needed is the outline. A zigzag pattern is then generated to connect those contours. By averaging the last zigzag slice points and an evenly spaced distribution of points on the contour, they obtained better stability and adaptability to changes in layers' height and contour length. The last diagonal is removed to be printable because it cannot be printed without colliding previously printed zigzags. Finally, the GCode is generated, and then, using a standard delta(3 DoF) printer with enhanced cooling abilities, the part is printed.

In order to test the ergonomics of a piece, the contact surface must be solid, i.e., the wireframe structure should not alter the texture too much. To do this, they used two techniques. The first consists of adding a step during which the object is dipped in glue to fill the wireframe's holes. The second is to modify the algorithm to add filled parts, where the surface is partially printed instead of having the wireframe.

Untzelmann [2018] also prints parallelly to the bed but optimize the spacing between the vertical wires to keep a near-constant wire density. The printing time is more significant than WirePrint, and they acknowledge it was mainly due to their cooling system.

On-the-Fly: Incremental Printing While Modelling

On-the-Fly is a direct improvement of WirePrint where design and print can be done in parallel[Peng et al., 2016]. The new features are 5DoF hardware, an enhanced cooling system, and a cutter to remove unwanted parts.

The 5DoF hardware extends a standard delta printer with an additional 2 DoF rotating bed. The cooling system is no more an air-cooling system but a water-cooling system with atomizing nozzle flanking the extruder. This enhancement led to a five-time faster print for the same size cell. The last hardware improvement is the cutter that removes parts during interactive design to correct mistakes, adjust geometry and perform subtractive operations. To be able to print and remove material, the software is made of two main operations, additive and subtractive.

Additive operations are executed when a feature is ready to be printed. This new feature is considered for printing if it has not been modified for 5 seconds and either touches the printing platform or a previously printed part. Once a new feature is ready to be printed, the resulting G-code is sent to the printer, which sends back an acknowledgment upon completion. As the printed wireframe leaves significant gaps in the object's external surface, printing another shape directly onto the surface may be impossible. To ensure the new geometry can be created at the location of a gap, they patch the surface, adding material where the part will lay.

Like in additive operations, if a cut feature is smaller than the size of the printed mesh cells, a patch is created to cut only the desired surface. In order to cut the printed cells, a subtractive operation must be executed. Subtractive operations work like additive ones. Once the user has created the cutting geometry, and no more modification is done, the intersection curve between the original object and the cutting geometry is computed. This curve is then used to drive the cutting blade. Once the cut is completed, the print head follows the same contour to redraw the edge of the cut.

By adding parts over parts, the object may become unprintable. On the first hand, if a new feature is not printable because a pending print part will generate collision, the print process is reordered. On the other hand, if the part is already printed, a best-effort policy will try to print as much as possible but highlight the designed parts that cannot be built.

Printing Arbitrary Meshes

On-the-Fly authors pursued their goals to print wire meshes and published the same year printing arbitrary meshes [Wu, Peng, Guimbretière and Marschner, 2016]. This enhancement addresses the problem of more complex shapes by generating a collision graph that encodes all the constraints making the object not printable. To do so, they encode the relationships between printing order, printing orientation, and collision. There are still cases where the algorithm fails, like cyclically dependent edges or when the peeling algorithm cannot find any traversal order.

Part III

Adaptive infills

Contents

Introduction **73**

6 Procedural band patterns **75**
6.1 Method . . . 76
6.1.1 Overview . . . 76
6.1.2 Procedural bands . . . 76
6.2 Controls and parameters . . . 79
6.3 Pseudo-code . . . 80
6.4 Results . . . 83
6.4.1 Bands and curves . . . 83
6.4.2 Animated patterns . . . 83
6.4.3 Infill patterns for 3D printing. . . . 84
6.5 Limitations . . . 88
6.6 Conclusion . . . 88

7 A brick in the wall: Staggered orientable infills for additive manufacturing **91**
7.1 Background on phasor noise . . . 92
7.2 Our approach . . . 93
7.2.1 Defining trajectories . . . 93
7.2.2 Periodic variable fields . . . 95
7.2.3 Graph extraction . . . 95
7.2.4 Clipping and ordering trajectories . . . 96
7.2.5 Staggered bead layout . . . 97
7.3 Results . . . 97
7.3.1 Comparison to curve hierarchies . . . 98
7.3.2 Singularities, staggered layout . . . 98
7.4 Discussion and conclusion . . . 104

Conclusion **107**

" *The important thing is to never stop questioning [or learning].* "

Albert Einstein

Introduction

As seen in the previous chapter, additive manufacturing produces the part layer by layer, gradually building up the object. In all deposition technologies, paths along which the material is deposited or solidified directly influence the properties of the final part. The printing trajectories thus play a crucial role in the structural soundness of the part, and special attention must be taken to minimize potential defects. As a result, controlling the deposition paths allows for outstanding control over how the volume is formed, providing new and unique opportunities to tailor a part to its final functionality.

Often, the deposition strategies have to enforce competing objectives. For instance, the process may be expected to produce dense parts with minimal porosity or the lightest part that can sustain a given load. It is then desirable for the trajectories to follow specific directions within layers due to manufacturing constraints or to maximize part functionality, e.g., stiffness under some external loads. In particular, additively manufactured parts tend to be stronger in the direction of deposition, while in the orthogonal direction, the bond between neighboring beads is less strong [Rybachuk et al., 2017]. In addition, some processes can deposit anisotropic materials (such as fibers) alongside the trajectories, resulting in even higher ratios of anisotropic mechanical response. The possibility of conforming the deposition trajectories to specific directions thus opens the way to several applications: more resistant parts [Steuben et al., 2016], controlled deformations [An et al., 2018], as well as the ability to pattern the surface roughness visually(light reflection) or physicaly (grip, Figure 44).

This chapter proposes two novel approaches to generate trajectories following an input direction field. The first approach, "Procedural band patterns" [p.75], allows the user to control the density inside the part guided with a density field and an orientation field. The second approach, "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91], aims to fill the part as much as possible while conforming to a direction field.

Figure 44: A screwdriver handle with enhanced grip.

Chapter 6

Procedural band patterns

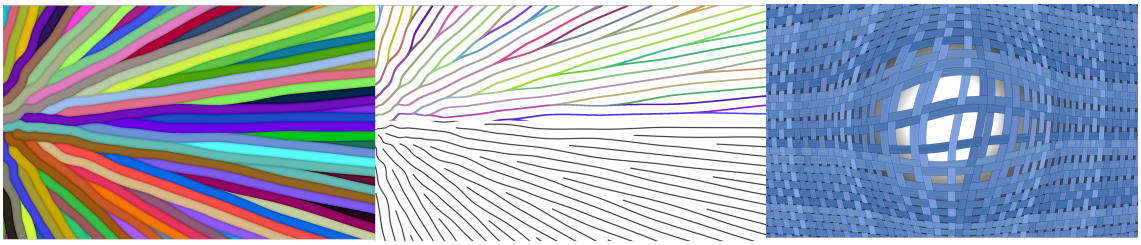


*Figure 45: Our technique generates band patterns following a parametric field, while adapting the density of bands per unit. Each band is uniquely identified (*left*) which affords for robust extraction of border trajectories (*middle colored top*) and center lines (*middle bottom*). The method is procedural which allows a wide range of dynamic shader effects, such as textile patterns that adapts to stretch (*right*). Contrary to classical subdivision approaches, our approach introduces new bands with a non power of two factor, allowing a more progressive gradation.*

The original purpose of this work was to create an infill pattern for 3D printing that was able to follow a density field. It was initially designed and shipped in IceSL as "Progressive infill", empowering the user to create variable density prints across layers. For debugging purposes, each strip received a different color, and after seeing the kind of pattern it was able to produce, the procedural method behind the infiller was used for strips generation inside a rendering shader.

This shader produces parallel band patterns along a given parametric field while locally adapting the number of bands to an input control field. More precisely, we expect as input a domain $\Omega \in \mathbb{R}^2$ (or $\Omega \in \mathbb{R}^3$), a parameter field $u : \Omega \to \mathbb{R}$, and a density field $d : \Omega \to \mathbb{R}$ which defines a target number of bands per unit of u.

These fields may be the direct result of an optimization [Pantz and Trabelsi, 2008; Groen et al., 2019] or may be painted by the user. The field d may be related to u (e.g., to compensate for distortion) or may be independent of it, in which case the actual local density of the bands will be a combination of the intrinsic stretch of u and the density factor of d. Both u and d are assumed to be smooth and continuous scalar fields.

We denote by p a point in Ω, such that we obtain the parameter value $u(p)$ and density value $d(p)$ at each point p. The objective is to produce a set of bands flowing along u iso-values where the local density of bands (or equivalently their spacing) changes according to d.

The originality of our approach is to define a lookup function $\mathcal{B}(v, s) : \mathbb{R} \times \mathbb{R} \to \mathbb{N}$ that returns the unique integer ID identifying the band enclosing parameter value v, where the local density is controlled by s. In our context the lookup is performed for a point $p \in \Omega$ as $\mathcal{B}(u(p), d(p))$. $\mathcal{B}$ is *procedural* [Lagae et al., 2010]: it has a minimal computational and memory requirement, allowing its implementation in a pixel shader for fast synthesis and interactive manipulation and animation of complex patterns.

The returned integers uniquely identify the bands allowing for a wide range of texturing effects.

6.1 Method

Given the fields d and u, we cover the domain Ω with a discrete grid and synthesize the bands by evaluating $\mathcal{B}$ on every node. This technique is implemented as a pixel shader in OpenGL Shading Language (GLSL) applied to a render target, and the code is given in section 6.3. The remainder of the text progressively introduces the principles and elements of the algorithm.

6.1.1 Overview

Given a spacing s, it is trivial to produce parallel bands in $u(p)$ by defining the band identifier as $\lfloor \frac{u(p)}{s} \rfloor$. The principle of our approach is to cover the domain with many overlapping sets of parallel bands, using different spacings. The spacings are decreasing powers of a base value $1 < step \leq 2$. We call *bands of level* L the set using a spacing of $step^{-L}$. The choice of $step$ allows a trade-off between a doubling split ($step = 2$) or a more progressive gradation, which we discuss in section 6.2.

We select which set should appear in a given location following $d(p)$ and interpolate (deform) the bands, with smaller bands' borders joining or closing onto coarser band borders. This junction of the fine and coarse band, in turn, defines a parent-child relationship which we use to define a global numbering through the band hierarchy.

6.1.2 Procedural bands

Given a point p, we seek to compute the global identifier of the band enclosing p. This computation is done in two steps. First we identify the local ID of the band to which p belongs at the density $d(p)$. The local ID is the rank of the band in its own set, it is computed by the function `getLocalID` (see code section 6.3). Second, we produce a global ID for the band in the function `getGlobalID`.

We now focus on `getLocalID` (line 13). From $d(p)$, we first determine the band levels bracketing the target density. This is achieved by quantizing $d(p)$ based on the value of $step$, see the `quantize` function line 5. This quantization gives the situation illustrated in Figure 46: p belongs to two bands, one in a set having more bands than the target $d(p)$, and one in a set having fewer bands.

We then define an interpolation that displaces the band borders, making the finer bands deform towards their coarser counterparts. This displacement is illustrated in Figure 47.

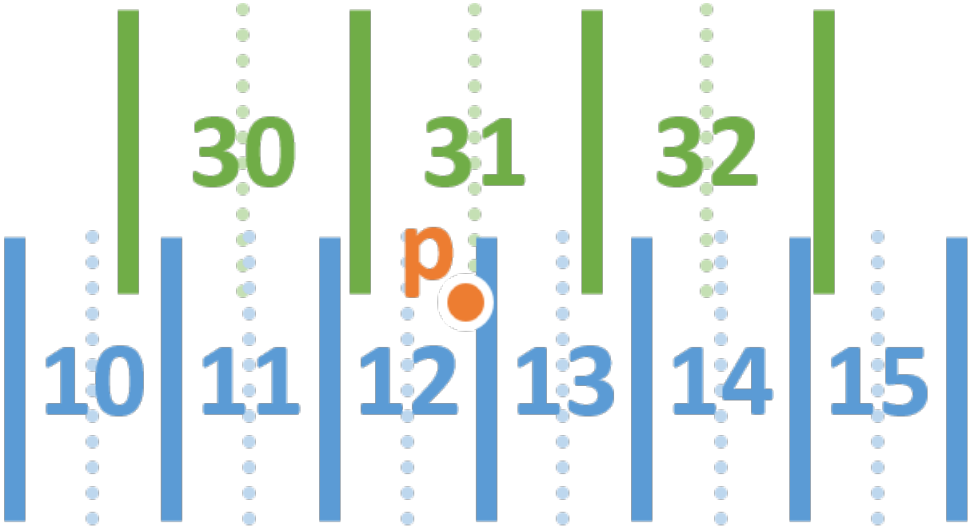


Figure 46: Every point p belongs to a band in a finer and a coarser set enclosing the target density $d(p)$: the finer set has more bands than desired at $d(p)$ while the coarser set has fewer bands.

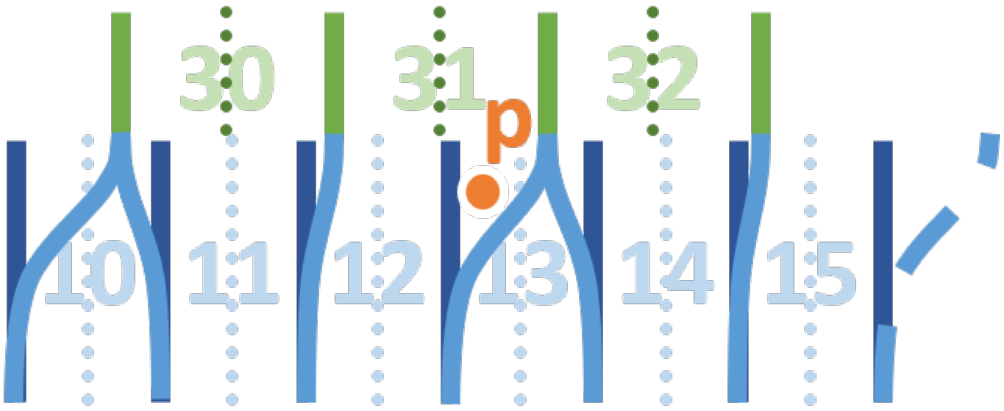


Figure 47: The borders of the finer bands are pulled towards their closest coarser band border. A point p in a transition area may end up in a different band after interpolation.

The interpolation is performed by pulling each finer band border towards its closest coarser band border. The interpolator is given by the position of $d(p)$ within the density interval of the bracketing levels. Note that as we change the location of the band boundaries, p may no longer be in the same initial band. This is tested and adjusted for lines 39 and 42. Through interpolation, some finer bands are *closed*: both their borders move towards the same coarser band border, e.g., band 13 in Figure 47.

After this process, we obtain the band's ID enclosing p in the level just finer than $d(p)$. This ID is local to this particular level: if used directly, it would change arbitrarily where a level change occurs in d. Instead, we produce a global ID, valid *across* levels. This is achieved in `getGlobalID` (line 51). Note that since we have $step \leq 2$, we at most double the number of bands from one level to the next. Thus, our approach is to number the bands the same way we number leaves in a binary tree. From the current band level, we move up

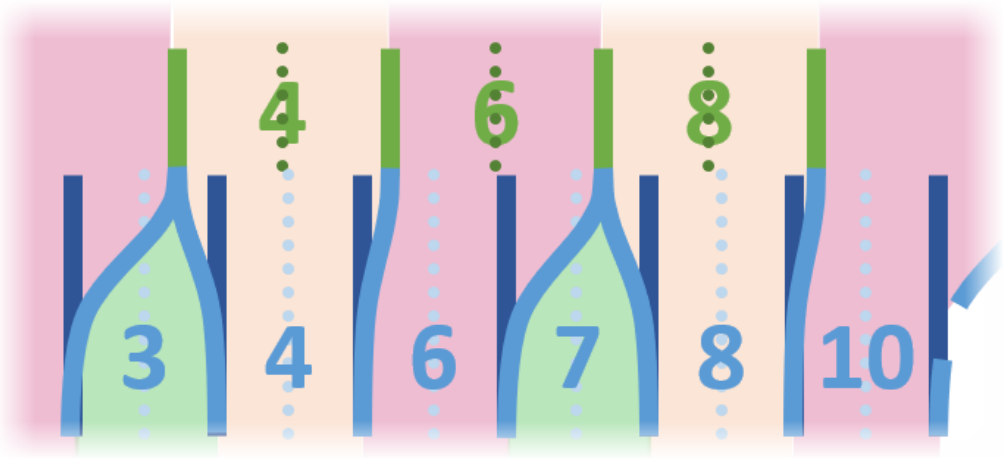


Figure 48: Band global IDs are computed in a hierarchical fashion, with parent defining most significant bits while children use least significant ones. The ID of closing bands no longer appears in lower levels, leaving a gap between IDs.

the hierarchy (lines 55–84), and insert a bit in the ID (line 71) each time the band closes (lines 60–69). Finally, the ID of the top-level band is added as the most significant bits of the global ID (line 87) and illustrated in Figure 48. In other words, the global ID is the ID of the top-level band, followed by the binary pattern of opening bands to reach the band enclosing p.

As an added bonus, we can easily detect when a band *just appears*, by checking if the first band is closing immediately (lines 72–79). This ability to detect appearing bands is an exciting capability. As such band (the appearing one) is smaller than the target spacing – there is not yet enough space between the parents for it to be fully deployed. In particular, we may choose to remove such a band or draw it in a different style.

Band shifting. Using an irregular subdivision pattern ($step < 2$) poses an exciting challenge. Around the origin, all band sets align, producing an undesirable pattern shown in Figure 49 left. This alignment can be suppressed by translating the band sets by a (pseudo-)random shift, which is different for each band level. The effect is shown in Figure 49, right.

For $step = 2$, we shift the bands at every level by half their spacing to obtain the traditional balanced subdivision pattern.

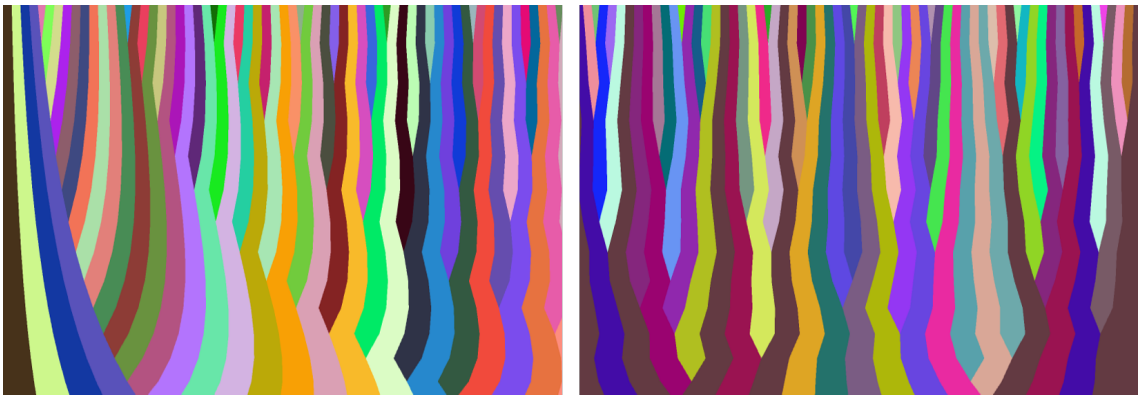


Figure 49: ***Left:*** *Without shifts, the alignment of all band sets at the origin produces an artificial, peculiar pattern.* ***Right:*** *The random shifts break the alignment, making the pattern similar everywhere.*

6.2 Controls and parameters

Progressive increase in density. By changing $step$, we control the number of bands opening at every level. For $step = 2$, our technique behaves similarly to traditional subdivision approaches, with one new band opening for each coarser band, effectively doubling the number of bands at every level.

It is worth outlining that this approach, while very typical, also gives only a crude approximation of the target density. Indeed, across an interpolation range, the produced density (bands' number) is wrong by as much as 50% (in the middle).

This technique affords a very progressive insertion of bands at every new level by using lower values of $step$. Let us consider a value $step = \frac{N}{M}$ with $M, N \in \mathbb{N}$ forming an irreducible fraction, and $M < N \leq 2M$. Such a step will open $N - M$ new bands for every M bands in the previous level – this stems from the fact that each level L corresponds to a density $step^L$.

$N = 2,\ M = 1$ gives the standard case (100% increase every level). $N = 3,\ M = 2$ opens one new band every two coarser bands (50% increase). $N = 17,\ M = 13$ opens four new bands every thirteen coarser ones (31% increase). These numbers can be freely chosen as long as $step$ remains in the $]1, 2]$ interval.

Regularity and periodicity. The produced pattern is entirely defined by the subdivision of the top-level bands (for $d = step$, we have one band per unit in u). It is intriguing to consider the periodicity of the split pattern. For the sake of clarity, we ignore the random shifts in this discussion and assume all band sets are aligned on the origin.

Let us again write $step$ as an irreducible fraction $\frac{N}{M}$. From one level to the next, a period occurs in the split pattern every M bands, as borders across both levels align: this repeats the pattern at the origin. However, this is the case for *two consecutive* levels. When we consider more levels, the period before all borders align grows. In fact, for k levels, the period becomes M^k. Indeed, given Q bands at the base level, the number of bands at level k is $Q\left(\frac{N}{M}\right)^k$. This number will only be an integer value for $Q = M^k$ since M and N are

coprime (irreducible fraction).

Interpolation profile. All our results use a simple linear interpolation. However, using a different profile (e.g., GLSL smoothstep) leads to different (smoother) band shapes in the opening region. In addition, the interpolation profile can control how 'quickly' new bands open to their full width.

6.3 Pseudo-code

The detailed pseudo-code is given next. We also refer the reader to our shadertoy implementations (see Table 6.1).

```
/**
 * @param d density
 * @return the density's quantization, based on the global parameter step
 */
float quantize(float d) {
    return pow(step, floor(log(d)/log(step)));
}
```

```
8  /**
9   * @param u orientation
10  * @param d density
11  * @return the local id of a pixel after interpolation
12  */
13 int getLocalID(float u, float d) {
14
15     int level = density2level(d);
16
17     // fine and coarse densities
18     float df  = level2density(level  );
19     float dc  = level2density(level-1);
20
21     // get enclosing band id
22     int local_id   = u2id(u, df);
23
24     // get the borders of the finer band
25     float lf_border = id2Leftborder(local_id    , df);
26     float rf_border = id2Leftborder(local_id + 1, df);
27
28     // get the closest coarser band borders
29     float lc_border = u2closestBorder(lf_brdr, dc);
30     float rc_border = u2closestBorder(rf_brdr, dc);
31
32     // interpolate between fine and coarse borders
33     float a     = interpolate(d);
34     float left  = lf_border * a + lc_border * (1.0 - a);
35     float right = rf_border * a + rc_border * (1.0 - a);
36
37     // adjust enclosing band after interpolation
38     if (u < left) {
39         local_id--;
40     }
41     if (u > right) {
42         local_id++;
43     }
44     return local_id;
45 }
```

```
46 /**
47  * @param local_id the local id
48  * @param d density of the local id
49  * @return the global id of a local id from its density
50  */
51 uvec4 getGlobalID(int local_id, float d) {
52     uvec4 global_id = uvec4(0u);
53     int start_level = d2level(d);
54     // Iterate over all levels until it reaches the root
55     for (int level = start_level; level > 0; level--) {
56         // get the density of the fine and coarse bands
57         float df      = level2density(level);
58         float dc      = level2density(level - 1);
59 
60         // get the borders of the fine band
61         float lf_border = id2Leftborder(local_id    , df);
62         float rf_border = id2Leftborder(local_id + 1, df);
63 
64         // get the closest coarser band borders
65         float lc_border = u2closestBorder(lf_border, dc);
66         float rc_border = u2closestBorder(rf_border, dc);
67 
68         // if this band is closing between parents
69         if (lc_border == rc_border) {
70             // Update the global id (bit shifting)
71             global_id = setIDBit(global_id, uint(96 - level));
72             local_id  = u2id(lf_border, dc);
73 
74             // if the start band is appearing
75             if (level == start_level) {
76                 /*******************************************/
77                 /* add code here to modify the split behavior */
78                 /*******************************************/
79             }
80         } else {
81             // move up to parent
82             local_id = lc_border;
83         }
84     }
85 
86     // parent ids into most significant 32 bits
87     global_id = setMostSignificantBits(global_id,uint(local_id));
88 
89     return global_id;
90 }
```

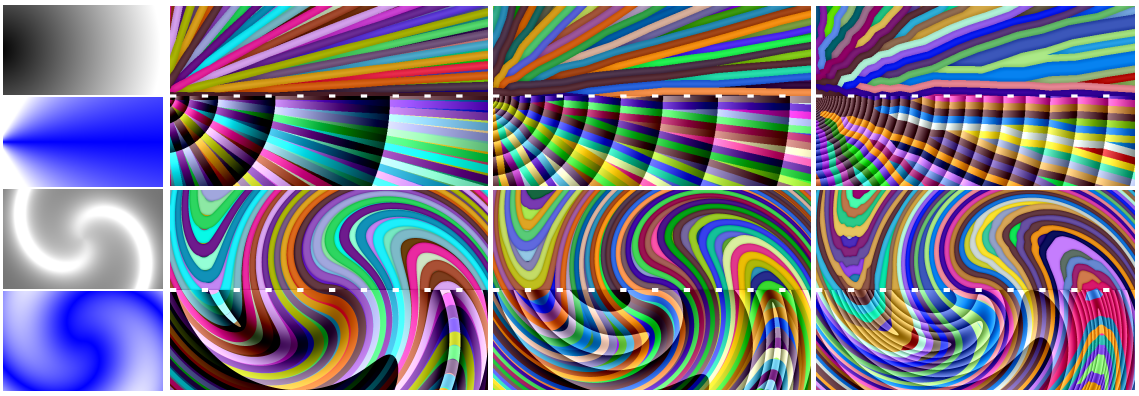


*Figure 50: Two results using complex u and d fields, using $step = 2, \frac{17}{13}, \frac{79}{71}$. **Left:** u field in blue-white, d in shades of gray. **Below dashed line:** discretization steps of d overlaid, the more steps the more precise is the fit between the actual and the target number of bands.*

6.4 Results

Our technique opens multiple possibilities to produce band and curve patterns. In this section, we first show some basic patterns obtained directly from our technique and discuss the effect of parameters in subsection 6.4.1. We then discuss how to use various properties to produce more elaborate animated patterns in subsection 6.4.2. Finally, we discuss how trajectory extraction can be used to produce graded fill patterns for additive manufacturing in subsection 6.4.3.

6.4.1 Bands and curves

Figure 50 shows two complex cases of fields u and d with bands produced for various values of $step$. The left most is $step = 2$ while the others are showing decreasing values ($step = \frac{17}{13}, step = \frac{79}{71}$). The parts below the dashed lines show the interpolation regions (discretization of d). Note how as $step$ decreases, the discretization of d refines, producing a better match to the input field. For small values of $step$, the distribution of splits becomes less regular, but the bands are also more 'jaggy'. A suitable tradeoff is easily found by interactive manipulation.

Figure 45 shows three drawing modes enabled by our approach. The first is to extract a network of splitting/merging curves from the boundaries of the band regions. The band IDs allow deciding which pixel edges belong to a curve robustly and provide a unique ID for each extracted trajectory (the pair of band IDs on either side). The second mode is to draw the middle lines of fully deployed bands (see test lines 72–79) to obtain evenly-spaced streamlines. The third is to draw bands directly, with a subtle shadowing effect. Much more patterning effects can be achieved, as we will discuss next.

6.4.2 Animated patterns

Note: These effects are best seen animated. We provide the shadertoy links at the end of this section (Table 6.1).

We provide examples of animated effects, exploiting several unique possibilities. We generate the effects by having procedural fields u and d dynamically drive one or multiple band patterns. The u field is obtained analytically (sum of sines, radial basis functions, and noise), while d is often computed from the gradient of u to compensate for the distortion. We then use band IDs and compute a local parameterization (distance of lookup point to enclosing band's left border) to achieve various effects.

Mesmerizing effects are obtained by coloring the bands in a stretching field with distortion compensation; see, for instance, *flow bands*. Driving the band density also produces exciting results, such as in *scaling bands* where the band density is increased in a moving circular area. The *band flag* effect combines two sets of adapting bands with a coloring that retains consistency along each direction, producing a typical towel coloring scheme.

We also explored tearing animations, where we discard all lines appearing in between the main parent lines. This discard step is easily done by checking the least significant bits of the band IDs and produces an effect where the main bands maintain their width while spaces appear in-between. The parameter $step$ then controls how and whether the bands form groups under stretch. The shaders *tear twist, lens, light claws, textile* and *net* are obtained in this manner. Instead of discarding, we can also change the coloring of the bands, producing effects such as the *hairdryer* shader. Here, new darker hair strands appear as the motion expands the textured area.

The combination of two sets of adapting bands and tearing leads to effect producing squares, for instance, in *lens* and *light claws*. The *lens* effect is a lens where instead of zooming, the squares maintain their size, and the bands tear to accommodate for the distortion. Note the uneven split pattern thanks to a choice of $step < 2$. The *light claws* effect drives a distortion field from sound, producing dynamically splitting bands along lines in two directions, disconnecting squares.

We can also weave two orthogonal sets of bands to produce textiles, for instance, in *net* and *textile*. We determine which band is front at crossings using binary checks on their IDs. We also have the ability to compute a local parameterization within the bands' width to make them thinner. Note how the weaving expands and slides under distortion while the bands maintain their width, revealing the background.

All these shaders are dynamic and real-time – thus, they would work in a rendering context to adapt a texture to stretch. Our technique only requires a single input coordinate (u), and it can also be applied to solids and implicit surfaces.

6.4.3 Infill patterns for 3D printing.

We use our technique to produce filling patterns inspired by so-called cubic infills [Lefebvre, 2015; Wu, Wang, Zhang and Westermann, 2016]. We extract the trajectories as the contours between pixels associated with different band IDs. This process is fast and robust – akin to extracting the contours of a white region in a binary image. The essential advantage is that our technique allows for spatial density grading while producing very continuous paths. One efficient use case is to harden the surface while hollowing in depths by grading the density depending on the distance to the surface, see Figure 51. This whole algorithm

Flow bands	https://www.shadertoy.com/view/wlt3DM
Scaling bands	https://www.shadertoy.com/view/tlt3DM
Tear twist	https://www.shadertoy.com/view/WttGD8
Band flag	https://www.shadertoy.com/view/3ttGD8
Textile	https://www.shadertoy.com/view/wl3GDH
Net	https://www.shadertoy.com/view/3ldGD8
Lens	https://www.shadertoy.com/view/WltGD8
Light claws	https://www.shadertoy.com/view/tt33W7
Hairdryer	https://www.shadertoy.com/view/3tdGW8

Table 6.1: Links to shaders.

has been implemented (and shipped to users) in the slicer IceSL [2022] for more than four years now and can be easily tested.

There are other contexts in additive manufacturing where isolines of equal spacing are desirable, for instance, to generate fill patterns following optimized fields [Steuben et al., 2016] or for curved 3D printing [Ezair et al., 2018]. We believe our approach to be especially promising to decompose a 3D shape into solid slabs before filling them with curved paths (using contouring and zigzag fill within the extracted curved slab [Chakraborty et al., 2008]), see Chapter "Curved printing" Chapter 9, "Another brick in the wall: Orientable trajectories for additive manufacturing" [p.139].

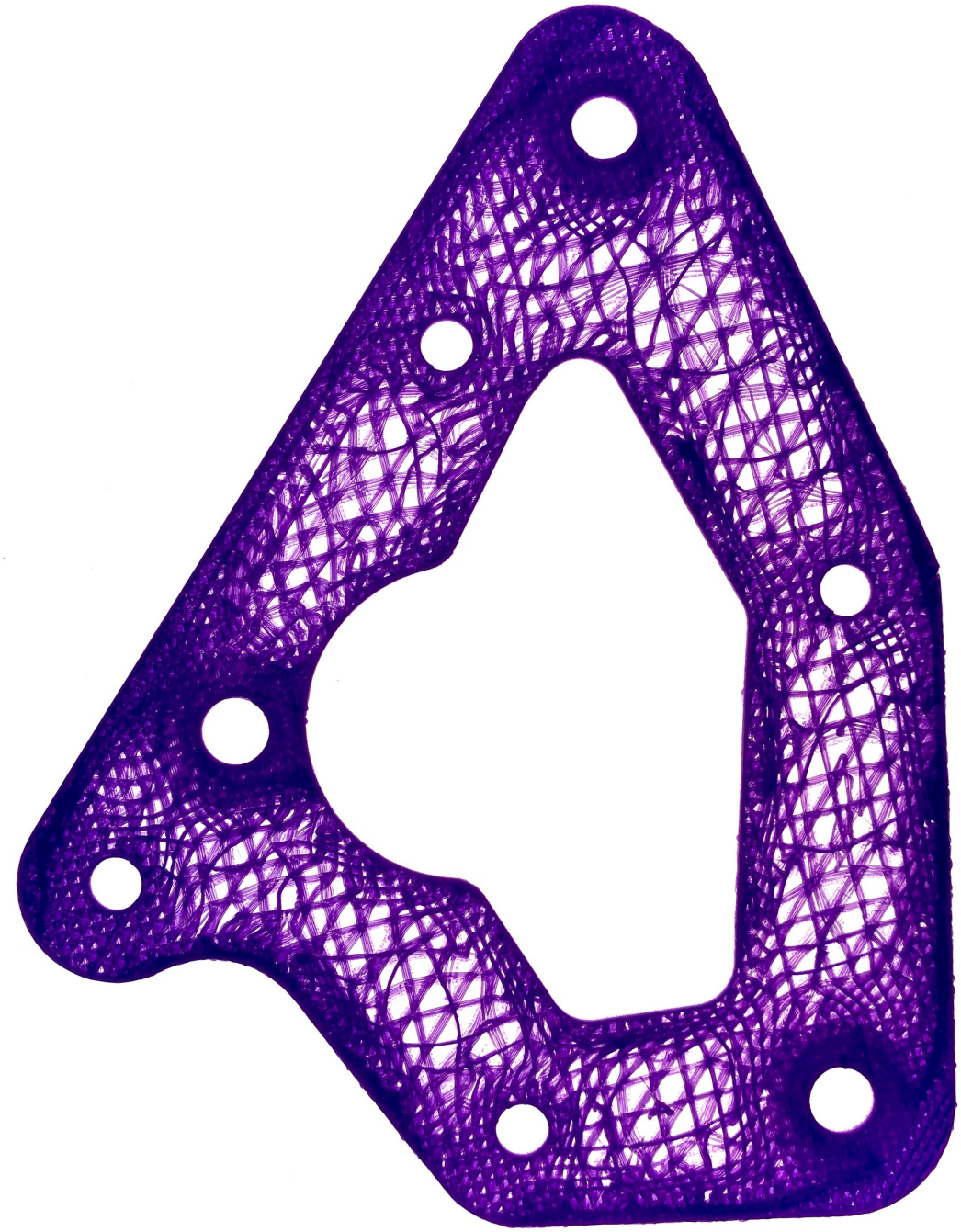


Figure 51: 3D printed part revealing an infill pattern obtained from our technique. Note how the density changes depending on the distance to the object's border. This behavior strenghten the final part while reducing its overall weight.

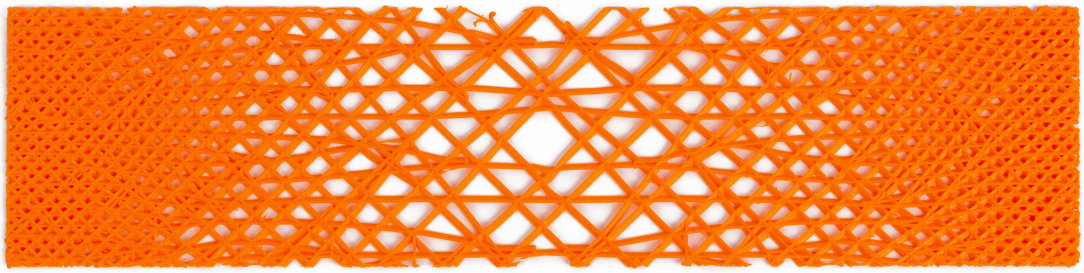


Figure 52: 3D printed part revealing an infill pattern obtained from our technique. Note the very progressive change in density and the curve network extracted at each layer.

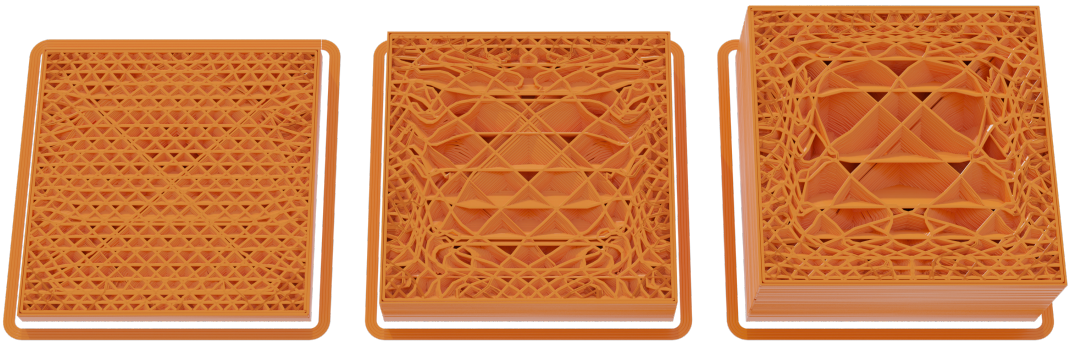


Figure 53: GCode of a cube with density relative to the border distance showing density changes in all axes. It has a high density variation (50% to 5%) in a restricted space, leading to some abrupt direction changes. ***Left:*** *5% printed,* ***Midle:*** *25% printed,* ***Right:*** *50% printed.*

6.5 Limitations

Our technique has some limitations that can be problematic depending on the intended use.

In areas where the density field d varies quickly, bands will appear distorted. This distortion is, for instance, visible in Figure 50, bottom right, and Figure 53, in areas of high curvature. It would be interesting to apply filtering to d to limit such defects. Nevertheless, when using this technique for additive manufacturing, abrupt changes are still printable, see Figure 53.

Intermediate density levels always exhibit partially deployed bands. A consequence is that, while the number of bands remains correct, their spacing may look uneven in areas of constant scale (linear u and constant d). This situation may be minimized – albeit not entirely resolved – by modifying the interpolation profile to quickly open/close bands (see section 6.2).

We believe it would be desirable in the general case to randomize the locations where the band splits occur. While our irregular splitting strategy ($step < 2$) reduces split alignments, they remain located along isolines of d and are not randomly distributed.

The numbering scheme (`getGlobalID`) relies on integers and might run out of numbers, being unable to distinguish between bands deep in the hierarchy. Using wider integers is not too difficult in our context as the only operation is to set bits in the IDs. Nevertheless, infinite zooming would require recycling IDs.

Another problem of this technique is related to the needs of a parametrisation. We tackled this problem in the next section, "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91] by extracting paths from an osciliating signal in space using a phasor noise.

6.6 Conclusion

We propose a fully procedural technique to generate band patterns. It is implemented as a fast pixel shader that interactively reacts to changes to the control fields. We believe our technique is helpful for a wide range of applications, such as visualization, texturing, and producing fill patterns in additive manufacturing.

Our technique extends trivially to 3D, for instance, allowing the decomposition of a shape into solid slabs – the equivalent of our bands – while retaining all our technique's procedural and adaptive capabilities. We envision direct applications in modeling solid properties and process planning for additive manufacturing.

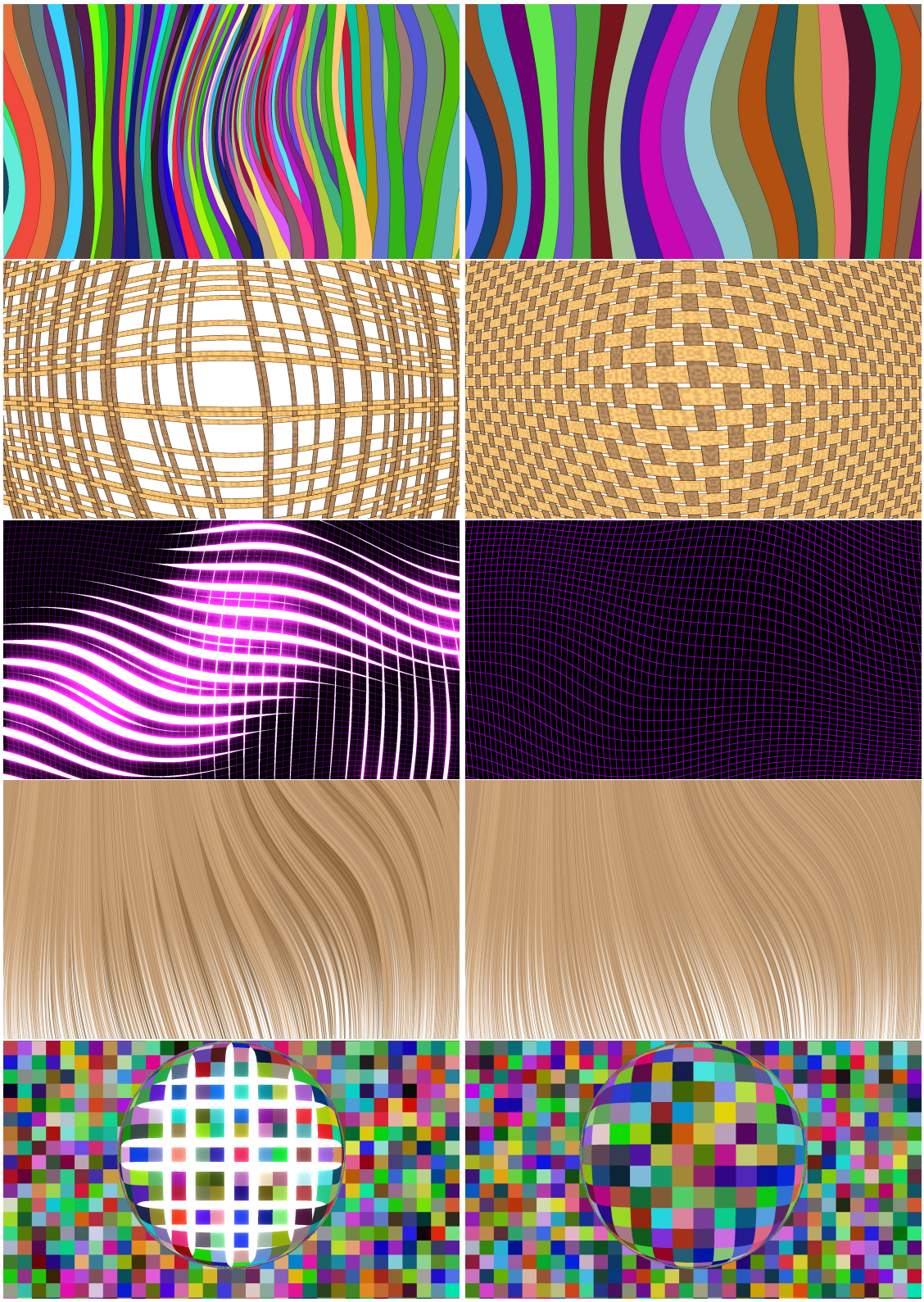


Figure 54: Various animated effects. Each pair shows the effect with our technique with and without enabling the field d (without the bands do not adapt). Please refer to text for details and Table 6.1 for links to shadertoy implementations.

Chapter 7

A brick in the wall: Staggered orientable infills for additive manufacturing

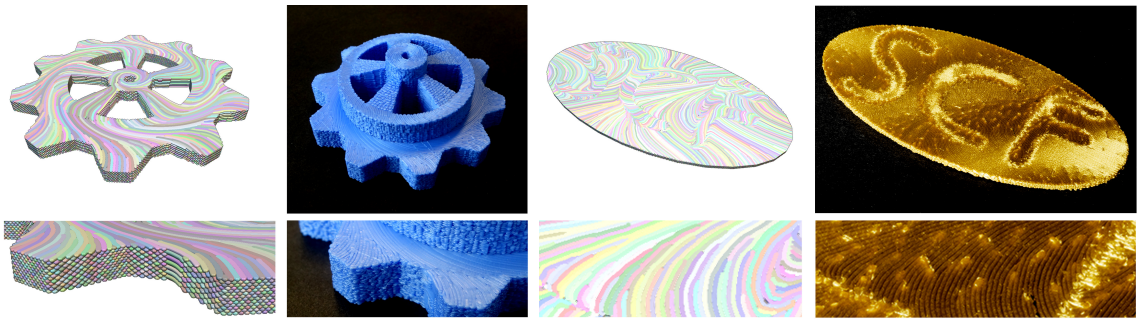


Figure 55: Our technique generates dense planar infill trajectories, precisely following an input direction field. Through the anisotropy of the deposition process, the direction field controls the appearance and the physical properties of the 3D printed object. The trajectories are arranged in a staggered layout across layers. ***Left:*** *A 3D printed gear where the direction of the trajectories is parameterized to adapt to the functionality of the part. Here, trajectories are mostly circular in the rim and hub while being aligned with the spokes. Note the staggered layout in the side views.* ***Right:*** *The anisotropy of the deposition results in anisotropy in the specular reflectance. Here, this is controlled to pattern the appearance of an otherwise flat part, resulting in a brushed metal effect.*

We have seen in the former method a way to cover a layer with paths by controlling the orientation and density through two input fields. This method was using a parametrisation in the space, and needed to correlate the direction field and the density field to generate those paths. We want to improve this technique, but without relying on a parametrization or a density field. Our objective can be rewritten as follow: We want to produce dense parts (removing the density field), using a phasor noise [Tricard et al., 2019](an osciliating signal in space, removing the parametrization).

As we still want to control the direction, we have to conciliate the contradictory objectives of fully covering an area with evenly spaced material beads while having them follow an arbitrary direction. We also seek a result where defects such as gaps have a minimal impact within a layer and across layers, as it weakens the printed object.

Even in the absence of a direction field to follow, avoiding defects in infills is a problem that remains a focus of research (see Chapter "Related work" Chapter 4, "Infills" [p.45]). Adding the direction objective makes this problem significantly harder, as following the

direction requires paths to move away from one another or to regroup in specific locations, creating branching patterns where tiny gaps are difficult to avoid. Due to spatial coherence of trajectories geometry across layers, such minor infill defects within a layer tend to accumulate *across layers* and degenerate into critical defects: internal tunnel-shaped voids that endanger the structural integrity of the part. This problem is worsened when following a given direction field that is similar from one layer to another.

Our technique generates trajectories for planar layers. However, to address the aforementioned challenges, we generate infill trajectories that not only follow the given directions but also form a coherent structure *across* layers. In particular, the deposited beads form a staggered structure akin to the layout of a brick wall. As with bricks and mortar [Nicholson, 1823], the intent is to prevent the weaker lateral bond between beads from forming vertical sheets prone to failure. This staggering also prevents tiny gaps in split/merge locations from forming coherent structures across layers within the volume. Please note, however, that in this work, we focus solely on the modeling aspects of a staggered infill. Measuring to what extent this truly benefits mechanical properties will require an in-depth mechanical study left as future work.

We achieve a staggered layout by building upon the *regularized phasor noise* technique introduced by Tricard et al. for microstructure design [Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre, 2020]. This efficient and scalable approach generates oscillating fields within volumes whose direction can be controlled. We define and extract our trajectories from crossed phasor fields (subsection 7.2.1, subsection 7.2.2 and subsection 7.2.3), and exploit the exposed argument of the oscillations to obtain a staggered layout (subsection 7.2.5). By controlling the oscillations frequency, we can further adjust the deposition width (subsection 7.2.2).

This results in a simple, robust, and efficient method for generating dense, oriented trajectories. Thanks to the stochastic nature of the underlying synthesis process, the unavoidable tiny gaps in branching locations are evenly distributed within the volume and are further prevented from aligning and forming long tunnel-shaped voids by the staggered layout of the beads. We analyze results and discuss applications in section 7.3.

7.1 Background on phasor noise

Our technique relies on the phasor noise [Tricard et al., 2019]. A phasor noise ϕ is a procedural noise that generates oscillating patterns suitable to create fiber-like patterns. It is defined as the argument of a complex Gabor noise G [Lagae et al., 2009]:

$$\phi(x) = \arg\left(G(x)\right) \tag{7.1}$$

with :

$$G(x) = \sum_j e^{-b\|x-x_j\|^2} e^{2if_j d_j.(x-x_j)+i\varphi_j} \tag{7.2}$$

It is an unstructured representation, defined from the convolution of complex Gabor kernels with random seed positions x_j. The kernel properties locally define the noise properties: its bandwidth b, frequency f_j, the direction of anisotropy d_j, and phase shift φ_j. These different kernels generate a scalar field in 3D space: an oscillating signal of precisely

controlled direction and scale. In our work, we use those oscillations to define trajectories for material deposition.

Without special treatment, a phasor noise exhibits local defects located around singularities of the phase field. These result in a less coherent pattern and, in our case, would result in less continuous deposition trajectories. In Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre [2020] two methods are combined, diminishing the number of defects in the generated pattern significantly. First, the phase φ_j of neighboring kernels is iteratively adjusted in a simple and efficient optimization process. Second, a closed-form regularization filter is applied when sampling values from the phasor noise. Through these combined methods, highly regular anisotropic patterns are obtained. The high quality and regularity of the fields enable the definition of local parametric patches for microstructure design; for more details, please refer to Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre [2020]. In our work, we extend this methodology to define deposition trajectories for dense infill patterns.

The phasor noise presents several advantages for our application. First, it is a meshless process. In other words, it can be computed in any Euclidean space while following arbitrary input fields. Secondly, it scales to large volumes as the complexity of the iterative phase alignment, in both space and time, is only proportional to the number of kernels – not the resolution of the generated patterns. Finally, the entire framework maps well onto GPUs, enabling fast, parallel optimization and evaluation of regularized phasor fields. For the rest of this chapter, it is essential to remember that the phasor noise is defined in 3D.

7.2 Our approach

Our technique takes as an input a 3D model for which to generate 3D trajectories and a direction field d^U that is everywhere orthogonal to z (the manufacturing direction). The algorithm defines two crossed phasor fields used to generate oscillating patterns everywhere throughout the object volume. The oscillation frequency f is chosen to match the target trajectory spacing – typically the nozzle width w (with $f = 1/w$). However, by varying the frequency, we naturally support varying deposition widths, see subsection 7.2.2. The direction of the phasor fields closely follows d^U.

Once the fields are optimized, we proceed layer by layer to efficiently generate actual trajectories. This step extracts a set of discrete curved trajectories from the continuous definition of two phasor fields. We describe it in subsection 7.2.1. Note that this is performed in a streaming fashion (we only process a fixed number of slices simultaneously) and that slices anywhere in space may be extracted in parallel.

To prevent trajectories from creating inter-layer artifacts by always being precisely aligned, we introduce a *phase shift* every other layer. This shift produces staggered trajectories across layers while still closely following the direction field.

7.2.1 Defining trajectories

This section describes our method to define dense infills (and their associated trajectories) following the user's arbitrary direction field d^U, whose directions must be constrained to

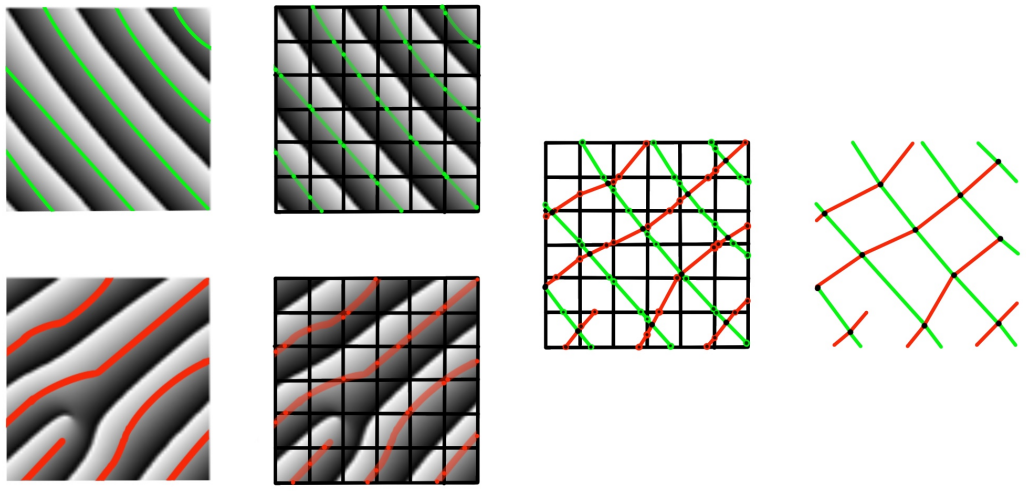


Figure 56: Representation of the graph extraction process. First, both phasor fields are sampled; their arguments are remapped in [0,1] for visualization. The set of curves C_π^U is highlighted in green while C_π^V is in red. Then intersections of C_π^U and C_π^V with edges of the grid are computed separately, creating multiple polylines. The intersection between polylines of both sets is computed to define the graph's vertices (in black here). Finally, we collapse edges until all vertices either belong to the two edge sets (valence equal four) or are curve end-points (valence equal one).

the slicing plane. Let us define d^V as a second field orthogonal to d^U, also constrained to the slicing plane (i.e., $d^V = d^U \times z$). Note that d^U and d^V are defined as 3D fields, and their values can evolve along the Z-axis. As a result, the trajectories' direction can evolve across layers.

The trajectory extraction is based on two orthogonal periodic variable fields, U and V, obtained from two phasor fields. To any given point x in space we associate a value pair $UV(x) \in [0, 2\pi]^2$. We note $U(x)$ and $V(x)$ the two components of $UV(x)$ (see Figure 56, left). In practice, $U(x)$ (respectively $V(x)$) defines a locally oscillating signal taking the shape of a sawtooth. This signal defines a periodic variable field following the direction d^U (and respectively d^V). From this periodic variable field we can construct a set of curves defined by $U(x) = \pi$ (respectively $V(x) = \pi$). The generated curves C_π^U (respectively C_π^V) are orthogonal to d^U (respectively d^V). Therefore, we define the trajectories as the set of curves C_π^V (see Figure 56, left). The orthogonal set of curves C_π^U is used during the computation of inter-trajectories distances and defines the sampling rate of the trajectories (see subsection 7.2.3).

In practice, the extraction of the trajectories is performed in three steps. First, we define and optimize our periodic variable fields (see subsection 7.2.2). Then, for a given printing slice, we extract a planar graph in whose vertices are defined as the intersection between C_π^V and C_π^U (see Figure 56, right). Edges of the graph are subdivided into two sets, E^U and E^V. The edges of E^U (resp. E^V) form a tessellation of the C_π^V curves (respectively C_π^U). The graph extraction is described in subsection 7.2.3.

Finally, we use E^U to define the trajectories and use E^V to measure the local spac-

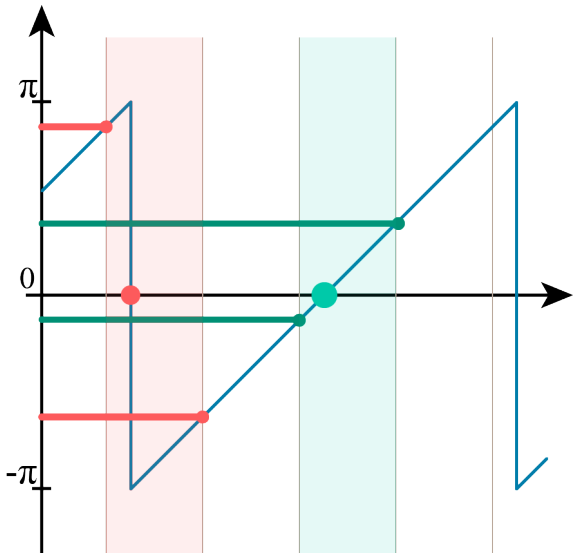


Figure 57: When $U(i) - \pi$ and $U(j) - \pi$ have opposite signs, we detect if the field as a 2π jumps. If it does not jump, then it is a valid vertex, and invalid otherwise.

ing between them. This information is later used to adjust the deposition flow (see subsection 7.2.4).

7.2.2 Periodic variable fields

We define the two local periodic variable fields U and V by relying on two orthogonal 3D phasor noises (see Equation 7.1). The kernel directions d_i are computed by sampling the input direction field d^U (resp. d^V) at the kernel position x_i. The frequencies f_i are chosen as $1/w$, with w the target deposition bead width. Note that w can be made to spatially vary by sampling an additional user's input field. Finally, parameter φ_i is computed by iterative phase alignment [Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre, 2020].

7.2.3 Graph extraction

As previously stated, we seek to extract two sets of edges E^U and E^V from the periodic variable field UV. Recall that while the fields are defined in 3D, the trajectory extraction is performed in 2D, layer by layer (possibly in parallel). We base our approach on a modified version of the marching square algorithm [Wyvill et al., 1986; An et al., 2018].

First, we sample our periodic variable field on a regular 2D grid which resolution is chosen to sample at four times the maximum field frequency. This sampling frequency is chosen to help to filter the 2π jump in the periodic variable fields. Then, for each square cell of the grid, we extract C^U_π and C^V_π separately. We extract vertices for the intersection between C^U_π (respectively C^V_π) along each edge of the cell. More precisely, given i and j the grid points defining a processed edge, we extract a vertex if $U(i) - \pi$ and $U(j) - \pi$ have opposite signs. We filter out vertices that do not verify $|U(i) - U(j)| \leq \pi$ as they are generated by the field's 2π jumps (Figure 57).

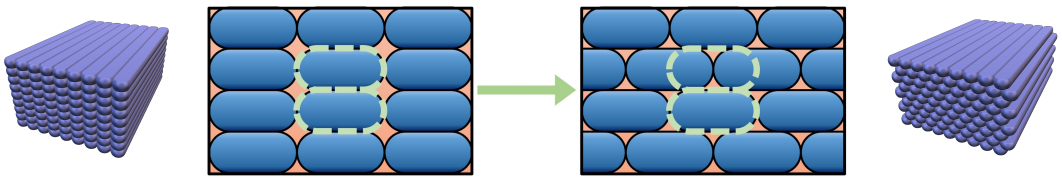


Figure 58: Trajectories can be offset every other layer by shifting the phases of the kernels. This offset allows the creation of staggered beam layouts. Handmade 3D example and sectional view of deposition paths. ***Left:*** *aligned.* ***Right:*** *staggered.*

This condition ensures robustness concerning the periodicity of the field.

The positions of the vertices are chosen based on the linear interpolation of the U (resp. V) parameter along the edges.

We only extract an edge, for a given field, in a given cell if precisely two vertices are created. Edges extracted from fields U (resp. V) are registered in the set E^V (resp. E^U). This extraction is done for both U and V, and a cell can contain 0 to 2 edges. Suppose the cell contains two intersecting edges (defined from two distinct fields). In that case, the latter two are subdivided, and a new vertex is created at the intersection point, resulting in four new connected edges.

When each square cell of the grid is processed, we stitch all the partial results together. To do so, we collapse edges until all vertices either belongs to the two edge sets (vertices of valence 4 verifying $UV \approx \{\pi, \pi\}$) or are curve end-points (vertices of valence one close to singularity or on the border of the domain).

This reduces the sampling frequency of C^V_π (respectively C^U_π) to the frequency of the V (respectively U). Thus the frequency of U controls the polygonization error in curved trajectories. In practice, we use the same frequency for both fields as they generate a sampling equal to or close to the nozzle diameter, which is enough for our application.

Note that we keep track of the edge origin during the decimation, thus updating the edge sets E^U (edge marked as V) and E^V (for edge marked as U).

7.2.4 Clipping and ordering trajectories

To extract the final trajectories from the graph, we clip the set of edge E^U by the contour of the slice; then, we use a greedy approach to extract the longest uninterrupted paths. We compute its average distance to all its direct neighbors on the E^V set for each point. This average distance (which represents the local bead width) is clamped to remain between half and twice the nozzle size. It is then used to adjust the deposition volume proportionally. Finally, we transform the trajectories into GCode and adjust the material deposition at each point by considering the inter-spacing previously computed.

We order the trajectories following a simple nearest-first heuristic (the first trajectory is picked at random, then the subsequent trajectory is the one with the closest end-point). This heuristic provides reasonably good results at a low cost. An ordering can be visualized in Figure 60, right.

7.2.5 Staggered bead layout

To obtain a staggered layout, we offset the periodic variable fields for every odd layer. This layout is easily achieved by shifting the phases of the kernels φ_i when odd sampling layers, adding an offset $\delta = \pi$. Doing so produces a pattern where the beads of one layer are shifted by half the width of the beads from the layer below, forming a coherent staggered layout. Note that this works naturally across directions and frequency (width) controls, as we directly manipulate the phase of the coherent 3D oscillations. This simple approach produces remarkably regular and aligned patterns, as can be seen in Figure 58.

7.3 Results

In this section, we discuss the specific properties of our method and demonstrate the range of possibilities it offers.

All our tests are done with fused-filament 3D printing, the most widespread additive manufacturing technique. Parts were printed with a variety of printers. Blue CuteOcto was printed with a Prusa MK2S, parts in grey with a micro delta, parts in multicolor with a modified Prusa printer equipped with a diamond mixing extruder, other parts were printed using a Creality CR10S pro. Note that all parts shown here are solely composed of infill to illustrate the method. In practice, our infills would likely be combined with a contouring technique for the outer object shell, replacing the internal typical zigzag solid fill. The list of all model sources is provided in Table 7.1. Our method was implemented in C++ using the Vulkan API, and shaders were programmed with GLSL. Trajectories are always extracted on the entire bounding box of the object before being clipped to the surface. Therefore, we measure runtimes for cubes of increasing volumes and only measure timings related to our method (e.g., alignment of kernel phases and graph extractions, leaving out the actual clipping of trajectories, scheduling, and GCode generation). Runtimes were measured on an i7-6700HQ CPU (@2.6GHz only one thread used) and GTX1070 graphic card and are given in Table 7.2.

The first thing to notice is that phase alignment runtime is bound to $28s$. This alignment is due to our strategy to compute kernel density and bandwidth such that the phase alignment of kernels can be done in a single pass on GPU while avoiding memory overflow (the maximal number of kernels used is approximately eight million). A streaming strategy could be adopted to increase the number of degrees of freedom in more significant parts if needed. As expected, the graph extraction (including the sampling of the phasor noises) is linear in terms of processed volume. The grid size for the marching square is based on the target bead width (the length of an edge is one-quarter of the minimal target bead width), and 64 slices are computed at once. For all tested models, the GPU memory used remains bound to $1.7GB$. Should the memory consumption become problematic, a smaller set of slices and XY tilling could be used.

Table 7.1: Table of all models used in the article, Figures they can be seen in, and their sources

Model	Figure	Sources
Gear	55	custom made
SCF	55	custom made
Badge	65	thingiverse.com/thing:1670621
Cute octo says hello	60, 63, 62	thingiverse.com/thing:27053
Screwdriver handle	64	thingiverse.com/thing:34852
Gecko	66	thingiverse.com/thing:1363148
Fandisc	67	github.com/dengwirda/jigsaw-models

Volume (cm^3)	Total (s)	Phase optimisation (s)	Graph extraction (s)	Time over volume (s/cm^3)
1	2.81	1.64	0.79	2.81
8	17.11	9.27	6.54	2.14
64	82.05	28.14	48.97	1.28
512	355.32	28.20	323.12	0.5

Table 7.2: Computation time in function of the volume for a 0.4mm nozzle and a layer height of 0.2mm.

7.3.1 Comparison to curve hierarchies

When a bi-variate parametric field is available, curve hierarchies may be used to produce trajectories that follow the field while roughly equally spaced [Elber and Cohen, 1996; Ezair et al., 2018]. We compare our approach to a curve hierarchy in Figure 59. As can be seen, the curve extraction leaves many gaps. It typically happens where the field exhibits a high divergence, as is the case with the radial field shown in Figure 59 (left). There, steep transitions of local density from 1 to 0.5 occur: the change of hierarchy level introduces a sudden jump. In addition, the gaps occur along specific transition curves of the field, which in turn leads to coherent defects within and across layers, weakening the overall part structure. Note that other curves extraction techniques could alleviate this problem [Etienne and Lefebvre, 2020].

In comparison, our method exhibits fewer transitions with tiny gaps well distributed throughout a slice (Figure 59 (right)) – keeping in mind that the paths are additionally laid out staggered from one layer to the next. Finally, recall that this is achieved directly from the direction field *without* the need to optimize for a bi-variate parametric field in the first place.

7.3.2 Singularities, staggered layout

Figure 60 shows the CuteOcto model sliced with a radial direction field and constant bead width (nozzle diameter). As can be seen, the branching pattern and split/merge locations are evenly distributed within each slice.

Figure 62 reveals the effect of the staggered layout across layers. It is the same model

as Figure 60 with a sectional view of the beads. Left: phase shifts disabled; Right: our result with the staggered layout. This cut reveals that singularities have a limited extent in the Z direction. It also shows how our staggering layout (right) reduces the influence of singularities by breaking the gap coherence, increasing the bond between beads in these areas.

It is essential to notice that the staggered layout visibility is reduced on printed parts (see Figure 55(left), Figure 63) due to travel artifacts between trajectories at the boundary of the object (visually similar to larger beads).

Another slicing of CuteOcto presents the combination of direction field and deposition bead width field (see Figure 63 and Figure 61). *Direction field* is defined as the gradient of the signed distance field to the surface, while the bead width field is radially defined.

Control over the direction field can help for a wide range of applications, from controlling the appearance to the physical properties of the printed object.

In the screwdriver handle (see Figure 64), the direction field is used to control the outer texture of the object. A grip-like pattern is created in the locally convex area (using trajectories locally orthogonal to the surface), while smooth surfaces are used in the concave area (trajectories parallel to the surface).

The direction field can also be used to control the appearance of horizontal parts of an object where filament directions are visible (see Figure 55(right) and Figure 65).

The direction field can also be used to align paths along branches in branch-like structures (see Figure 66). In this case, the direction field was created to favor long trajectories across branching regions (see junctions between legs and torso).

Finally, the direction field can also be used to adapt fiber direction in mechanical parts (see Figure 55(left) and Figure 67).

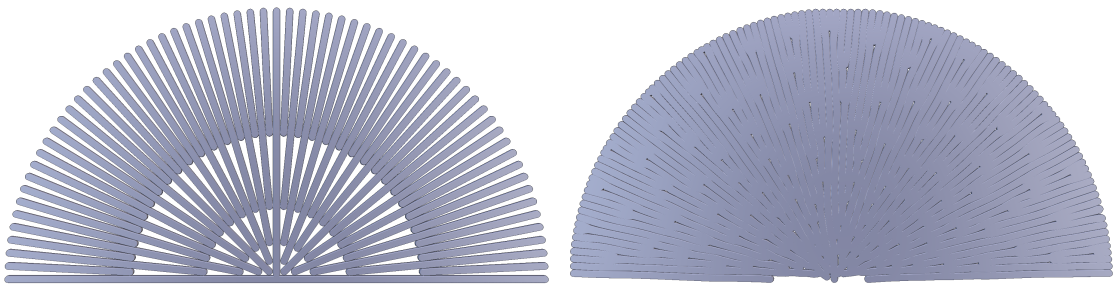


Figure 59: Comparison to Ezair et al. [2018] on a radial direction field. ***Left :*** *method of Ezair et al. [2018], steep transition of local density can be observed.* ***Right:*** *our methods, note that unavoidable tiny gaps are well distributed throughout the slice.*

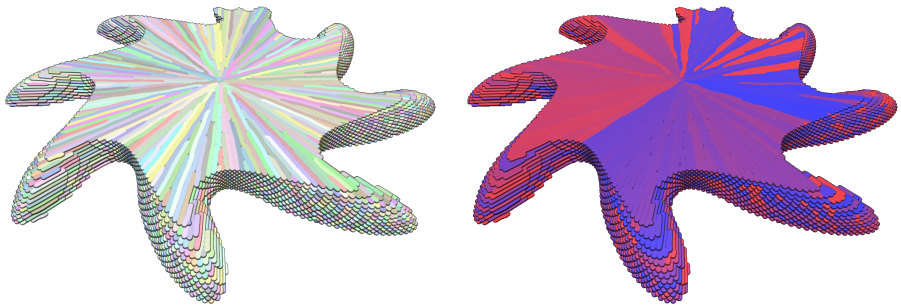


Figure 60: CuteOcto model sliced with a radial direction field, constant target beamwidth, and staggered beam layout. ***Left:*** *one layer with colored beads.* ***Right:*** *beads colored by order of printing.*

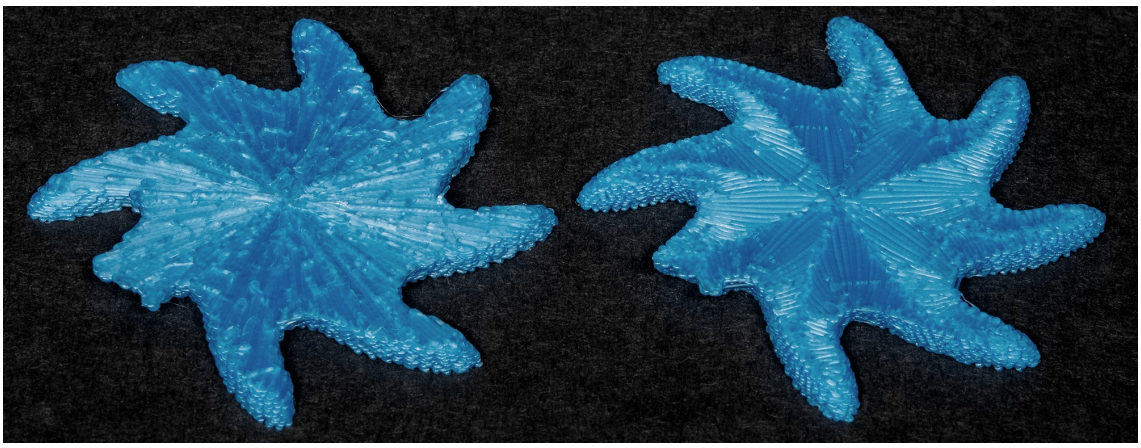


Figure 61: First slices of the CuteOcto model with constant (left) and varying (right) target bead width.

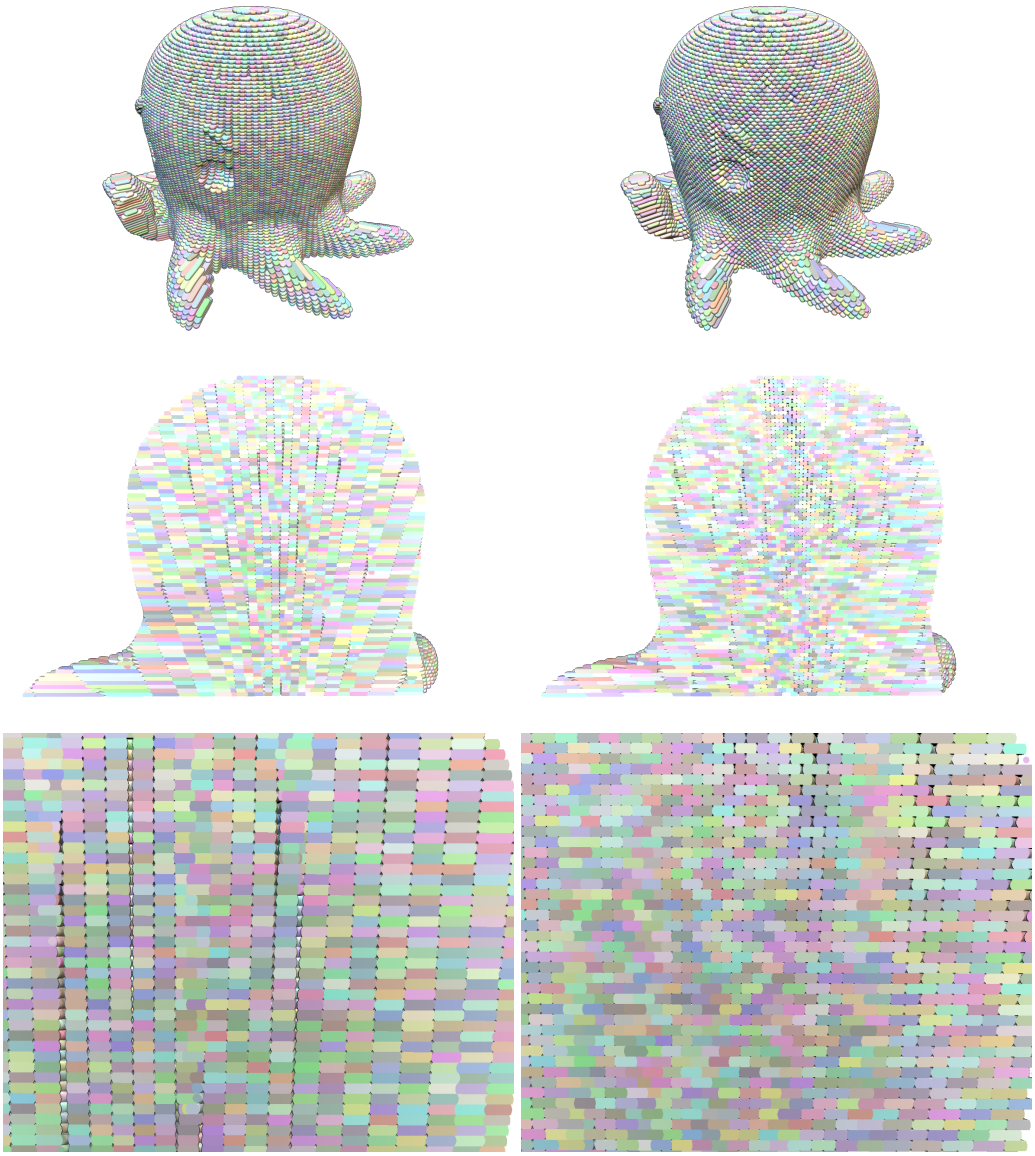


Figure 62: A sectional view of the beads within the CuteOcto model volume, under a radial direction field. ***Left:*** *phase shifts are disabled. Note the narrow vertical gaps near singularities.* ***Right:*** *our staggered layout breaks the gap coherence. It increases the bond between beads in these areas and inherently throughout the volume. Note that singularities are placed differently as the optimization of phasor kernels is non-deterministic.*

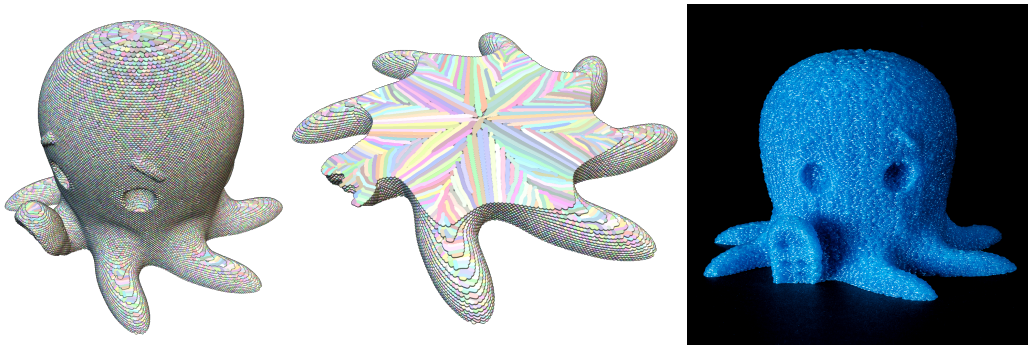


Figure 63: Bead width can be controlled with an additional user input field. CuteOcto model sliced with a radially defined beamwidth. Direction field is defined as the gradient of the signed distance field to the surface.

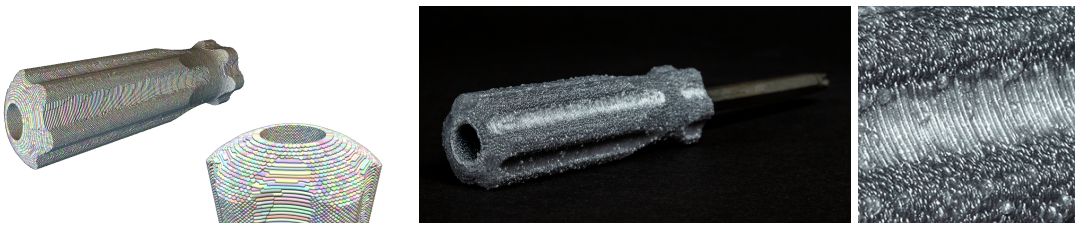


Figure 64: Direction field near the surface of an object can be used to control surface roughness. Here a screwdriver handle have a grip-like surface in convex area (direction orthogonal to the surface) and a smooth surface in concave area (direction in surface tangent plane).

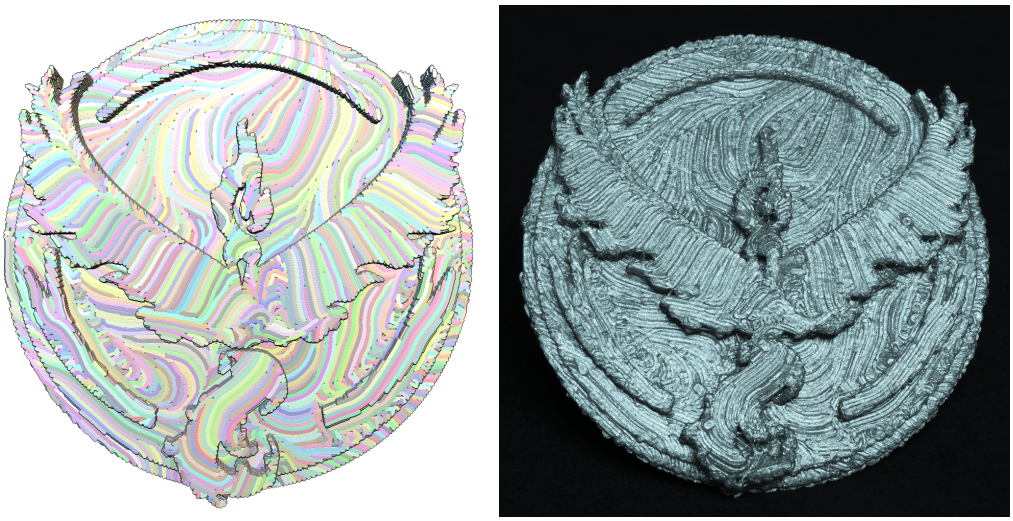


Figure 65: Direction field can be used to texture horizontal area by using the fact that filament deposition direction is clearly visible on 3D printed models.

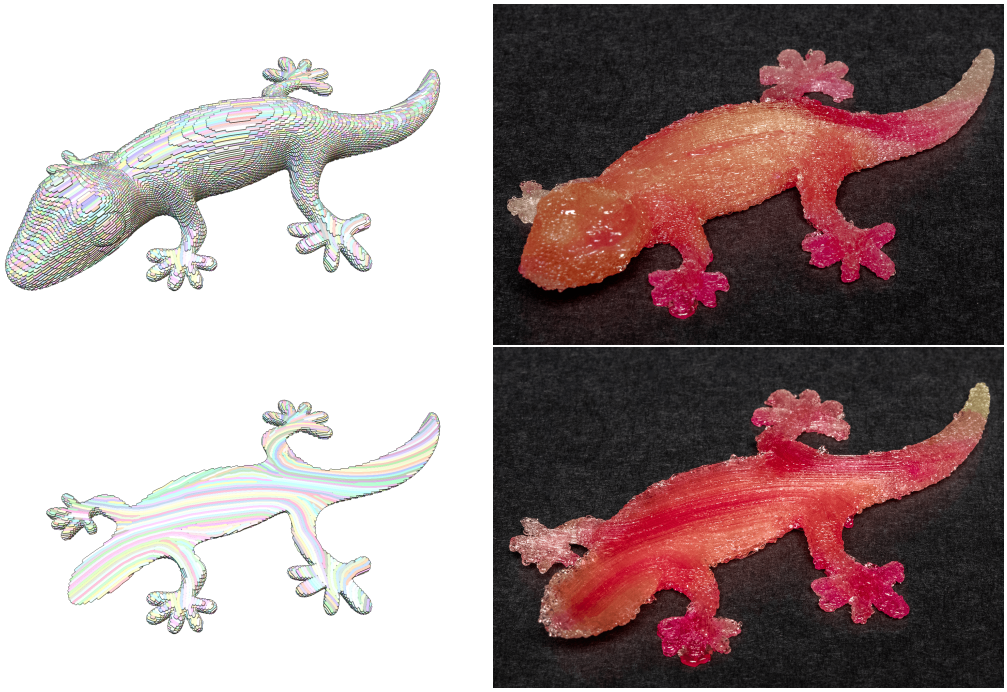


Figure 66: Gecko object exhibiting a branching structure. ***Left:*** *Paths are generated in the direction of branches while allowing long deposition paths in the neighborhood of the junctions.* ***Right:*** *The object was printed with a printer equipped with a diamond mixing extruder. The color mixing ratios were chosen to highlight trajectories' direction and continuity.*

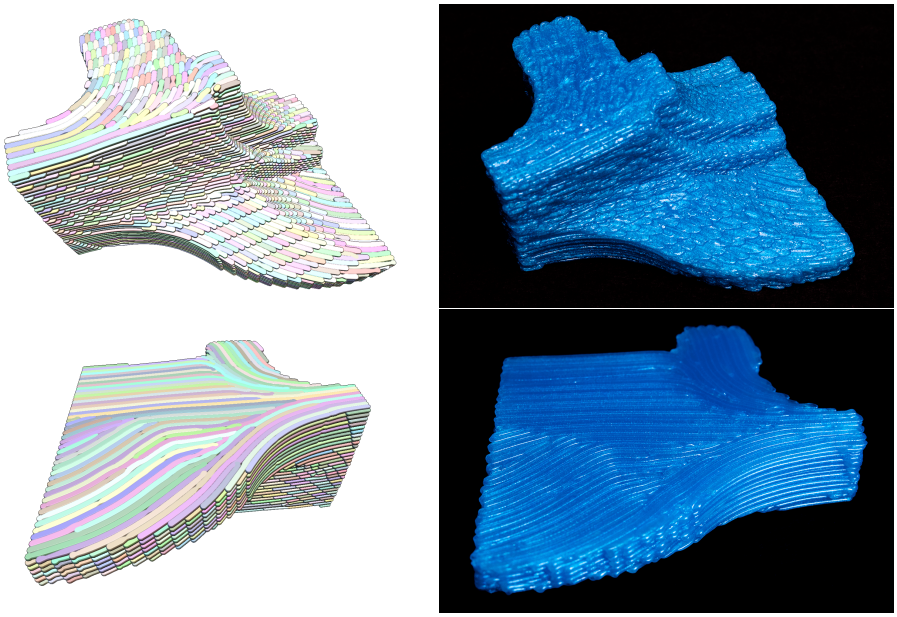


Figure 67: A mechanical part. A view from below is provided in the second row to highlight the input direction field.

7.4 Discussion and conclusion

We presented a method to stagger the beads forming a slice while orienting them. The staggered layout is achieved by adding a phase shift of π on the result of the phasor noise every odd slice. We chose this specific phase shift to offset the beads by half their size at each layer while in practice, it is possible to achieve different staggered layouts by varying the phase shift and the sets of layers onto which it is applied.

In our implementation, we input a 3D voxel field to define the orientation of the beads. In all our examples a field of $64 \times 64 \times 64$ voxels was used, ensuring reasonable compute times while giving adequate control.

To define the different orientation across the field, the user hand-painted it in a dedicated tool and then we sampled it to orient each phasor kernel. Other definitions of orientation fields could be used — e.g., procedural — provided they can be sampled at the center of each phasor kernel.

As explained in the "Constraints, defects and overcoming them" [p.37] section, using dense and strongly oriented patterns can create deformations (shrinkage and warping) while printing. This effect stems from the tendency of the polymer to retract when cooling but did not create specific difficulties when printing our example parts with PLA material. The 3D nature of the field allows the distribution of imperfections evenly throughout the volume. It also introduces a way to control spatial coherency across layers, leading to the staggered bead layout. While in-depth mechanical studies are required, we believe this inter-layer layout control to be a promising direction of research, as this degree of freedom has not been available before.

Before being printed, the paths are ordered using a greedy approach, where the next bead to be printed the one with an extremity closest to the last printed position. This heuristic encourages nearby paths to be printed successively while being fast to compute. Naturally, other ordering methods could be devised depending on the material and process requirements.

Interestingly, phasor fields have links with global parametrization techniques for quad- and hex-meshing, as discussed in Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre [2020]. Therefore, such approaches could likely be modified for a purpose similar to ours. However, our approach inherits the low setup cost, scalability, and simplicity of phasor noises, enabling the efficient extraction of trajectories.

Our method presents a few limitations. In comparison to traditional infill patterns, it produces heavier GCode due to the complex curved trajectories, that could be partially alleviated by relying on circular arc GCode capabilities or removing unnecessary points of the paths. When generating paths that follow the local curvature of the object, our approach does not guarantee a perfect alignment between the object boundary and the deposition paths. This non-alignment results in some trajectories being cut when encountering the object boundary, increasing the surface roughness in those areas. While surface appearance is not the main scope of this technique, it could be an exciting venue for future work. This could be approached by controlling the phase of the Gabor kernels where a precise alignment is required.

But the most exciting venue of future work in the context of this thesis stems in the potential to be usable for curved printing. Indeed, with this signal representation, this

approach paves the way to out of plane printing without having to define a parametrization field. As we will see in Chapter “Curved printing” Chapter 9, “Another brick in the wall: Orientable trajectories for additive manufacturing” [p.139], there are strong analogues between curvying and covering within a “2D” layer and doing it in 3D.

Conclusion

These two works, "Procedural band patterns" [p.75] and "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91], had the same goal, to create 3D printing paths. While the first one started from paths for 3D printing to finally become a rendering shader, the second one started from a shader to become printing paths. Procedural band patterns is a sparse infill which offers a good control over the density while Staggered infill is a dense infill that aims for fully filled layers. Both of them exhibit an excellent control over the orientation and can easily extends to 3D. They have the same limitations over the more complex curved paths, with a lot of singularities and heavy GCode, but the Phasor infill is easier to use. It's important to note that Progressive Band Patterns requires a global parametrization and in areas with high variation (either in orientation or density), the bands will be distorted while thanks to the phasor noise, the staggered infill is more stable.

Part IV

Curved printing

Contents

Introduction **113**

8 CurviSlicer: Slightly curved slicing for 3-axis printers **115**
8.1 Overview . . . 116
8.1.1 Fabrication constraints . . . 116
8.1.2 Improving surfaces . . . 117
8.1.3 Choosing whether to flatten surfaces . . . 118
8.2 Algorithm . . . 120
8.2.1 Main algorithm . . . 120
8.2.2 Fabrication constraints . . . 121
8.2.3 Objective function . . . 122
8.2.4 Relaxing flatness constraints . . . 124
8.2.5 Applying the mapping and its inverse . . . 125
8.2.6 Controlling the number of layers . . . 125
8.3 Results . . . 127
8.3.1 Printed models . . . 127
8.3.2 Influence of optimization parameters . . . 128
8.3.3 Quality . . . 128
8.3.4 Discussion . . . 128
8.4 Discussion and future work . . . 137

9 Another brick in the wall: Orientable trajectories for additive manufacturing **139**
9.1 Multi-axis print constraints . . . 139
9.2 Our approach . . . 140
9.2.1 Periodic variable fields . . . 140
9.2.2 Defining trajectories . . . 140
9.2.3 Graph extraction from a phasor field . . . 141
9.2.4 Layer extraction . . . 143
9.2.5 Path ordering . . . 143
9.2.6 Avoiding collisions . . . 146
9.3 Conclusion . . . 146

Discussion and future work **149**
1 Layer unstacking . . . 149
2 Layers from Procedural Band Pattern . . . 149
3 Layers based on a mapping . . . 149

" *Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.* "

Donald E. Knuth

Introduction

A perfect print surface would be smooth, without staircase defects or support scars. To get as close as possible to this objective, it would be necessary to print while perfectly following the surface curvature. However, traditional FDM printing involves slicing 3D objects into 2D layers. Flat layers present intrinsic limitations that limit the accuracy and mechanical performance of the printed parts.

This leads to one limitation of the FDM process, the staircase defect produced when the print geometries are not aligned with the deposition plane. A second one would be related to the deposition trajectories themselves. Related work shows that the deposition path's direction directly influences the part's strength. So to control the physical behavior of the object, we must control the orientation of deposition in 3D, not per layer.

So why do we still use flat layers? First of all, not many technologies can go out of the plane as it needs to be a 'vector-based' technology with specialized tools that can move freely in space to deposit material. So, if we remove the inadequate technologies, why do we still use planar layers for the others who can deposit material out of the plane? We can easily imagine robotic arms continuously depositing material without having the slightest problem, and it has even been shown for specific cases, like *Support-free volume printing by multi-axis motion* [Dai et al., 2018]. Here comes the second point; generating freeform movement is not trivial. Thus it is preferable to think in terms of layers to remove degrees of freedom. Constraints to the plan ensure the printing head will never collide with the already printed part, no matter the order of the paths.

As we have seen Chapter "Related work" Chapter 5, "Out-of-plane printing" [p.55], there is no method to get a perfect print, only snippets of answers (support free) or specific applications (wireframe).

In this chapter, we investigate further into curve printing. Our ultimate goal would be to have a spaghetti-like deposition. Not that we really enjoy pasta, but imagine being able to fill an object with a space-filling curve going in any direction. If the deposition paths must follow a direction field, we just align the spaghetti, so they follow that field. If the aim is aesthetic, like a smooth and perfect surface, we cover your object with the longest path (or the minimum number). This is an excellent and challenging problem and thus a good long-term goal. However, let us take a step in this direction. We will start more gently by relaxing the layer planarity to keep the simplicity of the slicing process: we curve the space, see "CurviSlicer: Slightly curved slicing for 3-axis printers" [p.115]. Then, we will investigate whether we can have more freedom, attempting to generalize the approach of Chapter "Adaptive infills" Chapter 7, "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91] with phasor to produce curved trajectories without having to produce explicit and traversal layers.

Chapter 8

CurviSlicer: Slightly curved slicing for 3-axis printers

As seen in Chapter "Related work" Chapter 1, "Additive manufacturing technologies" [p.15], many processes offer additional degrees of freedom. In particular, FDM allows one to deposit molten plastic along curved paths, as long as it is deposited onto an existing surface. This opportunity is used in current methods tackling the generic problem of free-form deposition, e.g., using the 6-DoF of a robotic arm. While offering many advantages over traditional flat deposition, these systems require expensive hardware (6-DoF robotic arms or 5-axis motion platforms), limiting their applicability. We propose an algorithmic solution to enable curved deposition using standard FDM machines, with the only requirement of having a specific nozzle shape — no flat area around the molten filament exit hole — which is the case on many printers already. Since replacement nozzles for the most common printers are available at a modest price (< 10 USD), our contribution has the potential to be widely adopted by makers and 3D printing companies. The key idea of our technique is first to compute a deformation of the input object, second to slice the deformed solid using standard uniform planar slicing, and lastly to deform the toolpaths back into the original space. By constraining the volumetric mapping, we guarantee that there will be no collision between the part and the extrusion device and that deposition thickness remain within feasible bounds. We propose a specific parameterization of the problem that reduces it to a simple set of constraints that can be solved using a standard QP solver. The mapping is optimized to reduce the surface reproduction error during slicing. When mapping back to the initial space, the deposition paths become curved, but the surface quality improvement is preserved. Our technique significantly improves surface accuracy and finish over traditional FDM, in particular strongly reducing the staircase defects due to planar layering. Our technique compares favorably to previous methods relying on robotic arms, offering a highly effective solution with minimal hardware requirements. The code is open source to foster the adoption of our technique with a dedicated branch for the reference implementation of our method.

8.1 Overview

Our approach starts from a mesh Ω correctly defining a solid (e.g., an STL file for fabrication) and oriented such that the Z-axis is the build direction. We then optimize for a mapping $\mathcal{M}$ from the object space to the slicing space. The mesh to be sliced is obtained through the mapping as $\mathcal{M}(\Omega)$. A standard slicer is then called to produce a set of toolpaths $\mathcal{T}$, using uniform slicing. The final toolpaths used for fabrication are obtained through the inverse mapping $\mathcal{M}^{-1}(\mathcal{T})$. Our algorithm optimizes a deformation for both the object inside and surrounding space, thus allowing for travel paths and auxiliary structures (e.g., supports) to be adequately curved alongside the object. The pipeline is illustrated in Figure 68.

The crux of the problem is how to compute $\mathcal{M}$ (and its inverse) to enforce all constraints while minimizing surface defects.

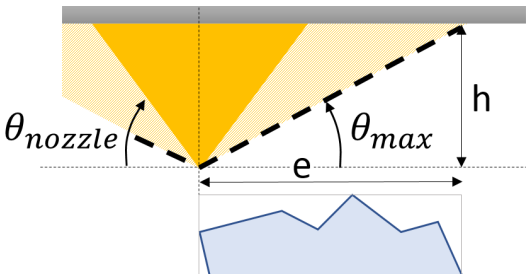
In terms of constraints, we have to ensure that no collisions occur and that the deposition thickness remains within the minimum and maximum bounds after mapping (Section 8.1.1).

In terms of objectives, we seek to improve surface finish, reduce staircase defects, and accurately follow the initial surface (Sections 8.1.2 and 8.1.3).

8.1.1 Fabrication constraints

In this work, we face two main fabrication constraints. The first relates to avoiding collision between the extrusion device (nozzle, extruder, and carriage) and the printed part, while the second relates to feasible deposition thicknesses. We assume the printer to be equipped with a pointed, conical nozzle without a flat area around the exit hole.

We model the collision constraint as an inverted cone. The Figure inset shows the cone constraint (dashed line), the printer nozzle (orange), the carriage (gray), and the object below (blue). The constraint cone has its apex aligned with the nozzle tip and represents the forbidden space above it (lighter orange hatches). It has to guarantee that if the part remains below the cone's surface, no collisions can occur with any part of the printer — first and foremost, the nozzle itself.



The collision cone must at least contain the nozzle, which typically has a conic geometry. We denote the angle of the nozzle cone with respect to the horizontal by θ_{nozzle} (see inset Figure). We denote by θ_{max} the collision cone angle with respect to the horizontal. It is obtained as $\theta_{max} = min(\theta_{nozzle}, \tan^{-1}\frac{h}{e})$, where h is the distance between the nozzle tip and carriage and e the maximum XY extent of the printed object.

We have to ensure that no already printed path enters the forbidden cone during deposition. This collision avoidance translates into a local constraint on the slope of the paths after inverse mapping: how quickly they are allowed to rise in Z by units of X and Y.

Deposition thickness bounds stem from the nozzle exit diameter. A general rule of thumb is that the maximum thickness should stay within $[0.1d, 0.75d]$ where d is the nozzle diameter. This rule gives from 0.04mm to 0.3 mm for a typical 0.4 mm nozzle. The

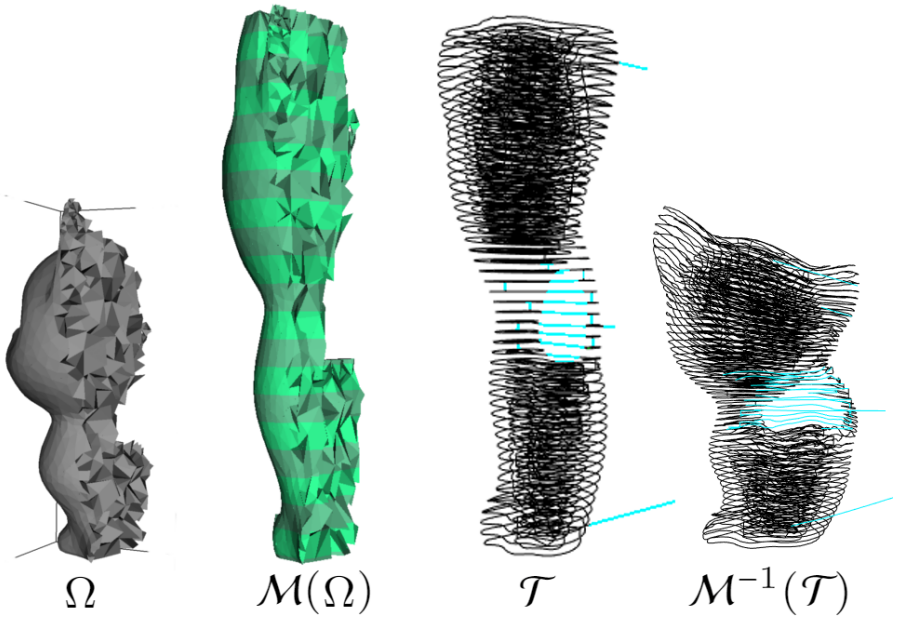


Figure 68: Overview. We start from a 3D model Ω and optimize a mapping $\mathcal{M}$. We slice the deformed model $\mathcal{M}$ to obtain the toolpaths $\mathcal{T}$, which are mapped back into the initial space as $\mathcal{M}^{-1}(\mathcal{T})$ for fabrication.

minimal thickness constraint is also impacted by printer mechanical quality and calibration accuracy. Indeed, as thickness decreases, even minor calibration errors significantly impact the result. For do-it-yourself printers, the typical minimal thickness is 0.1 mm – even though well-calibrated printers reliably print at 0.05 mm or less. We denote by τ_{min} and τ_{max} the minimum and maximum thicknesses.

8.1.2 Improving surfaces

The primary defect produced by planar slicing is the so-called staircase effect: the emergence of visible steps along the build direction. The defect is more pronounced when the slope of the surfaces decreases: the vertical sampling density of the planar slices is no longer sufficient to prevent significant steps from appearing between slices (see Figure 69). This fact leads to the observation that printed vertical walls exhibit minimal staircase effect while gently sloped surfaces suffer the most prominent staircase defects.

While using thinner slices reduces the size of the defect, it remains visible on surfaces with low slopes. In addition, thinner slices imply longer print times.

There is, however, a notable particular case: an exactly flat surface produces no staircase and thus is ideally reproduced *if a slice precisely aligns with the surface*.

To improve surface reproduction, we seek to minimize the staircase error of the printed part. We propose an attempt to make all surfaces either vertical or flat during the slicing step, as both cases lead to minimal errors. The consequence is to curve the slices, locally adapting to the surface slopes, as illustrated in Figure 70. Of course, due to the fabricability constraints, this ideal objective can only be approximated.

Note that to avoid errors on flat areas while slicing, the surface has to align with the top

of a slice precisely. Otherwise, significant errors occur due to misalignment. These errors are a specific source of concern that we address in our approach.

8.1.3 Choosing whether to flatten surfaces

A key question when computing the deformation is which set of surfaces should be flattened. Our proposal is that, ideally, we would like to flatten as many faces as possible—these regions later print as curved surfaces with no staircase error and offer accurate reproduction when well aligned with the tops of slices. However, if we seek to flatten a surface, we must ensure a single, well-aligned slice reproduces it. A misalignment or residual slope would have the surface intersect a slicing plane, resulting in a significant staircase error.

We, therefore, initially attempt to flatten all surfaces that could be reproduced under the maximum printable slope θ_{max}. This flattening typically leads to the most accurate results (we provide measurements in Section 8.3.3). Optionally, the user can choose a smaller initial set, allowing the selection of which surfaces are flattened.

The initial set usually leads to an infeasible problem due to fabrication constraints. The freedom required to incline, compress, or stretch slices around large flattened surfaces is often unavailable.

These non-flattenable surfaces lead to a challenging problem: flattening cannot be expressed as soft constraints, as any minor violation immediately results in worst-case stair-stepping errors. Instead, we propose an iterative scheme where we eliminate infeasible flattening requirements progressively, turning them into surfaces to be inclined. The scheme is described in Section 8.2.4.

Note that we never attempt to flatten downward-facing triangles, as, in general, these could not print without adding supports in a 3-axis system. There is one exception. The triangles that are flat in the input—either downward facing or upward facing—are *always* constrained to remain flat. This constraint, in particular, implies that the object's bottom remains flat.

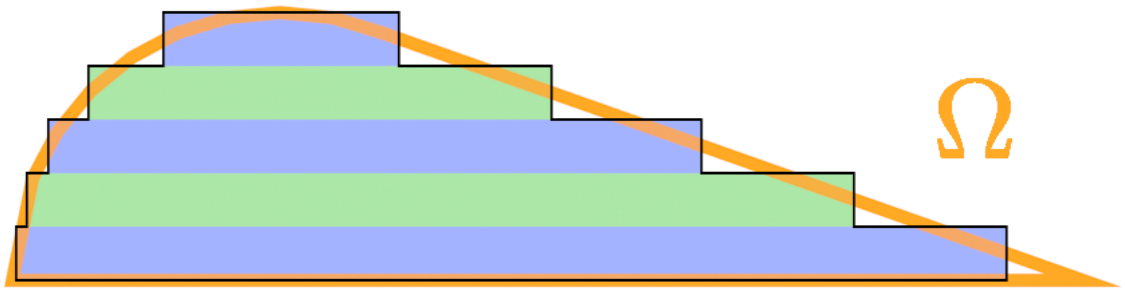


Figure 69: Slicing a part with uniform slicing, side view (actual result). The model outline to be matched is in orange, the slices are shown in green/blue. The staircase is more pronounced when the slope of the surfaces decreases: the left part suffers less from the staircase defect than the right part.

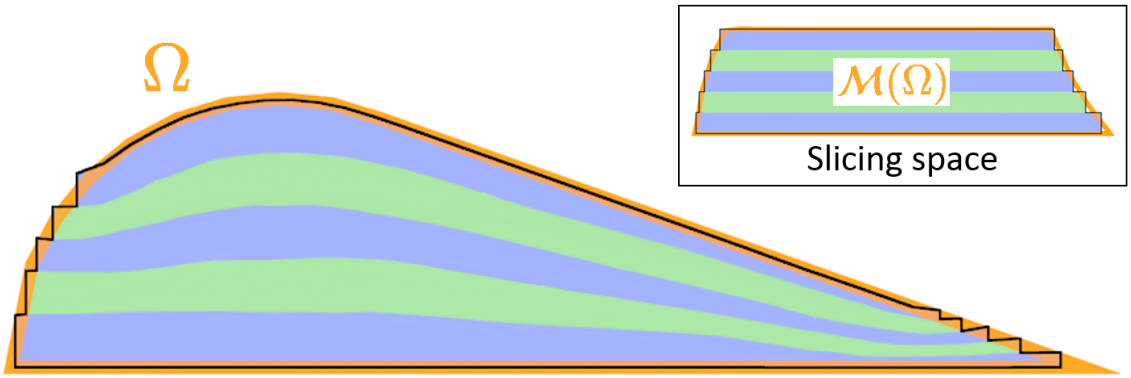


Figure 70: Slicing a part with curved slicing, side view (actual result). The model outline to be matched is orange, and the slices are shown in green/blue. Our algorithm curves the slices to follow the object surface accurately whenever possible. In this case, the slice thicknesses are allowed to vary from 0.1*mm to* 0.6*mm, while the maximum slope is constrained to* $\theta_{max} = 30$ *degrees. The inset shows the deformed space where slices have been produced. Note the alignment of the top surface with the top of the last slice.*

8.2 Algorithm

Prior to any computation, we remesh the input with TetWild Hu et al. [2018], to obtain a tetrahedral mesh Γ for both the inside and outside volume, as well as the triangle mesh Ω at their interface.

Minimizing staircases amounts to making surfaces either exactly flat, or making them as vertical as possible (Sections 8.1.2 and 8.1.3). To achieve this we optimize for a mapping that deforms the surfaces along the vertical direction only, locally compressing or stretching the initial solid and changing surface slopes. This parameterization of the problem is in line with the constrained motions of a 3–axis system, while leading to a practical optimization scheme only involving minimizing for quadratic objectives under linear constraints.

We represent the mapping $\mathcal{M}$ as a deformation field of Γ, computing a new vertical position $h(p)$ for each of its vertices p. These new positions define a continuous deformation field within the volume of Γ: any point inside can be mapped through linear interpolation from the enclosing tetrahedron vertices. Swapping optimized and original coordinates switches between $\mathcal{M}$ and $\mathcal{M}^{-1}$.

The unknown variables of the problem are thus the values $h(p)$, which will be optimized to define a mapping under the desired objective and constraints.

Additional notation. In the following we denote by $x(p)$, $y(p)$ and $z(p)$ the coordinates of vertices p of Γ in the original, undeformed space. We denote by $\mathcal{F}$ the set of surface triangles in Γ, that is the tetrahedron faces that lie on the boundary of the solid. We denote by $\underline{\mathcal{F}}$ the set of triangles to be flattened and by $\overline{\mathcal{F}} = \mathcal{F} \setminus \underline{\mathcal{F}}$ its complement: the triangles that we seek to incline. Since these sets change during optimization and relaxation, we index them as $\underline{\mathcal{F}}^0, ..., \underline{\mathcal{F}}^i$.

We denote by $t =< p_0, p_1, p_2 >$ a triangle in $\mathcal{F}$ and $\mathbf{n}_t$ its normal. We denote by $\mathbf{z}$ the vertical (build) direction. Faces pointing upwards verify $\mathbf{n}_t \cdot \mathbf{z} > 0$. We similarly denote tetrahedrons as $< p_0, p_1, p_2, p_3 >$ in Γ, and denote by Γ_I the set of inner tetrahedrons and by Γ_O the set of outer tetrahedrons.

Finally, we denote as $\mathcal{M}_h$ the mapping obtained within Γ from a vector of vertical positions h.

8.2.1 Main algorithm

Our main algorithm is iterative: each iteration starts with a previous solution (initialized with $h^0 = z$) and a set of surfaces to flatten. It then solves for new vertical positions h^{i+1} by minimizing an objective function on Γ under fabrication constraints. The objective attempts to flatten surfaces in $\underline{\mathcal{F}}^i$ and to incline surfaces in $\overline{\mathcal{F}}^i$ towards the vertical. The flattening requirements are often unfeasible if strictly enforced. Thus, we express flattening as a soft constraint (objective with a high weight).

Once h^{i+1} is obtained, the surfaces in $\underline{\mathcal{F}}^i$ — connected components of neighboring triangles — are checked for flatness. If all are sufficiently flat, the loop terminates and we proceed with slicing. If not, we relax the flattening objective by keeping only a subset of triangles from $\underline{\mathcal{F}}^i$ in $\underline{\mathcal{F}}^{i+1}$.

Note that the constrained minimization has at least one solution: the identity deformation $h = z$. Therefore, at worst this process always terminates as the flatness test passes for $\underline{\mathcal{F}} = \varnothing$; or rather, when $\underline{\mathcal{F}}$ contains only the set of surfaces already flat in the input as these are never relaxed. In practice, the algorithm succeeded in flattening more surfaces on all our test cases.

All main steps are described in the next Sections: Fabrication constraints in Section 8.2.2, optimization of h^i in Section 8.2.3, relaxation in Section 8.2.4, slicing and toolpath mapping in Section 8.2.5.

Algorithm 1 Curvislicer main algorithm

Initialize $\underline{\mathcal{F}}^0$, $h^0 \leftarrow z$
Setup fabrication constraints on Γ
for all i, from $i \leftarrow 1$ **do**
    $h^i \leftarrow \arg\min_h E(h, h^{i-1}, \underline{\mathcal{F}}^{i-1}, \overline{\mathcal{F}}^{i-1})$ on Γ
    if surfaces in $\underline{\mathcal{F}}^{i-1}$ using h^i are flat, or $\underline{\mathcal{F}}^{i-1} == \varnothing$ **then**
        $h \leftarrow h^i$
        break
    end if
    $\underline{\mathcal{F}}^i \leftarrow$ Relax($\underline{\mathcal{F}}^{i-1}, h^i$)
end for
$\mathcal{T} \leftarrow$ slice $\mathcal{M}_h(\Omega)$
return $\mathcal{M}_h^{-1}(\mathcal{T})$

8.2.2 Fabrication constraints

Thickness constraints. The thickness constraints are captured by limiting the local stretch that the mapping is allowed to introduce. The minimum and maximum allowed stretches are respectively 1 and $\frac{\tau_{max}}{\tau_{min}}$ with τ_{min} and τ_{max} respectively the min/max admissible thicknesses on the target printer. We slice the model at τ_{max} before deformation by $\mathcal{M}^{-1}$.

Using this setup, a stretch of 1 in $\mathcal{M}$ will lead to having curved slices with a local thickness of τ_{max}, while a stretch of $\frac{\tau_{max}}{\tau_{min}}$ results in a local thickness along the curved slices of τ_{min}. Any stretch above or below that range would violate the fabrication constraints.

The deformation field is defined by linear interpolation within the tetrahedrons. Within each tetrahedron, the gradient ∇h of the vertical coordinates h is thus constant. We formulate the thickness constraint directly on the gradient ∇h.

Let us consider a tetrahedron with vertices $< p_0, p_1, p_2, p_3 >$. The gradient of a function f defined on the vertices and linearly interpolated within is obtained as $\nabla f = \sum_{i=0}^{2} w_k(f_k - f_3)$, where $f_k = f(p_k)$ and w_k are vector weights in $\mathbb{R}^3$. These weights (nine unknowns) can be computed ahead of time for each tetrahedron, setting $f = x$, $f = y$ and $f = z$ to obtain nine equations. For more details, we refer the reader to literature on linear tetrahedra.

For each tetrahedron $t \in \Gamma_I$ we write the constraints as:

$$1 \quad \leq \quad z(\nabla h_t) \quad \leq \quad \frac{\tau_{max}}{\tau_{min}}$$

where $z(\nabla h_t) = \sum_{k=0}^{2} z(w_k^t) \cdot (h_k - h_3)$ with $z(w_k^t)$ the Z coordinate of the weight k of t and $h_k = h(p_k)$.

Slope constraints. The maximum admissible slope θ_{max} depends on the printer setup and printed part, as detailed in Section 8.1.1. The constraint is applied to ∇h within each tetrahedron, preventing the gradients along X and Y from varying too fast with respect to the gradient along Z. We write the constraints for each tetrahedron $t \in \Gamma_I$ as:

$$-z(\nabla h_t) \leq \frac{x(\nabla h_t)}{\tan(\theta_{max})} \leq z(\nabla h_t)$$
$$-z(\nabla h_t) \leq \frac{y(\nabla h_t)}{\tan(\theta_{max})} \leq z(\nabla h_t)$$

where $z(\nabla h_t) = \sum_{k=0}^{2} z(w_k^t) \cdot (h_k - h_3)$ with $z(w_k^t)$ the Z coordinate of weight k for the gradient within t and $h_k = h(p_k)$. The terms $x(\nabla h_t)$ and $y(\nabla h_t)$ are similarly defined.

8.2.3 Objective function

The objective is optimized under strict fabrication constraints (see Section 8.2.2). We also add a set of constraints to prevent foldovers in Γ_O. This is done by imposing for all $t \in \Gamma_O$ that $z(\nabla h_t) > 0$. Note that foldovers cannot occur in Γ_I thanks to thickness constraints.

Given the set of surfaces $\underline{\mathcal{F}}$ and $\overline{\mathcal{F}}$ and a previous solution h^{i-1}, the objective to minimize is made of four different terms :

$$\begin{aligned} E(h, h^{i-1}, \underline{\mathcal{F}}, \overline{\mathcal{F}}) =& \lambda_f E_{flat}(h, \underline{\mathcal{F}}) + \lambda_a E_{align}(h, h^{i-1}, \underline{\mathcal{F}}) \\ &+ \lambda_s\, E_{slope}(h, \overline{\mathcal{F}}) + \lambda_m\, E_{smooth}(h) \end{aligned}$$

E_{flat} attempts to flatten selected triangles while E_{align} encourages flat areas to align with slice tops. E_{slope} rotates other triangles towards the vertical. E_{smooth} regularizes the problem by guiding it towards smooth solutions, in particular in the less constrained empty regions. The λ_i weights control the tradeoff between the terms. For our target object scales (300 mm maximum in extent) and layer thicknesses (0.05 to 0.6 mm) we determined that a good tradeoff is given by $\lambda_f = 30$, $\lambda_a = 1$, $\lambda_s = 0.1$ and $\lambda_m = 0.02$. We use this setup for all results.

If significantly different scales are targeted, the λ_i weights should be rescaled noting that E_{flat} and E_{align} have the scale of τ_{max}^2, E_{slope} has the scale of L^2 with L the maximum object size (printer bed extent), and E_{smooth} is dimensionless.

Flattening. The objective E_{flat} is written as:

$$E_{flat}(h, \underline{\mathcal{F}}) = \sum_{t \in \underline{\mathcal{F}}} \frac{A(\{t\})}{A(\underline{\mathcal{F}})} \Big(\sum_{\substack{i,j \in [0,2] \\ i<j}} (h(t_i) - h(t_j))^2 \Big)$$

with t a triangle, t_i its i-th vertex and where $A(.)$ computes the area of a set of triangles in the initial model. This attempts to put all vertices of each triangle in $\underline{\mathcal{F}}$ to be at the same height.

Aligning. The alignment objective encourages flattened areas to be aligned with slice tops. We slice the object after deformation using uniform slicing, therefore the slices are located every $k \cdot \tau_{max}$ with k an integer. This allows to compute an alignment error with respect to the result of the previous iteration h^{i-1}.

To perform the alignment we consider connected components c of triangles to be flattened — sets of neighboring triangles in $\underline{\mathcal{F}}$. We attempt to snap the average height of each flat component to a slice top. We compute the height of a component c as a weighted average of its triangle positions, $H(c,h) = \sum_{t \in c} \frac{A(\{t\})}{A(c)} \frac{\sum_{i=0}^{2} h(t_i)}{3}$. We define the snapping position from the previous iteration as:

$$Snap(c, h^{i-1}) = \tau_{max} \left\lfloor 0.5 + \frac{H(c, h^{i-1})}{\tau_{max}} \right\rfloor$$

The alignment objective is defined as:

$$E_{align}(h, h^{i-1}, \underline{\mathcal{F}}) = \sum_{c \in \mathcal{C}(\underline{\mathcal{F}})} Flat(c) \frac{A(c)}{A(\underline{\mathcal{F}})} \left(H(c,h) - Snap(c, h^{i-1}) \right)^2$$

where $\mathcal{C}(\underline{\mathcal{F}})$ are connected components of $\underline{\mathcal{F}}$. $Flat(c)$ selects which alignment objectives are active, returning 1 or 0. An objective is active if 1) the connected component c is *flat* – as defined next – and 2) all *larger* components are also flat. The second rule prevents deciding on an alignment on small components before larger components are properly aligned.

Component flatness. We evaluate the non-flatness of a component in $\underline{\mathcal{F}}$ by computing the average max height difference between triangle vertices, weighted by triangle area:

$$err(c,h) = \sum_{t \in c} \frac{A(\{t\})}{A(c)} \max_{\substack{i,j \in [0,2] \\ i<j}} (h(t_i) - h(t_j))^2$$

This takes into account the fact that smaller triangles contribute less to the final error. A component is said flat if $err(c,h) < (\frac{\tau_{min}}{8})^2$. However, we reject as non flat any component where a triangle would have vertices separated by more than $\frac{\tau_{max}}{2}$: these could cross a slice boundary, producing a staircase.

Changing slope. The objective E_{slope} seeks to make surfaces in $\overline{\mathcal{F}}$ vertical. Given a triangle t with normal $\mathbf{n}_t$, we compute a tangential frame with vectors $\mathbf{u}_t = \frac{\mathbf{n}_t \wedge \mathbf{z}}{||\mathbf{n}_t \wedge \mathbf{z}||}$ and $\mathbf{q}_t = \mathbf{u}_t \wedge \mathbf{n}_t$. When changing the vertices heights $\mathbf{q}_t$ rotates around $\mathbf{u}_t$ (see Figure 71). The objective attempts to align $\mathbf{q}_t$ with $\mathbf{z}$, and thus we define a target vertical direction $\overline{\mathbf{q}}_t = sign(\mathbf{q}_t \cdot \mathbf{z}) \cdot \mathbf{z}$ where $sign(.)$ returns -1 if its argument is negative, 1 otherwise.

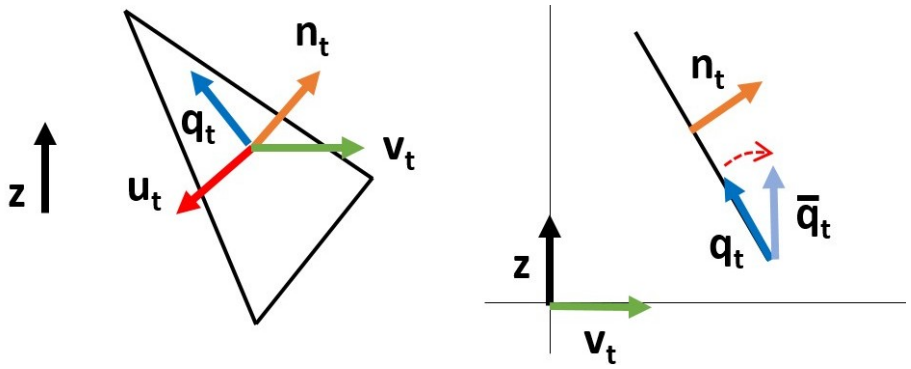


Figure 71: ***Left:*** *3D view of the triangle with normal* $\mathbf{n}_t$*, tangent frame* $\mathbf{u}_t$*,* $\mathbf{q}_t$*.* ***Right:*** *Side view of the* $\mathbf{z}, \mathbf{v}_t$ *plane, showing the rotation of* $\mathbf{q}_t$ *around* $\mathbf{u}_t$*.*

The objective seeks to obtain the correct distances between the vertices along $\overline{\mathbf{q}}_t$, that is:

$$E_{slope}(h, \eta) = \sum_{t \in \overline{\mathcal{F}}} \frac{A(\{t\})}{A(\overline{\mathcal{F}})} \Big(\sum_{\substack{i,j \in [0,2] \\ i<j}} (a_i - a_j + ((a_j - a_i) \cdot \mathbf{q}_t) \cdot \overline{\mathbf{q}}_t)^2 \Big)$$

where t is the triangle $< p_0, p_1, p_2 >$ and a_i is a vector using the original vertex XY coordinates and the variable height as the Z coordinate: $(x(p_i), y(p_i), h(p_i))$.

We exclude triangles that are already vertical, since no change of vertex heights can modify their angle. It is logical to exclude them, since there will be no staircase defect on a vertical slope, regardless of the slice thickness.

Smoothness. The smoothness term encourages neighboring tetrahedrons to be modified in a similar manner, that is, to have the same gradient. This is written as:

$$E_{smooth}(h) = \sum_{t \in \Gamma} \sum_{n \in \mathcal{N}(t)} \frac{V(\{t\}) + V(\{n\})}{V(\Gamma)} \Big(\nabla h_t - \nabla h_n \Big)^2$$

where $\mathcal{N}(t)$ is the set of tetrahedrons sharing a face with t, $V(.)$ returns the volume of a set of tetrahedrons, and ∇h_t is the (vector) gradient of t.

Solver. The problem we formulate is a quadratic objective minimization under linear constraints. We use the Gurobi solver Gurobi [2018] to obtain a solution at each iteration.

8.2.4 Relaxing flatness constraints

After obtaining a solution h^i for the current iteration, we check whether connected components in $\underline{\mathcal{F}}^{i-1}$ are all properly flattened. If not, we proceed to remove some of

the flattening requirements. This process is reminiscent of segmentation techniques in PolyCube optimization Gregson et al. [2011].

Removing many triangles from $\underline{\mathcal{F}}$ at each iteration would quickly lead to satisfying solutions, but may miss opportunities to flatten some areas. Removing them one by one—canceling the triangle with maximum vertex height differences—would require as many iterations as there are triangles in $\underline{\mathcal{F}}^0$, which is unreasonable.

We propose the following strategy. We observe that in most cases the error is concentrated along the boundaries of the connected components of $\underline{\mathcal{F}}^{i-1}$. We therefore relax all the boundary triangles that exceed the flatness threshold. In this process, we consider as boundary triangles only those whose neighbors were relaxed before. The rationale is that it is best to relax the flattened region progressively along the same front.

If there is no boundary triangle in a component—which is always the case on the first iteration—we relax the triangle with the largest vertex height differences. Intuitively, this opens a tear in the non-flat component that will then grow to relax the problem.

Note that, at each iteration, we relax all components which account for less than 5% of the total area to be flattened.

8.2.5 Applying the mapping and its inverse

The optimization gives us the mapping $\mathcal{M}_h$ from the initial space to the slicing space. We first apply the mapping to the input model, in order to obtain the mesh that will be sliced with a standard slicer. The mapping has been optimized to use τ_{max} as uniform slicing thickness.

After slicing we obtain a set of toolpaths in the form of G-code for filament printers. We apply the inverse mapping $\mathcal{M}_h^{-1}$ to the toolpath coordinates to obtain the final curved toolpaths.

As the deformation may vary along each toolpath, we first resample the toolpaths using a sampling rate matching the nozzle diameter. We then transform each point back to the inital space.

As the toolpath points lie on the top of each slice, we first offset them down by half a layer thickness, $\frac{\tau_{max}}{2}$. We then locate each point within $\mathcal{M}(\Gamma)$. Having identified the enclosing tetrahedra, we interpolate the z coordinate of the point from the original tetrahedron vertices. This gives us a new point in the inital object space, located in the middle of a slice.

We offset it back to the slice top. This requires knowing the local thickness. This information is directly available from the gradient of h within the enclosing tetrahedra. We similarly adjust the material flow, and adjust the deposition speed to maintain a constant extrusion rate.

8.2.6 Controlling the number of layers

By attempting to make surfaces vertical, our optimizer tends to make the object as tall as possible in the slicing space – equivalently using the thinnest slices. To control the number of layers we simply constrain the vertices of $\mathcal{F}$ to never exceed a certain height. That is, to

obtain n layers we add the constraint:

$$\text{forall } p \in \mathcal{F}, h(p) < h(p_{bottom}) + n \cdot \tau_{max}$$

where p_{bottom} is the lowest vertex of $\mathcal{F}$.

This affords for a time–quality trade-off that can be convenient for the user.

8.3 Results

We show a variety of prints in Section 8.3.1, we explore the influence of the input parameters in Section 8.3.2, and we provide measurements of surface accuracy in Section 8.3.3, comparing our technique to optimal discrete adaptive slicing Alexa et al. [2017].

8.3.1 Printed models

We produced a number of results using an off-the-shelf Ultimaker 2 (UM2) printer with minor modifications. It is equipped with a 0.8mm nozzle and we allow thicknesses from 0.6mm down to 0.1mm (1/6 ratio). To allow for increased freedom of motion we removed the metal part holding the fans around the nozzle. We use a wider nozzle to print with thicker layers, so as to clearly reveal the effect of curved slicing. We also used a standard Anet A8 printer for Figure 74.

All models use the same 30 degrees angle constraint θ_{max} and the algorithm attempts to flatten all surfaces below this angle, unless otherwise specified. Table 8.1 summarizes the main statistics and optimization times for the models shown in the paper. Optimization typically takes a few minutes, with the time depending essentially on the input size and number of relaxation iterations.

Figure 72 shows a mechanical part with a slanted area. Curved slicing closely follows the surfaces, printing the entire part around the hole with a matching slope. Note how the slices go from flat to curved in the bottom part. Figure 72 also compares with the result of uniform and adaptive slicing for the same number of slices.

Figure 73 shows a frog model sliced with our technique and adaptive slicing, using a low number of slices. Figure 74 shows the same model printed with thinner layers on a standard Anet A8 (cooling fan removed for clearance). In both Figures, note the smooth curved top of the frog. A bigger frog model is also shown in Figure 75. Note how the slices match the overall angle of the main body, and how the slices become thicker at the vertical extremity (mouth).

Figure 55 shows a wing cross-section model, revealing how curved slicing can accurately reproduce an entire curved top. The model is more accurately reproduced with our approach.

Model	# Tetrahedrons	# Iterations	Optimization time
Wing	9 711	4	< 1 minute
Foil cutter	19 740	3	< 1 minute
Anklebase	25 607	3	< 1 minute
Frog (small)	36 224	3	< 1 minute
Sports car	58 708	17	20 minutes
Kitten	133 838	3	14 minutes
Frog (big)	200 743	3	22 minutes

Table 8.1: Statistics for all models, with 0.1-0.6 mm layer thickness and $\theta_{max} = 30$. The first column is the number of tetrahedrons, the second column is the number of relaxation iteration and the third column reports total optimization times (Intel i7-4790K, 4 cores).

Figure 76 compares two car model prints, one via curved slicing and the other via adaptive slicing. The curved slices nicely follow the car outline at the top of the part, as well as reproducing the wheels more accurately. This reveals how fewer slices can be used to better reproduce slanted surfaces, allowing the reallocation of other slices to the more detailed parts of the model.

8.3.2 Influence of optimization parameters

We now study the impact of the optimization parameters. Figure 77 illustrates the effect of the angle constraint θ_{max} on an example model, showing the deformation obtained on printers offering different degrees of freedom in terms of achievable angles. The results use a similar number of slices, but smaller areas are flattened: increased angular freedom leads to larger smooth surfaces.

Figure 78 illustrates the effect of the thickness ratio $\frac{\tau_{max}}{\tau_{min}}$ on the same model. A ratio closer to one leaves little freedom to deform the model upwards, resulting in using fewer slices. However, the same set of surfaces are flattened.

In summary, increasing θ_{max} allows larger curved surfaces, while increasing $\frac{\tau_{max}}{\tau_{min}}$ leads to thinner slices.

Finally, Figure 79 illustrates the use of the number of layers target (Section 8.2.6). Please note, however, that models obtained this way are usually less accurate (in terms of volume error) than those obtained automatically.

8.3.3 Quality

We compare the surface accuracy of our results to that of uniform slicing and adaptive slicing in Figure 80. We use the volume assignment error, computed by discretization Alexa et al. [2017]. We run an optimization of our models to obtain the reference number of slices for comparison, and run each slicing algorithm on Ω using the same number of slices. As can be seen from the left column of each comparison in Figure 80, curved slicing generally results in superior accuracy.

To evaluate the importance of flattening, we force the optimization to start from artificially smaller sets of surfaces to flatten $\underline{\mathcal{F}}$ (all but first columns of each comparison in Figure 80). As expected, curved slicing is less accurate with less flattening: flattened surfaces — equivalently curved surfaces in the original space — are those that benefit the most from our approach. Nevertheless, thanks to verticalizing, even when little flattening is performed curved slicing produces results which are better than uniform slicing, even though less accurate than adaptive slicing on some models.

8.3.4 Discussion

Overall we can see that our technique outperforms both uniform and adaptive slicing. This is true both in terms of accuracy (volume error) and surface finish.

It is especially remarkable that our technique produces results that are more accurate than *optimal* adaptive slicing. This is explained by several factors. At low numbers of slices, adaptive slicing has little freedom, whereas we curve the deposition and align with surfaces.

Adaptive slicing cannot perform local adaptations, while by curving our algorithm can reallocate slice thicknesses as required. This is true within the same region of the object but also for disconnected components within a slice. Finally, adaptive slicing allocates many slices to low slopes, while we capture low slopes with a single slice.

However, optimal discrete slicing Alexa et al. [2017] is orders of magnitude faster as it computes slicing plans for all slices at once. Also, while we can target a specific number of slices (Section 8.2.6), we cannot guarantee that the target is reached.

In many cases the visual aspect of the prints does not reflect the actual volumetric error. This is for instance the case on the cars in Figure 81: The orientation of the top surface fill — the zigzag pattern — changes the perceived smoothness. The surface is nevertheless smoother than that of the planar slicing result. To reduce such issues the slicer could be made aware of a preferred direction for the covers.

It is worth noting that we leave vertical surfaces free in our approach. In the idealized vertical slice model, a vertical wall produces no error regardless of the layering thickness. In practice, on filament printers, slices have a rounded profile that changes the perceived appearance. We could add a penalty requiring vertical surfaces to be as tall as possible (and hence use as thin as possible slices). Nevertheless, we found that leaving them free adds flexibility to the solver and generally results in better solutions.

Finally, we did not consider structural properties in this evaluation. Based on observations from previous work prototyping curved 3D printing, we expect the prints to be stronger Singamneni et al. [2012]; Huang and Singamneni [2012]; Lim et al. [2016] than those obtained with planar slicing. As future work we will quantitatively evaluate the change in structural properties.

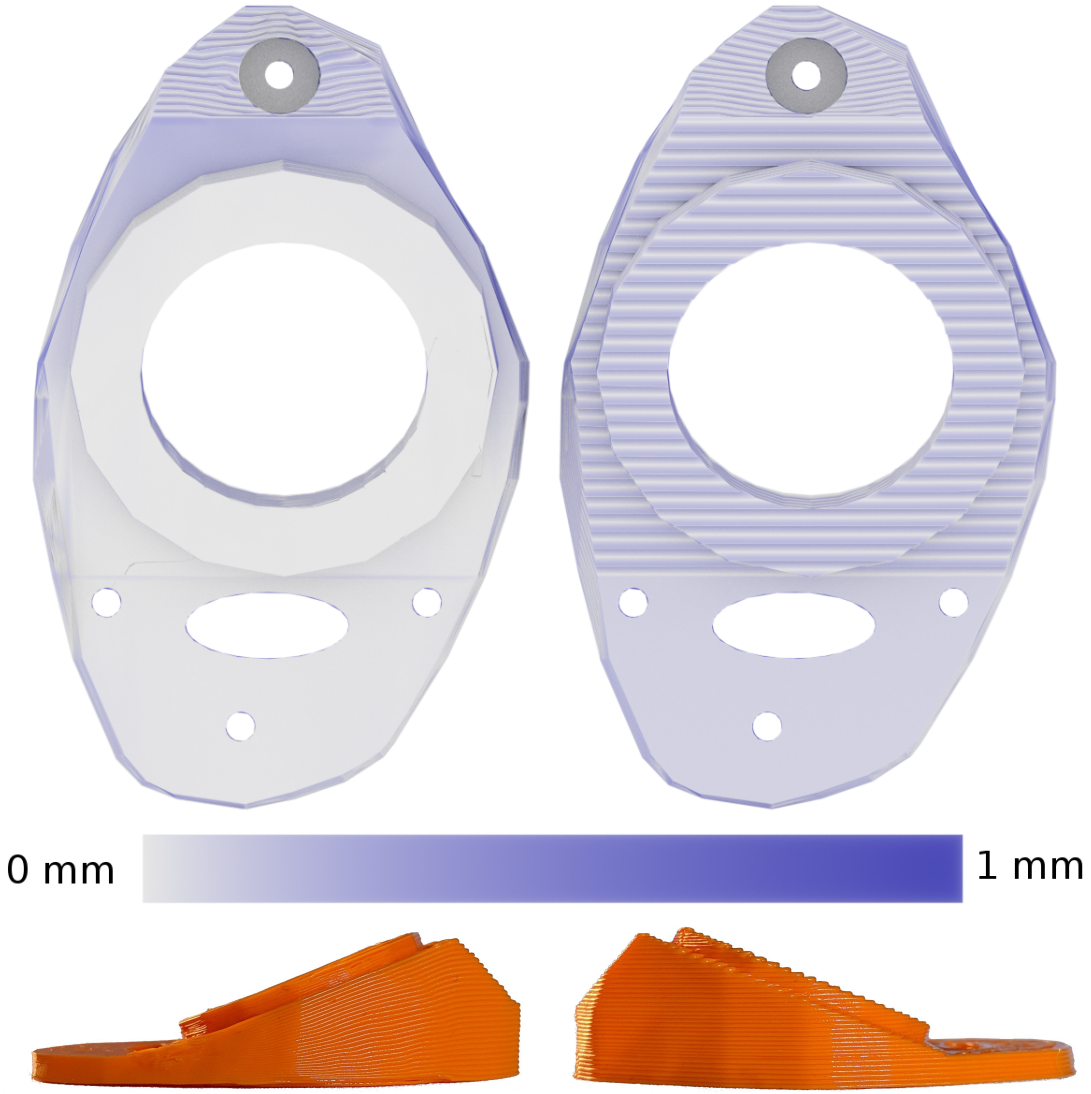


Figure 72: Anklebase model with different slicing algorithms, all using 38 slices. ***Top:*** *views of distance maps between the input mesh with curved slicing (left) and uniform slicing (right). A lighter color indicates a reduced error.* ***Bottom:*** *Sides views of curved slicing print (left) and adaptive slicing (right). All staircasing is eliminated while closely following the input.*

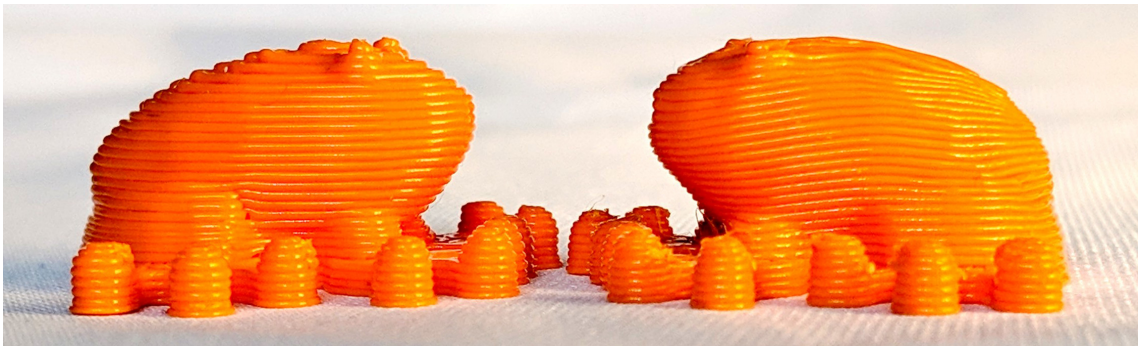


Figure 73: Frog model, adaptive slicing (left) and curved slicing (right) with 27 slices. Printed on the UM2. Note the difference in the silhouette.

Figure 74: Small frog model printed with curved slicing on an Anet A8 printer, with a 0.4*mm nozzle and* 103 *layers between* 0.3*mm and* 0.05*mm. Layers at the top smoothly follow the curvature.*

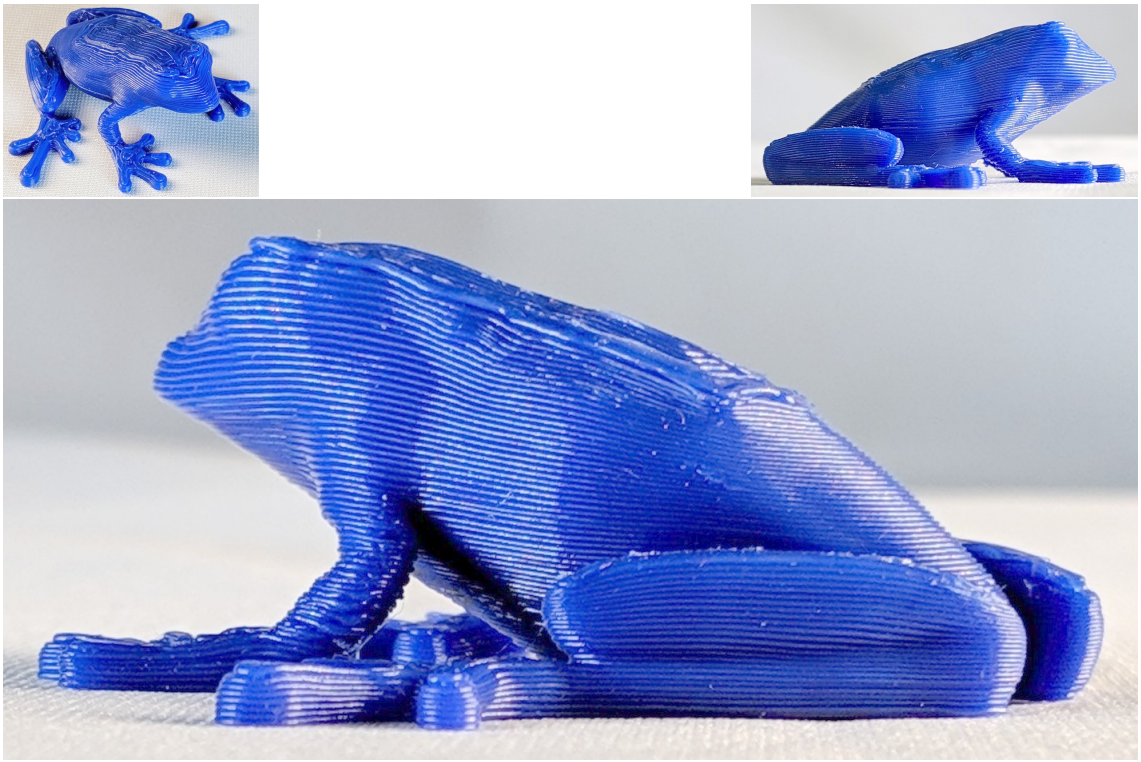


Figure 75: Big frog printed on the UM2 with 68 curved layers. Note how the layers have been curved to follow the overall shape of the body, and how they become thicker near the mouth.

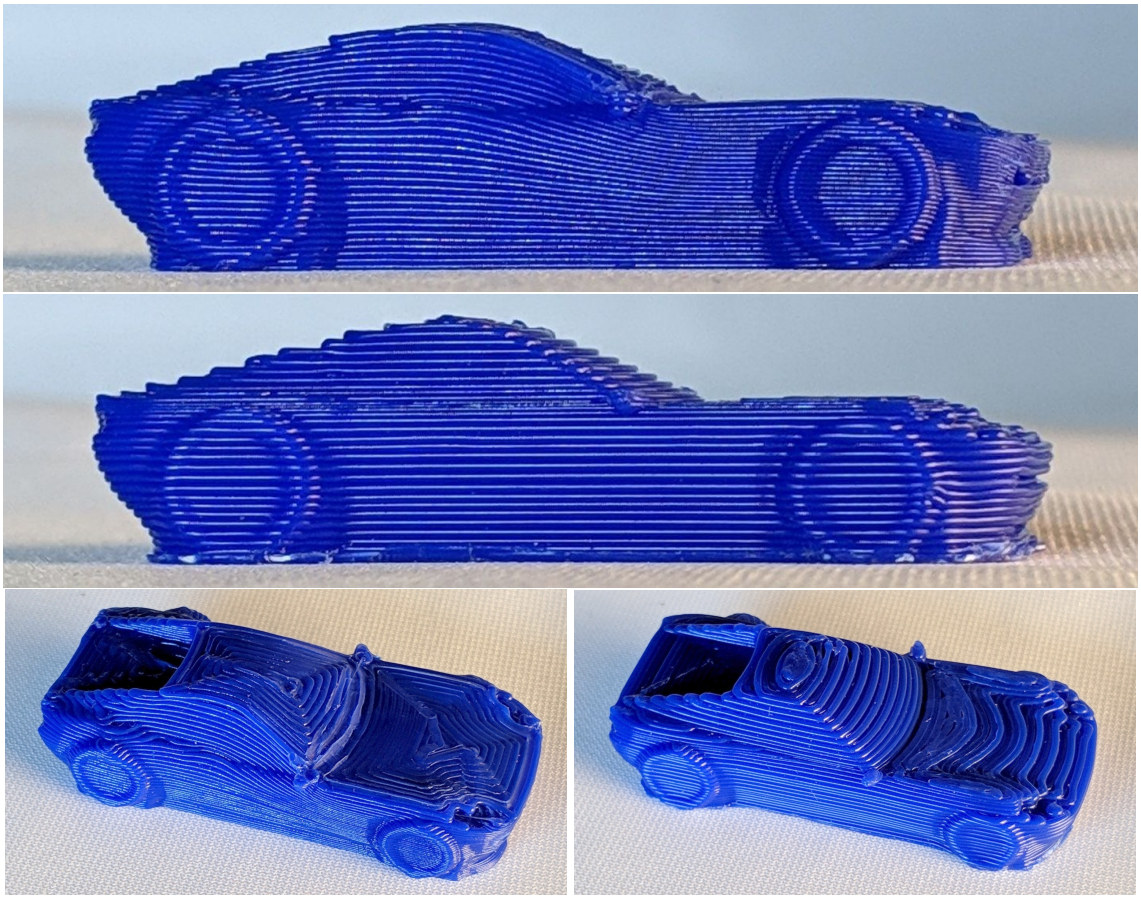


Figure 76: Sport car model printed on the UM2 using 34 layers, using our approach (top and then left) and adaptive slicing (bottom and then right). The two side views reveal the smooth outline of the curved layers version. Note the improved surface finish of the roof and hood, as well as the way the curved layers uncompress to produce the lateral windows. Adaptive slicing uses most layers around the hood.

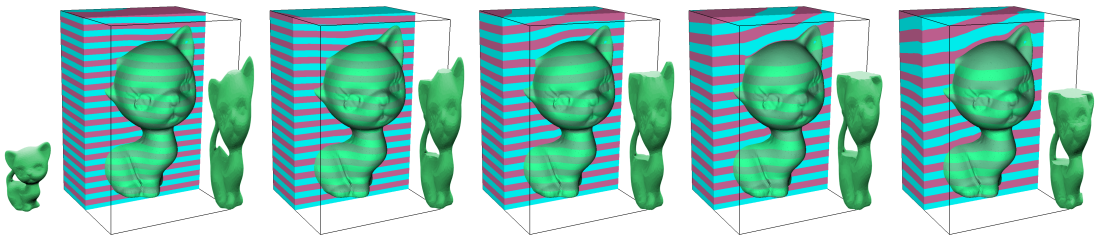


Figure 77: Optimization of kitten model for $\theta_{max} = 10$, 20, 30, 40, and 50 degrees. This shows both the object and the empty space around (cut open). The stripes correspond to slices. The 3D models shown inset are obtained after mapping $\mathcal{M}(\Omega)$. The original model is shown in the leftmost bottom corner. Note how additional angular freedom results in more areas being curved and smoothly reproduced.

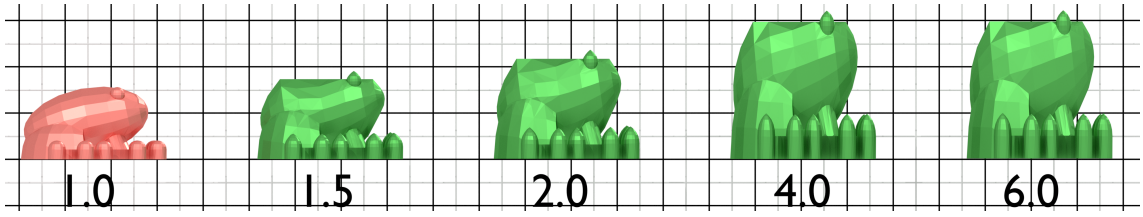


Figure 78: Optimization of the frog model for different min/max thickness ratios. The same surfaces are flattened, but the additional freedom leads to using thinner layers (taller models imply more layers). The red frog (ratio 1.0) is deemed unfeasible by the optimizer ; this is due to numerical issues as we request all slices to have exactly the same thickness.

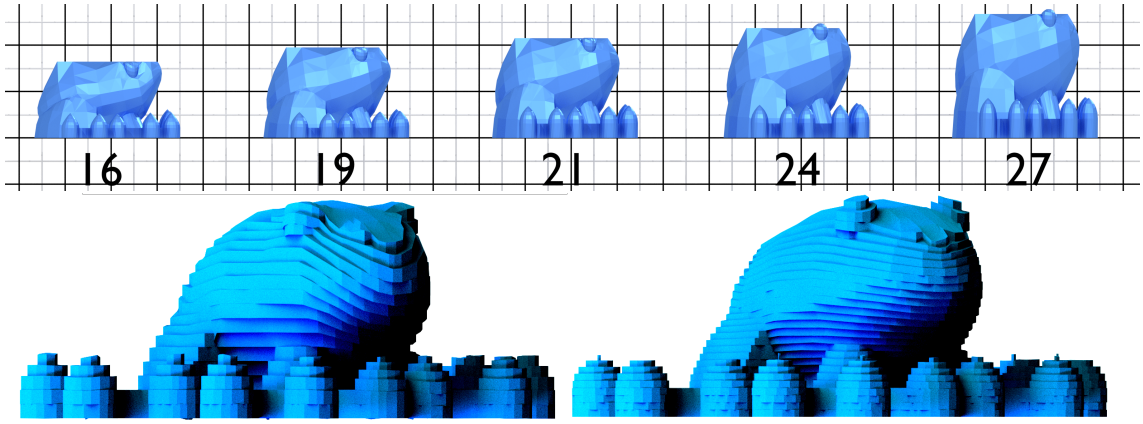


Figure 79: Optimization of the frog model targeting 16, 19, 21, 24, and 27 layers of 1mm, using the approach from Section 8.2.6. The renderings show the models for 16 and 27 slices. Lighting enhanced to highlight slices.

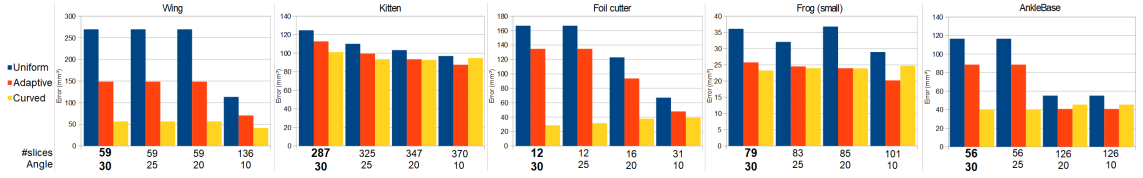


Figure 80: Comparison of uniform slicing (blue), optimal adaptive slicing (orange) and curved slicing (yellow) for five different models. We measure the volume error in mm^3. Lower is better. We run our approach first and then slice with uniform and adaptive slicing using the same number of slices. The result obtained by our method in the normal setup is the leftmost in each graph (angle = 30 degrees). The other columns show the result obtained when starting from smaller sets $\underline{\mathcal{F}}$, selecting only faces whose slopes are below the indicated angle (in degrees). Please refer to the text for discussion.

Figure 81: Different toolpath orientations for the top layers in curved slicing result in different surface finish. The visible stripes are shallower than the layer thickness, yet they create a visual effect that is detrimental to perceived quality. The effect also depends on lighting and view angle.

3D models

wing	https://www.thingiverse.com/thing:95502
anklebase (50% scale)	http://inmoov.fr/inmoov-stl-3d/?bodyparts=Legs-Ankle&parts=AnkleBaseV1.stl
frog (small)	https://www.thingiverse.com/thing:3284
frog (large)	https://www.thingiverse.com/thing:182144
kitten	https://www.thingiverse.com/thing:12694
sportscar	https://www.thingiverse.com/thing:1587558

8.4 Discussion and future work

Reducing the staircase defect was our main objective. We partially succeeded in this direction, even though the mapping/inverse mapping scheme still forced layers to appear on the boundaries where it would be preferable to have a unique layer on the surface.

Another limitation relates to the range of 3D printers where this technique applies. Some printers have very short extrusion nozzles; in such cases, there is very little angular freedom. Moreover, some machines have a low precision on the Z-axis or a heavy build plate, limiting the speed and precision of movement.

While this technique will improve appearance in most cases on 3-axis machines, curved printing results better when using higher DoF technologies to remove more printability constraints. At this point of my thesis, I wanted to continue this work with other objectives, such as mechanical ones, and use more degrees of freedom to curve the layers with fewer restrictions.

Curvislicer was released on Github with *GNU Affero General Public License* and has quickly gained some visibility. At this time, it has been nearly forked 20 times, has 150 stars, and some videos made by hobbyists can be found online. This version uses OSQP, an open-source solver, instead of Gurobi, leading to some artifacts not present in the paper implementation.

This works also inspired the community, as a few months after this publication, Fang et al. [2020] tackled the two objectives at once (other objective and 5DoF) using a similar idea of constraints over a tetrahedral mesh to optimize for different properties, but this time for a 5DoF printer. This work aims for structural strength, and they acknowledge that changing the objective function to the one in Curvislicer should improve surface accuracy. They used multiple objectives to obtain a printable object with a 5DoF printer.

Anisotropic strength, is the main objective. It adds an objective for every tetrahedron based on the stress distribution externally computed from the user-specified loads. The second objective, *layer thickness*, aims to smooth the layer height to keep it as constant as possible and as close as possible to the medial thickness. The medial thickness is defined to stay between the minimal and maximal printable layer height. More than layer thickness, gradients need to be smoothed (*harmonic of gradients*), as regions with significant variation will exhibit highly curved or even closed isosurfaces, causing a potential collision when printing. When optimizing, an *ambiguity* may rise as there are two directions possible for the gradient. Furthermore, if every tetrahedron is constrained, it will result in an overconstrained problem. So there are some *critical regions* selections so that not all the tetrahedrons will be constrained.

As they also tried to ensure fabricability, those objectives and constraints can find an equivalent in Curvislicer. The anisotropic strength was, in our case, the surface alignment; the layer thickness smoothness and harmonic of gradients were the same. The ambiguity was also present, and we forbid foldovers for the same purpose. Finally, they faced the same problem of over-contrition, solved with critical region selection.

In *CurviSlicer: Slightly curved slicing for 3-axis printers* and *Reinforced FDM: Multi-axis filament alignment with controlled anisotropic strength*, the slicing scheme was similar and had two major practical issues: it needs a tetrahedral mesh, and it has to be optimized. We start with a triangular mesh that has to be transformed into a tetrahedral mesh. It is

suitable for this tetrahedral mesh to have the least number of tetrahedrons, as it directly impacts the optimization time. Having a decent number of tetrahedrons is a heavy process and can add some error with respect to the triangular input mesh. Then, the optimization is performed on the tetrahedral mesh and minimizes quadratic objectives under linear constraints (QP). This operation can be prolonged and is done multiple times because of the iterative relaxation of the constraints. In addition, a layer will forcefully be traversal because of the mapping. We must revise this layered way of slicing to go further in curved printing.

The following approach is trying to avoid that. As the requirements over a parametrization are entirely removed from *Procedural band pattern* in *Staggered infill* thanks to a phasor noise, we apply the same idea to layers. Instead of computing layers in a parametric domain, let us define an oscillating signal in space to extract the layers and trajectories.

Chapter 9

Another brick in the wall: Orientable trajectories for additive manufacturing

Fabrication constraints in multi-axis printing are essentially the same as in standard printing. However, special care must be taken to avoid collisions of the machine itself or with the printed object. The most relevant constraint for multi-axis printing is that the material must be supported by previous paths or by additional structures. Now that the tool head can move freely in space, we hope to reduce the need for these structures.

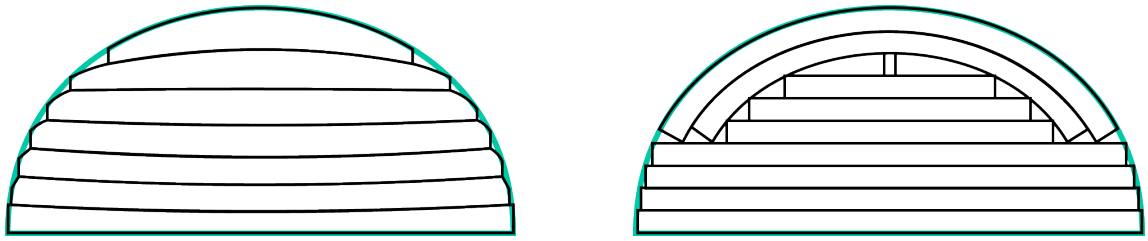
This work is an extension of the previous Chapter “Adaptive infills” Chapter 7, “A brick in the wall: Staggered orientable infills for additive manufacturing” [p.91] to curved printing

9.1 Multi-axis print constraints

The primary constraint when printing curves is to ensure there is no collision. Note that a collision can occur while printing a same trajectory. We can represent the plausible collision space with a collision cone, similarly to Chapter “Curved printing” Chapter 8, “CurviSlicer: Slightly curved slicing for 3-axis printers” [p.115], but it only encapsulates the tool head. Therefore, depending on the printing system used, it is better to simulate the full movements while checking collisions – or for robotic arms, to control the different axis and not by using positions and orientations of the tool. This ensures the absence of collision but is expensive.

When using multi-axis machines, some points in the print space requires special care, especially with a robotic arm. Those particular points are called singularities and occurs where a slight movement of the nozzle implies abrupt movement over multiple axes. The problem with those singularities is that they can block a continuous movement from being achieved or that the position of the different axis may lead to a robot part colliding the printed object or surroundings.

In this chapter, we ignore the issues regardind the singularities and axis controls to only focus on generating trajectories that will not collide if we only consider the tool head.



(a) Full curved layers *(b) Partial layers*

Figure 82: A comparison between the two available layer schemes on curved printing. ***Left:*** *Fully-traversal layers. They may vary in height to better approximate the surface, but they must cross the entire domain (e.g., Curvislicer).* ***Right:*** *the partial layering allows adding non-fully covering layers, allowing, for example, to have curved layers with constant deposition height.*

9.2 Our approach

Our technique takes as an input a 3D model and two direction fields d^U and d^V for which to generate 3D trajectories. The orientation constraints are not taken into account by our algorithm, so the input fields must be constrained. The algorithm defines three crossed phasor fields used to define oscillating patterns everywhere throughout the object volume. One defines the *partial layers* (see Figure 82), while the other two are used for the path extraction. The oscillation frequency f is chosen to match the target trajectory spacing – typically the nozzle width w (with $f^U = 1/w$) for d^U and d^V, and the layer height h (with $f^W = 1/h$) for d^W. However, we naturally support varying deposition widths or layer height by varying the frequency, see subsection 9.2.1. The direction of the phasor fields closely follows d^U.

Once the phasor fields are optimized, we proceed partial layer by partial layer to efficiently generate actual trajectories. This step extracts a set of discrete curved trajectories from the continuous definition of two phasor fields. We describe it in subsection 9.2.2.

9.2.1 Periodic variable fields

We define the three local periodic variable fields U, V, and W by relying on three orthogonal 3D phasor noises (see Equation 7.1). The kernel directions d_i are computed by sampling the input direction field d^U (respectively d^V and d^W) at the kernel position x_i. The frequencies f_i^U and f_i^V are chosen as $1/w$, with w the target deposition bead width while he frequencies f_i^W are chosen as $1/h$, with h the target deposition bead height. Note that w or h can be spatially varied by sampling an additional user's input field. Finally, parameter φ_i is computed by iterative phase alignment [Tricard, Tavernier, Zanni, Martínez, Hugron, Neyret and Lefebvre, 2020] to reduce the number of singularities in the generated field.

9.2.2 Defining trajectories

This section describes our method to create curved slices (and their associated trajectories) following the user's direction fields d^U and d^V, with d^V orthogonal to d^U. Let us define

d^W, a field orthogonal to d^U and d^V (i.e., $d^W = d^U \times d^V$). Note that d^U, d^V, and d^W are 3D fields. As a result, the trajectorie's direction evolve in space.

The layer and trajectory extraction is based on three orthogonal periodic variable fields, U, V, and W, obtained from three phasor fields. For any given point x in space we associate a tuple $UVW(x) \in [0, 2\pi]^2$. We note $U(x)$, $V(x)$ and $W(x)$ the three components of $UVW(x)$. In practice, $U(x)$ (respectively $V(x)$ and $W(x)$) define a locally oscillating signal taking the shape of a sawtooth. This signal defines a periodic variable field following the direction d^U (respectively d^V and d^W). From this periodic variable field we can construct a set of curves defined by $U(x) = \pi$ (respectively $V(x) = \pi$ and $W(x)$). The generated curves in C^U_π are orthogonal to d^U, (respectively C^V_π with d^V and C^W_π with d^W).

We define the layers as multiple sets of C^U_π and C^V_π curves and use C^W_π to order them. Therefore, we define the trajectories as the set of curves C^V_π . The orthogonal set of curves C^U_π is used during the computation of inter-trajectories distances and defines the sampling rate of the trajectories.

In practice, the extraction of the trajectories is performed in three steps. First, we define and optimize our periodic variable fields (see subsection 7.2.2). Then, we extract a graph whose vertices are defined as the intersection between C^U_π, C^V_π and C^W_π (see **??**, right). Edges of the graph are subdivided into three sets, E^U, E^V, and E^W. The edges of E^U (respectively E^V and E^W) form a tessellation of the C^U_π curves (respectively C^V_π and C^W_π). The graph extraction is described in subsection 7.2.3.

Finally, we use E^V to define the trajectories and E^U to measure their local spacing and create travel paths without collision. The E^V and E^W are later used to adjust the deposition flow (see subsection 7.2.4).

9.2.3 Graph extraction from a phasor field

Switching from 2D extraction to 3D implies some changes. First, instead of a marching square algorithm, we use a marching cube (subsection 9.2.3). Then, we combine the three different extractions (one for each field) into one graph. As the resulting graph may have degree two vertices along edges which are not usefull for the rest of the process, we can do a graph simplification (subsection 9.2.3). Finally, we extract the layers and deposition paths from that graph (subsection 9.2.4).

Marching cube

While the previous implementation used a marching square algorithm to extract the 2D graph, we used a marching cubes algorithm [Lorensen and Cline, 1987] to obtain a 3D one. The marching cubes algorithm traverses the scalar field, taking eight points at a time (thus defining an imaginary cube or voxel), and determines which polygons to create (if any) to represent a portion of the isosurface contained in that cube. This algorithm creates an index into a precomputed array of all the 256 possible polygon configurations in a cube($2^8 = 256$), treating each scalar value as a bit in an 8-bit integer. If the scalar value is greater than the isosurface value (here, when the field value crosses 0), then the corresponding bit is set to 1; otherwise, it is set to 0. The final value after the 8-point test is the index of the correct polygon configuration in the precomputed array. Finally,

each vertex of the created polygons is placed at its final position along the cube's edge by linearly interpolating the two scalar values connected by this edge.

In our implementation, we kept only vertices and edges to extract the paths as our d^U, and d^V fields' frequency matches the deposition width. However, it would be possible to use the faces generated by d^W and generate the paths directly from the extracted layer surfaces.

Merging the marching cubes

The edges' extraction is done for every field, resulting in 3 different sets of edges. Suppose some cell contains two or three intersecting edges defined from distinct fields. In that case, they are subdivided, and a new vertex is created at the intersection point, resulting in four or six new connected edges.

As each grid cell is processed, we stitch all the partial results together. To do so, we stitch vertices sharing the same grid edge or surface, creating the graph $\mathcal{G}$.

Graph simplification

The obtained graph contains a lot of unnecessary edges and vertices, and the first step of the extraction is to simplify this graph.

Notation For any graph $\mathcal{G}$ extracted from the marching cubes, we use $\mathcal{V}$ and $\mathcal{E}$ to denote its vertices and edges. Individual edges are expressed as a pair of vertices, e.g., $ij \in \mathcal{E}$ denotes the edge with vertices $i, j \in \mathcal{V}$. For any edge ij, $\overrightarrow{ij}$ defines the vector between the vertices i and j.

An expression of the form $a_i = \sum_{ij \in \mathcal{E}} b_{ij}$ indicates that a value a_i associated with vertex i is obtained by summing the quantity b_{ij} over all edges incident on i. It can be extended to edges in the same form, e.g. $c_{ij} = b_i + b_j$.

To have a facilitated acces to incident edges from a vertex, consider $edges_i$ the list of all incident edges to i, defined by $edges_i = edges_i \cup ij \quad \forall j \in \mathcal{V}, ij \in \mathcal{E}$. Every list member can be accessed using $edges_i[a]$, with a the position in the list. For readibility, $above_i$ and $below_i$ refers to the vertices extracted from d^W following in the input field direction for $above$ vertices, and on the opposite direction for $below$ vertices.

Algorithm The degree of a vertex is equal to the number of connected edges and is defined by $Degree(\mathcal{G}, i) = \sum_{ij \in \mathcal{E}} 1$. Any vertex i with $Degree(\mathcal{G}, i) = 0$ can then be safely removed from the graph $\mathcal{G}$, as they are not connected. Then, to reduce the number of vertices further, if $Degree(\mathcal{G}, i) = 2$ and the angle between the two adjacent edges are small enough, the two edges can collapse into one, removing the point. This simplification is constrained by the sampling, as a vertex of a higher degree than two is highly plausible to be space by approximately a nozzle width.

Algorithm 2 Decimation of unnecessary vertices and edges

procedure DECIMATION($\mathcal{G}$) ▷ $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  for all $i \in \mathcal{V}$ **do**
    if DEGREES($\mathcal{G}, i$) $= 0$ **then** ▷ isolated point
      Remove i from $\mathcal{V}$
    else if DEGREES($\mathcal{G}, i$) $= 2$ **then** ▷ i is connected to two edges
      $ji \leftarrow edges_i[0]$
      $ik \leftarrow edges_i[1]$
      if $\frac{\vec{ji} \cdot \vec{ik}}{\|\vec{ji}\| \|\vec{ik}\|} < \epsilon$ **then** ▷ low curvature point
        Remove i from $\mathcal{V}$
        Remove ji and ik from $\mathcal{E}$
        Insert jk in $\mathcal{E}$
      end if
    end if
  end for
end procedure

9.2.4 Layer extraction

Now that the graph is cleaned, we need to extract the different partial layers. In order to be printable, every path inside that layer should be printable without collision and have material supproting it from below.

Every layer is obtained by a growth algorithm. Starting from a given seed, it expands as much as possible while enforcing some constraints. The first constraint is directly related to printability. If the vertex below the one we want to add to the layer is not processed, then we cannot add this vertex as it will prevent the one below from being printed. Then, if the current vertex can be used in this layer, we can add its neighboring vertex if they are valid. A valid neighbor is forcefully originating from d^U or d^V and not in the "to process" queue or already processed. This strategy is repeated until every vertice is treated (see Algorithm 3).

The algorithm to find a good seed is similar to the one above. However, instead of adding the valid vertex to the processed set, we add a tag on it so that we know we already visited it. The algorithm starts by choosing the lowest point as a possible seed (any point can be used, but it converges faster starting from a plausible seed). Then, a growth process is done, tagging the vertex, and if ever a vertex below the current one was not previously processed or tagged, the process restarts starting from that "below" vertex (see Algorithm 4).

9.2.5 Path ordering

As seen in previous work, ordering paths efficiently is a complex problem. In curved printing, the issue is not only ordering the paths, but also generate adequate travel movements such they do not collide with previously deposited material.

Algorithm 3 Extract a set of ordered layers

function LAYEREXTRACTION($\mathcal{G}$) ▷ $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  $layers \leftarrow$ LIST() ▷ list of layers
  $P \leftarrow$ SET() ▷ set of Processed vertices
  $Q \leftarrow$ QUEUE() ▷ queue of vertices to process
  while $\mathcal{V} - P \neq \emptyset$ **do**
    $layer \leftarrow$ SET()
    PUSH(Q, FINDSEED($\mathcal{G}, P$)) ▷ Push a valid seed in the queue
    while $Q \neq \emptyset$ **do**
      $i \leftarrow$ FRONT(Q)
      POP(Q)
      if $below_i = \emptyset$ OR $below_i \in P$ **then** ▷ below point already processed
        INSERT($layer, i$)
        INSERT(P, i)
        for all $ij \in edges_i$ **do** ▷ for all edges connected to i
          if $j \notin P \cup Q \cup \{below_i\} \cup \{above_i\}$ **then**
            PUSH(Q, j) ▷ Adds unprocessed neighbors to the queue
          end if
        end for
      end if
    end while
    INSERT($layers, layer$)
  end while
  return $layers$
end function

Algorithm 4 Finds a new seed to start from, prioritizing lower vertices

```
1: function FindSeed(G, P)                                   ▷ G = (V, E)
2:     seed ← LowestPoint(G \ P)                  ▷ choose the lowest vertex as a seed
3:     found ← false
4:     while not found do
5:         blacklist ← Set()
6:         Q ← Queue()                                  ▷ queue of vertices to process
7:         Push(Q, seed)
8:         while Q ≠ ∅ do
9:             i ← Front(Q)
10:            Pop(Q)
11:            Insert(blacklist, i)
12:            if ∃ below_i then
13:                if below_i ∈ blacklist then          ▷ below point is already visited
14:                    continue                         ▷ so current must be ignored
15:                else if below_i ∉ P then             ▷ below point hasn't been processed
16:                    seed ← below_i                   ▷ it is then a better seed
17:                    break                            ▷ restart the process with the new seed
18:                end if
19:            end if
20:            for all ij ∈ edges_i do
21:                if j ∉ P ∪ Q ∪ blacklist ∪ {below_i} ∪ {above_i} then
22:                    Push(Q, j)          ▷ Adds unprocessed neighbors to the queue
23:                end if
24:            end for
25:        end while
26:        found ← true                                 ▷ Q is empty, return the current seed
27:    end while
28:    return seed
29: end function
```

Travel paths

All the paths must be generated such that they will never have collisions. For printing paths, it is ensured by the extraction of the paths, but for travel paths, if we go straight from a path endpoint to the start point of the subsequent one, there is a high probability of collision. We can work with the graph G to ensure no collision occurs. If we remove a layer from the graph every time a layer is printed, then using an $A*$ algorithm to navigate from one point to another is guaranteed collision-free. But G may not be connexe. In this case, we can keep in memory the highest point (or a convex hull if fully curved) and only navigate on this surface.

9.2.6 Avoiding collisions

When using this technique to 3D print on a standard 3-axis machine, a good way to ensure no collision is to constrain the direction field not to exceed a specific angle variation relative to the bed. For multi-axis printing, as the carriage or the nozzle shape does not bound the limit, the constraints over the bed are too restrictive. A way to ensure the object is printable is to, instead of limiting the angle to the bed, we can limit it with respect to a convex hull. With a convex hull, we are guaranteed to have a printable orientation if the collision cone does not penetrate its volume. Then, the direction can be constrained to be a quasi-convex hull, with small concavities not exceeding the printable angle to remain printable.

9.3 Conclusion

This chapter presents an algorithm to sort paths into a printable instruction set. While this algorithm also works to extract and order trajectories directly without layers, the processing time is much longer (Table 9.1). The way the path is extracted does not consider the proximity of the new seed with the position of the last processed trajectory, leading to a complex ordering of the print paths and a lot of auxiliary travel movements to avoid collisions.

Another problem is related to the number of singularities in the field. Even if the part is close to being fully dense, the continuity of the trajectories is doomed by the holes in the printing space. To improve the quality of the paths, the phasor noise must be optimized with new criterions that minimize the number of singularities.

size (mm^3)	graph extraction (s)	layer extraction (s)	path extraction (s)
20	29.0	12.5	0.1
30	112.7	130.5	0.5
40	227.1	626.9	1.3
50	521.4	1998.4	2.7

Table 9.1: Computationnal time of the algorithm depending on the size of the domain. The graph extraction is mainly done over a GPU (here a RTX 3070FE), and the layer and path extraction are done on a single core @4.2GHz.

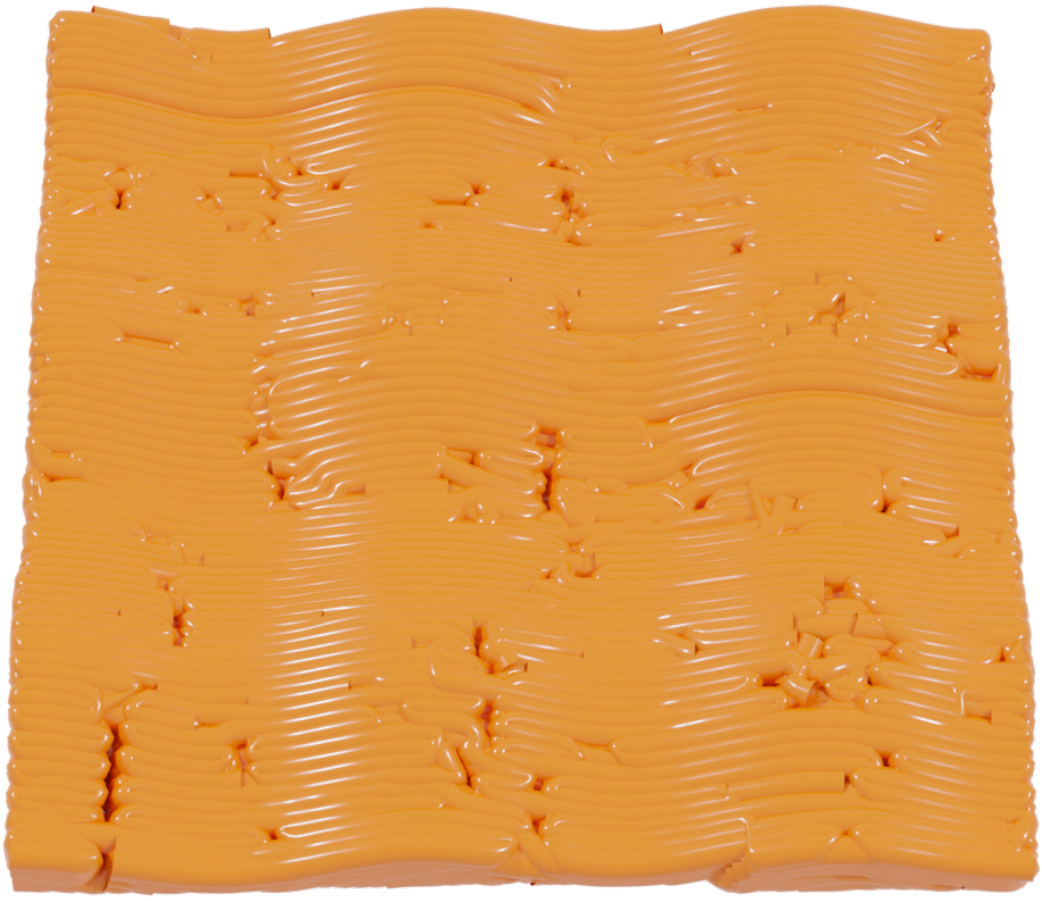


Figure 83: GCode generated with our technique. The visibility of the singularities is enhanced by the render, while when printed, the material tnds to flow and fill the gaps.

Discussion and future work

Curved Discussion We presented an algorithm to generate trajectories suitable for curved print by extending "A brick in the wall: Staggered orientable infills for additive manufacturing" [p.91]. While only the 3-axis constrained version was used to print the example object, using full abilities of multi-axis motion can be derived from our work as the constraints are set on the direction field. This algorithm's central part is extracting ordered layers and trajectories from a graph in a printable order.

1 Layer unstacking

Another way to see the problem of ordering layers would be to start from the entire shape and unstack layers instead of stacking layers above layers. If no portion of the layer we want to unstack is covered in any part, then we could remove it and repeat this until all the layers are extracted.

2 Layers from Procedural Band Pattern

In order to avoid having trouble sorting all the layers and ensuring we are generating the most extended traversing layer when possible, using "Procedural band patterns" [p.75] can be a reasonable choice. In addition, by using its property of identifying bands with growing identifiers, it ensures a correct layering as the bands are already sorted. Furthermore, the way the bands split gives the most extended layers as possible, as a new band only appears if necessary, and the distribution of appearing band is distributed following the split ratio such that there will be no singularity alignment. However, as previously stated, this approach requires a global parameterization which may not be trivial to generate. Once the layers are extracted, any other technique can be used for slicing (we may need to map the curved layer to a flat layer).

3 Layers based on a mapping

As stated in Chapter "CurviSlicer: Slightly curved slicing for 3-axis printers" subsection 8.3.4, "Discussion" [p.128], I believe that using mapping is a good idea. Rewriting the constraint of CurviSlicer to enforce convexity (or quasi-convexity) is a good way to minimize the number of layers and improve part strength "*Reinforced FDM: Multi-axis filament alignment with controlled anisotropic strength*".

Recently, other methods using isosurface extraction to generate the layers were used, Shan et al. [2021] using thermal diffusion and Li et al. [2022] a vector field; keeping the constraints of the layers nevertheless, which is a huge limiting factor.

Part V

Conclusion and perspectives

Conclusion

The main objective of this thesis was to find ways to do curved printing within slices and across layers to improve the overall quality, giving additional user control.

At first, we focused on controlling deposition paths inside a layer, with two new infill techniques offering additional freedom. Even though not directly applicable to curved layers, these works are highly related to our primary goal, as creating paths and ensuring printability under orientation constraints is not trivial. Then, we dived into the curved layer objective, where we first discussed the extraction of curved layers with CurviSlicer (for three-axis machines). Finally, we combined curved layers and controlled paths to print dense curved parts. It is worth noting that we first started using a parametrization in both the infills and the curved layer extraction. Then, we explored techniques that only need a direction field as input, thanks to phasor noise. In addition, all those methods can be used for any input shapes, making all those generic techniques and algorithms usable with every printer physically able to follow the generated paths. I now highlight some specific advantages of my different contributions.

First, in Part III, "Adaptive infills" [p.71], we focused on 2D infills to give more control to the user. In the first instance, we have empowered the user to control the filling density by avoiding the usual subdivision front effects. Indeed, the first method introduced, ***Procedural Band Patterns***, allows specifying the ratio at which each path will split. For example, with a ratio of 1 to 2, each filament will split into 2, so we find ourselves with the classical case of subdivision edges. However, if we use the ratio 2 out of 3, only one filament out of 2 will split, the ratio being the input path's number over the output path's number. Moreover, this first method allows to freely choose the orientation of the different paths, which is the main objective of the second approach. ***A brick in the wall: Staggered orientable infills for additive manufacturing*** generates dense layers while affording excellent orientation control. This method allows generating print paths according to a given orientation field while avoiding accumulating defects (such as holes inherent to dense filling) across layers. It differs from the previous method first by its aim at filling dense layers and second by its simplier settings as it does not require a parametrization of the space. The main interest of these methods is that they are uncorrelated to the application. It suffices to provide the control fields and parameters, such as the thickness of the deposit, which generates paths, which then facilitates the use of these techniques in combination with others, such as topological optimization.

Lastly, in Part IV, "Curved printing" [p.111], we introduced the possibility of improving the surface accuracy. These curved printing methods work with any extrusion printer and remarkably well with standard 3-axis machines. Obviously, in the case of 3-axis machines, the curvature is restricted by the machine's capabilities (shape and clearance of the nozzle, tool, and carriage). All those constraints are taken in account in ***CurviSlicer: Slightly curved slicing for 3-axis printers***. This method can be combined with all the current slicers, as it pre-processes (deforms) the mesh to be sliced by a standard slicer. Then the obtained GCode is post-processed (deformed back) to be curved, and material flow and printing speed are adjusted accordingly to ensure a constant feed rate of the extrusion motor. Our method allows for higher quality surfaces with a slight increase in printing time. Nevertheless, we consider this additional print time negligible, as the time required

to print an object approximating the same surface roughness will be much more extensive (as small slopes will require the smallest layers possible). In ***Another brick in the wall: Orientable trajectories for additive manufacturing***, we outlined a new method to generate curved paths within a volume using a Phasor noise. We obtained more control over layers and path deposition by circumventing the need for a global parametrization.

Outlooks

The results obtained during this research allowed us to glimpse the complexity of the subject of Curved slicing for additive manufacturing. In order to complete this work and tend towards industrialized solutions, many issues remain to be unraveled.

One of the main areas of research to pursue would be the extraction of less constrained paths in space. In all the previous work, the concept of layers is still overly present, although, in the extraction of curved paths from a Phasor noise, the layers were not fully traversing like in CurviSlicer. Another exception is ***Volumetric covering print-paths for additive manufacturing of 3D models*** by Ezair et al. [2018], but this work led to many small segments. As seen previously, small segments tend to lessen structural strength, so continuous deposition, or at least long paths, are preferred.

Removing the need for layers might be the best way to overcome the primary defects of 3D printing. The layering scheme is used to simplify the problem by reducing its degrees of freedom, but it significantly limits the application. Maybe it is time to look further and relax the constraints we artificially add to the modelization of the problem. A way to enter this problem is to use additional pieces of information given by the user or by the model itself. User information can be, as in ***Another brick in the wall: Orientable trajectories for additive manufacturing***, given by some input fields. Extracting paths, without any layer involved, would be possible by revisiting this algorithm. The fields can then be generated to match the user's requirements, e.g., surface quality or part strength. In addition, other information can be directly extracted from the model, such as medial axes or specialized models directly generated from, e.g., skeletons. In the recent approaches going toward this direction, the medial axes are presently used to generate paths inside a layer [Ding et al., 2016; Wang et al., 2021]. Hobbyists are also doing experiments, creating multi-axis machines to test out curved printing, but the objects generated are simple enough that the algorithms do not have to consider complex collisions. Avoiding collision is done either locally, ensuring a layer is printable on top of the previous one, or using a decomposition scheme [Mitropoulou et al., 2022].

At the speed at which 3D printing is developing in industry and the home, it should not be long before we see multi-axis manufacturing tools appearing massively among 3D printing hobbyists. As it stands, there is still no viable and easily usable solution for curved printing. But we have hope that new methods of slicing and adapted tools and control interfaces will appear in the coming years.

Bibliography

Ahlers, D., Wasserfall, F., Hendrich, N. and Zhang, J. [2019], 3d printing of nonplanar layers for smooth surface generation, *in* '2019 IEEE 15th international conference on automation science and engineering (CASE)', IEEE, pp. 1737–1743. 58, 59

Ahn, S.-H., Montero, M., Odell, D., Roundy, S. and Wright, P. K. [2002], 'Anisotropic material properties of fused deposition modeling abs', *Rapid prototyping journal* . 17

Alexa, M., Hildebrand, K. and Lefebvre, S. [2017], 'Optimal discrete slicing', *ACM Trans. Graph.* **36**(1), 1 – 16. 38, 127, 128, 129

Alexander, P., Allen, S. and Dutta, D. [1998], 'Part orientation and build cost determination in layered manufacturing', *Computer-Aided Design* **30**(5), 343–356. 40

Allen, S. and Dutta, D. [1995], 'Determination and evaluation of support structures in layered manufacturing', *Journal of Design and Manufacturing* **5**, 153–162. 40

Alsoufi, M. S., Alhazmi, M. W., Suker, D. K., Alghamdi, T. A., Sabbagh, R. A., Felemban, M. A. and Bazuhair, F. K. [2019], 'Experimental characterization of the influence of nozzle temperature in fdm 3d printed pure pla and advanced pla+', *American Journal of Mechanical Engineering* **7**(2), 45–60. 42

Alsoufi, M. S. and Elsayed, A. [2017], 'Warping deformation of desktop 3d printed parts manufactured by open source fused deposition modeling (fdm) system', *Int. J. Mech. Mechatron. Eng* **17**, 7–16. 41

Alvarez C, K. L., Lagos C, R. F. and Aizpun, M. [2016], 'Investigating the influence of infill percentage on the mechanical properties of fused deposition modelled abs parts', *Ingeniería e Investigación* **36**(3), 110–116. 48

An, B., Tao, Y., Gu, J., Cheng, T., Chen, X., Zhang, X., Zhao, W., Do, Y., Takahashi, S., Wu, H.-Y. et al. [2018], Thermorph: Democratizing 4d printing of self-folding materials and interfaces, *in* 'Proceedings of the 2018 CHI conference on human factors in computing systems', pp. 1–12. 46, 73, 95

An, H.-C., Kleinberg, R. and Shmoys, D. B. [2015], 'Improving christofides' algorithm for the st path tsp', *Journal of the ACM (JACM)* **62**(5), 1–28. 25

André, J., Le Mehaute, A. and De Witte, O. [1984], 'Dispositif pour réaliser un modèle de pièce industrielle', *French patent* **8411241**. 18

Ashby, M. F. [2006], 'The properties of foams and lattices', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **364**(1838), 15–30. 52

Attene, M. [2015], 'Shapes in a box: Disassembling 3d objects for efficient packing and fabrication', *Comput. Graph. Forum* **34**(8), 64–76. 40

Attene, M., Campen, M. and Kobbelt, L. [2013], 'Polygon mesh repairing: An application perspective', *ACM Computing Surveys (CSUR)* **45**(2), 1–33. 23

Bächer, M., Whiting, E., Bickel, B. and Sorkine-Hornung, O. [2014], 'Spin-it: Optimizing moment of inertia for spinnable objects', *ACM Transactions on Graphics (TOG)* **33**(4), 1–10. 49, 51

Barile, C., Casavola, C. and Cazzato, A. [2018], 'Acoustic emissions in 3d printed parts under mode i delamination test', *Materials* **11**(9), 1760. 41

Baumann, F. W. and Roller, D. [2017], Watermarking for fused deposition modeling by seam placement, *in* 'MATEC Web of Conferences', Vol. 104, EDP Sciences, p. 02023. 42

Bedel, A., Coudert-Osmont, Y., Martínez, J., Nishat, R. I., Whitesides, S. and Lefebvre, S. [2022], Closed space-filling curves with controlled orientation for 3d printing, *in* 'Computer Graphics Forum', Vol. 41, Wiley Online Library, pp. 473–492. 24, 47

bold design [2016], 'Collection poilu, aybar gallery, miami'. https://bold-design.fr/lab/poilu. 61

Brischetto, S., Ciano, A. and Ferro, C. G. [2016], 'A multipurpose modular drone with adjustable arms produced via the fdm additive manufacturing process', *Curved and Layered Structures* **3**(1). 15

Brooks, B. J., Arif, K. M., Dirven, S. and Potgieter, J. [2017], 'Robot-assisted 3d printing of biopolymer thin shells', *The International Journal of Advanced Manufacturing Technology* **89**(1), 957–968. 62, 63

Budarapu, P., YB, S. S. and Natarajan, R. [2016], 'Design concepts of an aircraft wing: composite and morphing airfoil with auxetic structures', *Frontiers of Structural and Civil Engineering* **10**(4), 394–408. 59

Buj-Corral, I., Domínguez-Fernández, A. and Durán-Llucià, R. [2019], 'Influence of print orientation on surface roughness in fused deposition modeling (fdm) processes', *Materials* **12**(23), 3834. 40

Carneiro, O. S., Silva, A. and Gomes, R. [2015], 'Fused deposition modeling with polypropylene', *Materials & Design* **83**, 768–776. 17

Castelino, K., D'Souza, R. and Wright, P. K. [2003], 'Toolpath optimization for minimizing airtime during machining', *Journal of manufacturing systems* **22**(3), 173–180. 25

Chakraborty, D., Aneesh Reddy, B. and Roy Choudhury, A. [2008], 'Extruder path generation for curved layer fused deposition modeling', *Comput. Aided Des.* **40**(2), 235–243. 62, 85

Chalsani, K., Jones, L. and Roscoe, L. [1995], Support generation for fused deposition modeling, *in* '1995 International Solid Freeform Fabrication Symposium'. 39

Chan, F. [2017], 'A high resolution full color 3d printer'. https://www.kickstarter.com/projects/517014753/the-world-first-high-resolution-full-color-3d-prin. 34

Chen, H., Yang, X., Chen, L., Wang, Y. and Sun, Y. [2016], 'Application of fdm three-dimensional printing technology in the digital manufacture of custom edentulous mandible trays', *Scientific reports* **6**(1), 1–6. 15

Chen, L., Chung, M.-F., Tian, Y., Joneja, A. and Tang, K. [2019], 'Variable-depth curved layer fused deposition modeling of thin-shells', *Robotics and Computer-Integrated Manufacturing* **57**, 422–434. 61, 62

Chen, Y., Li, K. and Qian, X. [2013], 'Direct geometry processing for telefabrication', *Journal of computing and information science in engineering* **13**(4). 39

Cheng, W., Fuh, J., Nee, A., Wong, Y., Loh, H. and Miyazawa, T. [1995*a*], 'Multi-objective optimization of part- building orientation in stereolithography', *Rapid Prototyp J.* **1**(4), 12–23. 40

Cheng, W., Fuh, J., Nee, A., Wong, Y., Loh, H. and Miyazawa, T. [1995*b*], 'Multi-objective optimization of part-building orientation in stereolithography', *Rapid Prototyping Journal* . 40

Childs, T., Hauser, C. and Badrossamay, M. [2005], 'Selective laser sintering (melting) of stainless and tool steel powders: experiments and modelling', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **219**(4), 339–357. 16

Christofides, N. [1976], Worst-case analysis of a new heuristic for the travelling salesman problem, Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group. 25

Comminal, R., Serdeczny, M. P., Pedersen, D. B. and Spangenberg, J. [2019], 'Motion planning and numerical simulation of material deposition at corners in extrusion additive manufacturing', *Additive Manufacturing* **29**, 100753. 24

Crump, S. S. [1991], Fused deposition modeling (fdm): putting rapid back into prototyping, *in* 'The 2nd International Conference on Rapid Prototyping. Dayton, Ohio', pp. 354–357. 17

Cuan-Urquizo, E., Espinoza-Camacho, J. I., Álvarez-Trejo, A., Uribe, E., Treviño-Quintanilla, C. D., Crespo-Sánchez, S. E., Gómez-Espinosa, A., Roman-Flores, A. and Olvera-Silva, O. [2021], 'Elastic response of lattice arc structures fabricated using curved-layered fused deposition modeling', *Mechanics of Advanced Materials and Structures* **28**(14), 1498–1508. 58

Cuan-Urquizo, E., Martínez-Magallanes, M., Crespo-Sánchez, S. E., Gómez-Espinosa, A., Olvera-Silva, O. and Roman-Flores, A. [2019], 'Additive manufacturing and mechanical properties of lattice-curved structures', *Rapid Prototyping Journal* . 58, 61

Dai, C., Wang, C. C., Wu, C., Lefebvre, S., Fang, G. and Liu, Y.-J. [2018], 'Support-free volume printing by multi-axis motion', *ACM Transactions on Graphics (TOG)* **37**(4), 1–14. 62, 64, 65, 113

Deckard, C. R. [1988], Selective laser sintering, PhD thesis, The University of Texas at Austin. 16

Dehaeck, S., Scheid, B. and Lambert, P. [2018], 'Adaptive stitching for meso-scale printing with two-photon lithography', *Additive Manufacturing* **21**, 589–597. 24

DeWerd, L. A. and Kissick, M. [2014], *The phantoms of medical and health physics*, Springer. 49

Di Angelo, L., Di Stefano, P., Dolatnezhadsomarin, A., Guardiani, E. and Khorram, E. [2020], 'A reliable build orientation optimization method in additive manufacturing: the application to fdm technology.', *International Journal of Advanced Manufacturing Technology* **108**. 40

Diegel, O., Singamneni, S., Chowdhury, R., Gibson, I. and Huang, B. [2010], 'Curved-layer fused deposition modelling', *Journal for New Generation Sciences* **8**(2), 95–107. 62

Diegel, O., Singamneni, S., Huang, B. and Gibson, I. [2011], Curved layer fused deposition modeling in conductive polymer additive manufacturing, *in* 'Advanced Materials Research', Vol. 199, Trans Tech Publ, pp. 1984–1987. 62

Dilberoglu, U. M., Simsek, S. and Yaman, U. [2019], 'Shrinkage compensation approach proposed for abs material in fdm process', *Materials and Manufacturing Processes* **34**(9), 993–998. 42

Ding, D., Pan, Z., Cuiuri, D., Li, H. and Larkin, N. [2016], 'Adaptive path planning for wire-feed additive manufacturing using medial axis transformation', *Journal of Cleaner Production* **133**, 942–952. 46, 154

Ding, D., Pan, Z. S., Cuiuri, D. and Li, H. [2014], 'A tool-path generation strategy for wire and arc additive manufacturing', *The international journal of advanced manufacturing technology* **73**(1), 173–183. 46

Dinh, H. Q., Gelman, F., Lefebvre, S. and Claux, F. [2015], Modeling and toolpath generation for consumer-level 3d printing, *in* 'ACM SIGGRAPH 2015 Courses', SIGGRAPH '15, pp. 17:1–17:273. 49

Diourté, A., Bugarin, F., Bordreuil, C. and Segonds, S. [2021], 'Continuous three-dimensional path planning (ctpp) for complex thin parts with wire arc additive manufacturing', *Additive Manufacturing* **37**, 101622. 60

Dolenc, A. and Mäkelä, I. [1994], 'Slicing procedures for layered manufacturing techniques', *Comput. Aided Des.* **26**(2), 119–126. 38

Dreifus, G., Goodrick, K., Giles, S., Patel, M., Foster, R. M., Williams, C., Lindahl, J., Post, B., Roschli, A., Love, L. et al. [2017], 'Path optimization along lattices in additive manufacturing using the chinese postman problem', *3D Printing and Additive Manufacturing* **4**(2), 98–104. 25

Drummer, D., Cifuentes-Cuéllar, S. and Rietzel, D. [2012], 'Suitability of pla/tcp for fused deposition modeling', *Rapid Prototyping Journal* . 17

Dumas, J., Hergel, J. and Lefebvre, S. [2014], 'Bridging the gap: Automated steady scaffoldings for 3d printing', *ACM Trans. Graph.* **33**(4), 98:1–98:10. 40

Efremov, S., Martínez, J. and Lefebvre, S. [2021], '3d periodic cellular materials with tailored symmetry and implicit grading', *Computer-Aided Design* **140**, 103086. 53

Elber, G. and Cohen, E. [1996], 'Adaptive isocurve-based rendering for freeform surfaces', *ACM Transactions on Graphics (TOG)* **15**(3), 249–263. 47, 49, 98

Etienne, J. and Lefebvre, S. [2020], Procedural band patterns, *in* 'Symposium on Interactive 3D Graphics and Games', pp. 1–7. 5, 98

Etienne, J., Ray, N., Panozzo, D., Hornus, S., Wang, C. C., Martínez, J., McMains, S., Alexa, M., Wyvill, B. and Lefebvre, S. [2019], 'Curvislicer: Slightly curved slicing for 3-axis printers', *ACM Transactions on Graphics (TOG)* **38**(4), 1–11. 6

Ezair, B., Fuhrmann, S. and Elber, G. [2018], 'Volumetric covering print-paths for additive manufacturing of 3d models', *Computer-Aided Design* **100**, 1–13. 59, 63, 85, 98, 99, 154

Ezair, B., Massarwi, F. and Elber, G. [2015], 'Orientation analysis of 3d objects toward minimal support volume in 3d-printing', *Computers & Graphics* **51**, 117–124. 40

Fang, G., Zhang, T., Zhong, S., Chen, X., Zhong, Z. and Wang, C. C. [2020], 'Reinforced fdm: Multi-axis filament alignment with controlled anisotropic strength', *ACM Transactions on Graphics (TOG)* **39**(6), 1–15. 46, 137

Feng, X., Cui, B., Liu, Y., Li, L., Shi, X. and Zhang, X. [2021], 'Curved-layered material extrusion modeling for thin-walled parts by a 5-axis machine', *Rapid Prototyping Journal* . 62

Fok, K.-Y., Ganganath, N., Cheng, C.-T. and Chi, K. T. [2016], A 3d printing path optimizer based on christofides algorithm, *in* '2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)', IEEE, pp. 1–2. 25

Frank, D. and Fadel, G. [1995], 'Expert system-based selection of the preferred direction of build for rapid prototyping processes', *Journal of Intelligent Manufacturing* **6**(5), 339–345. 40

Fry, N. R., Richardson, R. C. and Boyle, J. H. [2020], 'Robotic additive manufacturing system for dynamic build orientations', *Rapid prototyping journal* . 63

Ganganath, N., Cheng, C.-T., Fok, K.-Y. and Chi, K. T. [2016], Trajectory planning for 3d printing: A revisit to traveling salesman problem, *in* '2016 2nd International Conference on Control, Automation and Robotics (ICCAR)', IEEE, pp. 287–290. 25

Gao, F. and Sonin, A. A. [1994], 'Precise deposition of molten microdrops: the physics of digital microfabrication', *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* **444**(1922), 533–554. 16

Gao, W., Zhang, Y., Nazzetta, D. C., Ramani, K. and Cipra, R. J. [2015], RevoMaker: Enabling multi-directional and functionally-embedded 3D printing using a rotational cuboidal platform, *in* 'Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology', pp. 437–446. 64

Ghaiebi, H. and Solimanpur, M. [2007], 'An ant algorithm for optimization of hole-making operations', *Computers & Industrial Engineering* **52**(2), 308–319. 25

Gibson, I., Rosen, D. and Stucker, B. [2015], Directed energy deposition processes, *in* 'Additive manufacturing technologies', Springer, pp. 245–268. 18

Golomb, S. W. and Selfridge, J. L. [1970], 'Unicursal polygonal paths and other graphs on point lattices', *Pi Mu Epsilon Journal* **5**(3), 107–117. 26, 45

Gong, J. and Zhang, L. [2015], Genetic algorithm with unit processor applied in fused deposition manufacturing (fdm) for minimizing non-productive tool-path, *in* 'Industrial Engineering, Machine Design and Automation (IEMDA 2014) & Computer Science and Application (CCSA 2014) Proceedings of the 2014 Congress on IEMDA 2014 & Proceedings of the 2nd Congress on CCSA 2014', World Scientific, pp. 191–197. 25

Gosselin, C., Duballet, R., Roux, P., Gaudillière, N., Dirrenberger, J. and Morel, P. [2016], 'Large-scale 3d printing of ultra-high performance concrete–a new processing route for architects and builders', *Materials & Design* **100**, 102–109. 36

Gray, R. W., Baird, D. G. and Bøhn, J. H. [1998], 'Effects of processing conditions on short tlcp fiber reinforced fdm parts', *Rapid Prototyping Journal* . 46

Gregson, J., Sheffer, A. and Zhang, E. [2011], 'All-hex mesh generation via volumetric polycube deformation', *Comput. Graph. Forum* **30**(5), 1407–1416. 125

Groen, J. P., Wu, J. and Sigmund, O. [2019], 'Homogenization-based stiffness optimization and projection of 2d coated structures with orthotropic infill', *Computer Methods in Applied Mechanics and Engineering* **349**, 722–742. 47, 49, 75

Gurobi [2018], 'Gurobi optimizer reference manual, gurobi optimization, llc'. 124

Gurrala, P. K. and Regalla, S. P. [2012], Optimization of support material and build time in fused deposition modeling (fdm), *in* 'Applied Mechanics and Materials', Vol. 110, Trans Tech Publ, pp. 2245–2251. 40

Hayasi, M. T. and Asiabanpour, B. [2013], 'A new adaptive slicing approach for the fully dense freeform fabrication (fdff) process', *Journal of Intelligent Manufacturing* **24**(4), 683–694. 38

He, L., Peng, H., Lin, M., Konjeti, R., Guimbretière, F. and Froehlich, J. E. [2019], Ondulé: Designing and controlling 3d printable springs, *in* 'Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology', pp. 739–750. 59

Heidari-Rarani, M., Rafiee-Afarani, M. and Zahedi, A. [2019], 'Mechanical characterization of fdm 3d printing of continuous carbon fiber reinforced pla composites', *Composites Part B: Engineering* **175**, 107147. 46

Held, M. [1991], 'A geometry-based investigation of the tool path generation for zigzag pocket machining', *The Visual Computer* **7**(5), 296–308. 45

Held, M., Lukács, G. and Andor, L. [1994], 'Pocket machining based on contour-parallel tool paths generated by means of proximity maps', *Computer-Aided Design* **26**(3), 189–203. 45

Hergel, J. and Lefebvre, S. [2014], Clean color: Improving multi-filament 3d prints, *in* 'Computer Graphics Forum', number 2, Wiley Online Library, pp. 469–478. 35

Herholz, P., Matusik, W. and Alexa, M. [2015], 'Approximating free-form geometry with height fields for manufacturing', *Comput. Graph. Forum* **34**(2), 239–251. 41

Hertzmann, A. and Zorin, D. [2000], Illustrating smooth surfaces, *in* 'Proceedings of the 27th annual conference on Computer graphics and interactive techniques', pp. 517–526. 46

Hilbert, D. [1891], 'Über die stetige abbildung einer linie auf ein flächenstück'. 46

Hildebrand, K., Bickel, B. and Alexa, M. [2013], 'Orthogonal slicing for additive manufacturing', *Computers & Graphics* **37**(6), 669 – 675. Shape Modeling International (SMI) Conference 2013. 40

Hongyao, S., Xiaoxiang, Y. and Jianzhong, F. [2018], 'Research on the flexible support platform for fused deposition modeling', *The International Journal of Advanced Manufacturing Technology* **97**(9), 3205–3221. 63

Hope, R., Roth, R. and Jacobs, P. [1997], 'Adaptive slicing with sloping layer surfaces', *Rapid Prototyp J.* **3**(3), 89–98. 38

Hornbeck, L. J. [1996], 'Digital light processing™: A new mems-based display technology'. 18

Hornus, S., Kuipers, T., Devillers, O., Teillaud, M., Martínez, J., Glisse, M., Lazard, S. and Lefebvre, S. [2020], 'Variable-width contouring for additive manufacturing', *ACM Transactions on Graphics (TOG)* **39**(4), 131–1. 46

Hornus, S. and Lefebvre, S. [2017], Iterative carving for self-supporting 3D printed cavities, PhD thesis, Inria Nancy-Grand Est. 50, 51

Hu, R., Li, H., Zhang, H. and Cohen-Or, D. [2014], 'Approximate pyramidal shape decomposition', *ACM Trans. Graph.* **33**(6), 213:1–213:12. 40

Hu, Y., Zhou, Q., Gao, X., Jacobson, A., Zorin, D. and Panozzo, D. [2018], 'Tetrahedral meshing in the wild', *ACM Trans. Graph.* **37**(4), 60:1–60:14. 120

Huang, B. and Singamneni, S. [2012], 'Alternate slicing and deposition strategies for fused deposition modelling of light curved parts', *J. of Achievements in Materials and Manufacturing Engineering* **55**(2), 511–517. 129

Huang, B. and Singamneni, S. B. [2015], 'Curved layer adaptive slicing (clas) for fused deposition modelling', *Rapid Prototyping Journal* . 61

Huang, P., Wang, C. C. and Chen, Y. [2013], Algorithms for layered manufacturing in image space, PhD thesis, Chinese University of Hong Kong. 39

Huang, X., Ye, C., Mo, J. and Liu, H. [2009], 'Slice data based support generation algorithm for fused deposition modeling', *Tsinghua Science and Technology* **14**(S1), 223–228. 39

Huang, X., Ye, C., Wu, S., Guo, K. and Mo, J. [2009], 'Sloping wall structure support generation for fused deposition modeling', *The International Journal of Advanced Manufacturing Technology* **42**(11), 1074–1081. 39

Hull, C. W. [1986], 'Apparatus for production of three-dimensional objects by stereolithography'. US Patent 4,575,330. 18

Hussein, A., Hao, L., Yan, C., Everson, R. and Young, P. [2013], 'Advanced lattice support structures for metal additive manufacturing', *Journal of Materials Processing Technology* **213**(7), 1019–1026. 39

Hutmacher, D. W., Schantz, T., Zein, I., Ng, K. W., Teoh, S. H. and Tan, K. C. [2001], 'Mechanical properties and cell cultural response of polycaprolactone scaffolds designed and fabricated via fused deposition modeling', *Journal of Biomedical Materials Research: An Official Journal of The Society for Biomaterials, The Japanese Society for Biomaterials, and The Australian Society for Biomaterials and the Korean Society for Biomaterials* **55**(2), 203–216. 15

IceSL [2022], 'Icesl, advanced modeling and slicing for 3d printing'. 42, 85

Ingole, D., Deshmukh, T., Kuthe, A. and Ashtankar, K. [2011], 'Build orientation analysis for minimum cost determination in fdm', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **225**(10), 1925–1938. 40

Isa, M. A. and Lazoglu, I. [2019], 'Five-axis additive manufacturing of freeform models through buildup of transition layers', *Journal of Manufacturing Systems* **50**, 69–80. 57, 62

Jiang, J. [2020], 'A novel fabrication strategy for additive manufacturing processes', *Journal of Cleaner Production* **272**, 122916. 25

Jin, Y.-a., He, Y., Fu, J.-z., Gan, W.-f. and Lin, Z.-w. [2014], 'Optimization of tool-path generation for material extrusion-based additive manufacturing technology', *Additive manufacturing* **1**, 32–47. 24

Jin, Y., Du, J. and He, Y. [2017], 'Optimization of process planning for reducing material consumption in additive manufacturing', *Journal of Manufacturing Systems* **44**, 65–78. 46

Jin, Y., He, Y., Fu, G., Zhang, A. and Du, J. [2017], 'A non-retraction path planning approach for extrusion-based additive manufacturing', *Robotics and Computer-Integrated Manufacturing* **48**, 132–144. 24

Jobard, B. and Lefer, W. [1997], Creating evenly-spaced streamlines of arbitrary density, *in* 'Visualization in scientific computing'97', Springer, pp. 43–55. 46

Kao, J.-H. and Prinz, F. B. [1998], Optimal motion planning for deposition in layered manufacturing, *in* 'International Design Engineering Technical Conferences and Computers and Information in Engineering Conference', Vol. 80364, American Society of Mechanical Engineers, p. V006T06A018. 46

Keating, S. and Oxman, N. [2013], 'Compound fabrication: A multi-functional robotic platform for digital design and fabrication', *Robotics and Computer-Integrated Manufacturing* **29**(6), 439–448. 61

Kelly, B. E., Bhattacharya, I., Heidari, H., Shusteff, M., Spadaccini, C. M. and Taylor, H. K. [2019], 'Volumetric additive manufacturing via tomographic reconstruction', *Science* **363**(6431), 1075–1079. 18

Kirschman, C., Jara-Almonte, C., Bagchi, A., Dooley, R. and Ogale, A. [1991], Computer aided design of support structures for stereolithographic components, *in* 'International Design Engineering Technical Conferences and Computers and Information in Engineering Conference', Vol. 6227, American Society of Mechanical Engineers, pp. 443–448. 39

Klipper3D [2022], 'Klipper'. https://www.klipper3d.org/. 24

Klosterman, D. A., Chartoff, R. P., Osborne, N. R., Graves, G. A., Lightman, A., Han, G., Bezeredi, A. and Rodrigues, S. [1999], 'Development of a curved layer lom process for monolithic ceramics and ceramic matrix composites', *Rapid Prototyping Journal* . 61

Knöppel, F., Crane, K., Pinkall, U. and Schröder, P. [2015], 'Stripe patterns on surfaces', *ACM Transactions on Graphics (TOG)* **34**(4), 1–11. 47

Kruth, J.-P., Mercelis, P., Van Vaerenbergh, J., Froyen, L. and Rombouts, M. [2005], 'Binding mechanisms in selective laser sintering and selective laser melting', *Rapid prototyping journal* . 16

Kuipers, T., Doubrovski, E. L., Wu, J. and Wang, C. C. [2020], 'A framework for adaptive width control of dense contour-parallel toolpaths in fused deposition modeling', *Computer-Aided Design* **128**, 102907. 46

Kuipers, T., Elkhuizen, W., Verlinden, J. and Doubrovski, E. [2018], 'Hatching for 3d prints: Line-based halftoning for dual extrusion fused deposition modeling', *Computers & Graphics* **74**, 23–32. 35

Kulkarni, P. and Dutta, D. [1996], 'An accurate slicing procedure for layered manufacturing', *Comput. Aided Des.* **28**(9), 683 – 697. 38

Labelle, C. [2016], 'The palette: 3d printing evolved'. https://www.kickstarter.com/projects/mosaic3d/the-palette-3d-printing-evolved. 34

Lagae, A., Lefebvre, S., Cook, R., DeRose, T., Drettakis, G., Ebert, D. S., Lewis, J. P., Perlin, K. and Zwicker, M. [2010], A survey of procedural noise functions, *in* 'Computer Graphics Forum', Vol. 29, Wiley Online Library, pp. 2579–2600. 76

Lagae, A., Lefebvre, S., Drettakis, G. and Dutré, P. [2009], 'Procedural noise using sparse gabor convolution', *ACM Transactions on Graphics (TOG)* **28**(3), 1–10. 92

Lam, Y. and Santhikumar, S. [2003], 'Automated rib location and optimization for plate structures', *Structural and multidisciplinary optimization* **25**(1), 35–45. 50

Lee, K. and Jee, H. [2015], 'Slicing algorithms for multi-axis 3-d metal printing of overhangs', *Journal of Mechanical Science and Technology* **29**(12), 5139–5144. 64

Lefebvre, S. [2015], '3d infilling: faster, stronger, simpler'. http://sylefeb.blogspot.fr/2015/07/3dprint-3d-infilling-faster-stronger.html. 49, 84

Li, W., Zheng, A., You, L., Yang, X., Zhang, J. and Liu, L. [2017], Rib-reinforced shell structure, *in* 'Computer Graphics Forum', Vol. 36, Wiley Online Library, pp. 15–27. 50

Li, Y., He, D., Yuan, S., Tang, K. and Zhu, J. [2022], 'Vector field-based curved layer slicing and path planning for multi-axis printing', *Robotics and Computer-Integrated Manufacturing* **77**, 102362. 149

Lia [2014], 'Filament sculptures'. https://www.liaworks.com/theprojects/filament-sculptures/. 60

Lim, S., Buswell, R. A., Valentine, P. J., Piker, D., Austin, S. A. and De Kestelier, X. [2016], 'Modelling curved-layered printing paths for fabricating large-scale construction components', *Additive Manufacturing* . 129

Lim, T.-C. [2015], *Auxetic materials and structures*, Vol. 2779, Springer. 59, 61

LithophaneMaker [2022], 'Color lithophane maker'. https://www.lithophanemaker.com/ColorLithophane.html. 34

Liu, B., Cheng, H., Liu, M., Cao, W. and Jiang, K. [2021], 'Adaptive anisotropic porous structure design and modeling for 2.5 d mechanical parts', *Materials & Design* **206**, 109786. 48

Liu, Y., Pan, H., Snyder, J., Wang, W. and Guo, B. [2013], 'Computing self-supporting surfaces by regular triangulation', *ACM Transactions on Graphics (TOG)* **32**(4), 1–10. 39

Livesu, M., Ellero, S., Martínez, J., Lefebvre, S. and Attene, M. [2017], 'From 3D models to 3D prints: an overview of the processing pipeline', *Computer Graphics Forum* **36**(2). 21, 40

Lorensen, W. E. and Cline, H. E. [1987], 'Marching cubes: A high resolution 3d surface construction algorithm', *ACM siggraph computer graphics* **21**(4), 163–169. 141

Lu, L., Sharf, A., Zhao, H., Wei, Y., Fan, Q., Chen, X., Savoye, Y., Tu, C., Cohen-Or, D. and Chen, B. [2014], 'Build-to-last: Strength to weight 3d printed objects', *ACM Transactions on Graphics (ToG)* **33**(4), 1–10. 52

Luo, L., Baran, I., Rusinkiewicz, S. and Matusik, W. [2012], 'Chopper: Partitioning models into 3d-printable parts', *ACM Transactions on Graphics (TOG)* **31**(6), 1–9. 40

Luxner, M. H., Stampfl, J. and Pettermann, H. E. [2007], 'Numerical simulations of 3d open cell structures–influence of structural irregularities on elasto-plasticity and deformation localization', *International Journal of Solids and Structures* **44**(9), 2990–3003. 52

Ma, Q., Rejab, M., Kumar, A. P., Fu, H., Kumar, N. M. and Tang, J. [2020], 'Effect of infill pattern, density and material type of 3d printed cubic structure under quasi-static loading', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* p. 0954406220971667. 48

Ma, W., But, W.-C. and He, P. [2004], 'Nurbs-based adaptive slicing for efficient rapid prototyping', *Computer-Aided Design* **36**(13), 1309–1325. 38

Madamesila, J., McGeachy, P., Barajas, J. E. V. and Khan, R. [2016], 'Characterizing 3d printing in the fabrication of variable density phantoms for quality assurance of radiotherapy', *Physica Medica* **32**(1), 242–247. 49

Majhi, J., Janardan, R., Schwerdt, J., Smid, M. and Gupta, P. [1999], 'Minimizing support structures and trapped area in two-dimensional layered manufacturing', *Computational Geometry* **12**(3-4), 241–267. 40

Mani, K., Kulkarni, P. and Dutta, D. [1999], 'Region-based adaptive slicing', *Comput. Aided Des.* **31**(5), 317 – 333. 38

Maragos, P. and Schafer, R. [1986], 'Morphological skeleton representation and coding of binary images', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **34**(5), 1228–1244. 39

Martínez, J., Dumas, J. and Lefebvre, S. [2016], 'Procedural voronoi foams for additive manufacturing', *ACM Transactions on Graphics (TOG)* **35**(4), 1–12. 52

Martínez, J., Hornus, S., Song, H. and Lefebvre, S. [2018], 'Polyhedral voronoi diagrams for additive manufacturing', *ACM Transactions on Graphics (TOG)* **37**(4), 1–15. 52

Martínez, J., Skouras, M., Schumacher, C., Hornus, S., Lefebvre, S. and Thomaszewski, B. [2019], 'Star-shaped metrics for mechanical metamaterial design', *ACM Transactions on Graphics (TOG)* **38**(4), 1–13. 53, 59

Martínez, J., Song, H., Dumas, J. and Lefebvre, S. [2017], 'Orthotropic k-nearest foams for additive manufacturing', *ACM Transactions on Graphics (TOG)* **36**(4), 1–12. 52

Masood, H. S., Rattanawong, W. and Iovenitti, P. [2000], 'Part build orientations based on volumetric error in fused deposition modelling', *The International Journal of Advanced Manufacturing Technology* **16**(3), 162–168. 38

MateRiel [2022], 'The swapper3d: No more purge blocks!'. https://www.kickstarter.com/projects/bigbrain3d/swapper3d. 34

Mazumder, J., Choi, J., Nagarathnam, K., Koch, J. and Hetzner, D. [1997], 'The direct metal deposition of h13 tool steel for 3-d components', *Jom* **49**(5), 55–60. 18

McCaw, J. C. and Cuan-Urquizo, E. [2018], 'Curved-layered additive manufacturing of non-planar, parametric lattice structures', *Materials & Design* **160**, 949–963. 58, 61

Mebarki, A., Alliez, P. and Devillers, O. [2005], Farthest point seeding for efficient placement of streamlines, *in* 'VIS 05. IEEE Visualization, 2005.', IEEE, pp. 479–486. 46

Milewski, J., Lewis, G., Thoma, D., Keel, G., Nemec, R. and Reinert, R. [1998], 'Directed light fabrication of a solid metal hemisphere using 5-axis powder deposition', *Journal of Materials Processing Technology* **75**(1-3), 165–172. 18

Mitropoulou, I., Bernhard, M. and Dillenburger, B. [2022], 'Nonplanar 3d printing of bifurcating forms', *3D Printing and Additive Manufacturing* **9**(3), 189–202. 154

Moore, E. H. [1900], 'On certain crinkly curves', *Transactions of the American Mathematical Society* **1**(1), 72–90. 46

Moradi, M., Meiabadi, S. and Kaplan, A. [2019], '3d printed parts with honeycomb internal pattern by fused deposition modelling; experimental characterization and production optimization', *Metals and Materials International* **25**(5), 1312–1325. 48

Moraru, E., Dontu, O., Besnea, D. and Constantin, V. [2018], Study and realization of prosthetic dental models by additive technologies, *in* 'IOP Conference Series: Materials Science and Engineering', Vol. 444, IOP Publishing, p. 042017. 15

Mosaic [2021], 'The palette: 3d printing evolved'. https://www.mosaicmfg.com/products/palette-3-pro. 34

Mueller, B. and Kochan, D. [1999], 'Laminated object manufacturing for rapid tooling and patternmaking in foundry industry', *Computers in Industry* **39**(1), 47–53. 18

Mueller, S. and Baudisch, P. [2015], 'Laser cutters', *interactions* **22**, 72–74. 43

Mueller, S., Im, S., Gurevich, S., Teibrich, A., Pfisterer, L., Guimbretière, F. and Baudisch, P. [2014], Wireprint: 3d printed previews for fast prototyping, *in* 'Proceedings of the 27th annual ACM symposium on User interface software and technology', pp. 273–280. 43, 66

Mueller, S., Mohr, T., Guenther, K., Frohnhofen, J. and Baudisch, P. [2014], fabrickation: fast 3d printing of functional objects by integrating construction kit building blocks, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', pp. 3827–3834. 42

Murr, L. E., Gaytan, S. M., Ramirez, D. A., Martinez, E., Hernandez, J., Amato, K. N., Shindo, P. W., Medina, F. R. and Wicker, R. B. [2012], 'Metal fabrication by additive manufacturing using laser and electron beam melting technologies', *Journal of Materials Science & Technology* **28**(1), 1–14. 16

Nicholson, P. [1823], *The New Practical Builder and Workman's Companion, Containing a Full Display and Elucidation of the Most Recent and Skilful Methods Pursued by Architects and Artificers... Including, Also, New Treatises on Geometry..., a Summary of the Art of Building..., an Extensive Glossary of the Technical Terms..., and The Theory and Practice of the Five Orders, as Employed in Decorative Architecture*, Vol. 59, Thomas Kelly. 92

Pandey, P. M., Reddy, N. V. and Dhande, S. G. [2003], 'Real time adaptive slicing for fused deposition modelling', *International Journal of Machine Tools and Manufacture* **43**(1), 61–71. 37, 38

Pandey, P. M., Thrimurthulu, K. and Reddy*, N. V. [2004], 'Optimal part deposition orientation in fdm by using a multicriteria genetic algorithm', *International Journal of Production Research* **42**(19), 4069–4089. 40

Pantz, O. and Trabelsi, K. [2008], 'A post-treatment of the homogenization method for shape optimization', *SIAM Journal on Control and Optimization* **47**(3), 1380–1398. 75

Papacharalampopoulos, A., Bikas, H. and Stavropoulos, P. [2018], 'Path planning for the infill of 3d printed parts utilizing hilbert curves', *Procedia Manufacturing* **21**, 757–764. 24, 46

Peano, G. [1890], 'Sur une courbe, qui remplit toute une aire plane', *Mathematische Annalen* **36**(1), 157–160. 46

Peng, H., Briggs, J., Wang, C.-Y., Guo, K., Kider, J., Mueller, S., Baudisch, P. and Guimbretière, F. [2018], Roma: Interactive fabrication with augmented reality and a robotic 3d printer, *in* 'Proceedings of the 2018 CHI conference on human factors in computing systems', pp. 1–12. 43

Peng, H., Wu, R., Marschner, S. and Guimbretière, F. [2016], On-the-fly print: Incremental printing while modelling, *in* 'Proceedings of the 2016 CHI conference on human factors in computing systems', pp. 887–896. 43, 67

Peng, H., Zoran, A. and Guimbretière, F. V. [2015], D-coil: A hands-on approach to digital 3d models design, *in* 'Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems', pp. 1807–1815. 43

Podroužek, J., Marcon, M., Ninčević, K. and Wan-Wendner, R. [2019], 'Bio-inspired 3d infill patterns for additive manufacturing and structural applications', *Materials* **12**(3), 499. 48

Prater, T., Bean, Q., Beshears, R., Rolin, T., Werkheiser, N., Ordonez, E., Ryan, R. and Ledbetter III, F. [2016], 'Summary report on phase i results from the 3d printing in zero g technology demonstration mission, volume i'. 65

Prévost, R., Whiting, E., Lefebvre, S. and Sorkine-Hornung, O. [2013], 'Make it stand: balancing shapes for 3d fabrication', *ACM Transactions on Graphics (TOG)* **32**(4), 1–10. 51

Prusa, J. [2016], 'Original prusa i3 mk2 multi material upgrade'. https://blog.prusa3d.com/original-prusa-i3-mk2-multi-material-upgrade-release_4557/. 34

Ramírez-Gutiérrez, D. L., Cuan-Urquizo, E. and Gómez-Espinosa, A. [2020], 'Adaptable bed for curved-layered fused deposition modeling of nonplanar structures: a proof of concept', *3D Printing and Additive Manufacturing* **7**(4), 198–201. 63

Raut, S., Jatti, V. S., Khedkar, N. K. and Singh, T. [2014], 'Investigation of the effect of built orientation on mechanical properties and total cost of fdm parts', *Procedia materials science* **6**, 1625–1630. 40

Ray, N., Nivoliers, V., Lefebvre, S. and Lévy, B. [2010], Invisible seams, *in* 'Computer Graphics Forum', Vol. 29, Wiley Online Library, pp. 1489–1496. 42

Rebong, R. E., Stewart, K. T., Utreja, A. and Ghoneima, A. A. [2018], 'Accuracy of three-dimensional dental resin models created by fused deposition modeling, stereolithography, and polyjet prototype technologies: A comparative study', *The Angle Orthodontist* **88**(3), 363–369. 15

Reiner, T., Carr, N., Měch, R., Šťava, O., Dachsbacher, C. and Miller, G. [2014], Dual-color mixing for fused deposition modeling printers, *in* 'Computer Graphics Forum', number 2, Wiley Online Library, pp. 479–486. 35

Reiner, T. and Lefebvre, S. [2016], Interactive modeling of support-free shapes for fabrication, *in* 'EUROGRAPHICS'. 39

Roberts, A. and Garboczi, E. J. [2002], 'Elastic properties of model random three-dimensional open-cell solids', *Journal of the Mechanics and Physics of Solids* **50**(1), 33–55. 52

Rybachuk, M., Mauger, C. A., Fiedler, T. and Öchsner, A. [2017], 'Anisotropic mechanical properties of fused deposition modeled parts fabricated by using acrylonitrile butadiene styrene polymer', *Journal of Polymer Engineering* **37**(7), 699–706. 46, 73

Sabourin, E., Houser, S. A. and Bøhn, J. H. [1996], 'Adaptive slicing using stepwise uniform refinement', *Rapid Prototyp J.* **2**(4), 20–26. 38, 58

Sabourin, E., Houser, S. A. and Bøhn, J. H. [1997], 'Accurate exterior, fast interior layered manufacturing', *Rapid Prototyp J.* **3**(2), 44–52. 38

Sammut, E. [2016], 'The prometheus system - intuitive multi-filament 3d printing'. https://www.kickstarter.com/projects/811909269/the-prometheus-system-intuitive-multi-filament-3d. 34

Sanami, M., Ravirala, N., Alderson, K. and Alderson, A. [2014], 'Auxetic materials for sports applications', *Procedia Engineering* **72**, 453–458. 59

Schoen, A. H. [1970], Infinite periodic minimal surfaces without self-intersections, Technical report, NASA. 49

Schöppner, V. and KTP, K. P. [2011], Mechanical properties of fused deposition modeling parts manufactured with ultem* 9085, *in* 'Proceedings of the 69th Annual Technical Conference of the Society of Plastics Engineers (ANTEC'11), Boston, MA, USA', pp. 1–5. 17

Shan, Y., Gan, D. and Mao, H. [2021], 'Curved layer slicing based on isothermal surface', *Procedia Manufacturing* **53**, 484–491. 149

Singamneni, S., Roychoudhury, A., Diegel, O. and Huang, B. [2012], 'Modeling and evaluation of curved layer fused deposition', *Journal of Materials Processing Technology* **212**(1), 27–35. 46, 62, 129

Song, H.-C., Ray, N., Sokolov, D. and Lefebvre, S. [2017], 'Anti-aliasing for fused filament deposition', *Comput. Aided Des.* **89**(C), 25–34. 38, 42, 58, 59

Song, H., Martínez, J., Bedell, P., Vennin, N. and Lefebvre, S. [2019], 'Colored fused filament fabrication', *ACM Transactions on Graphics (TOG)* **38**(5), 1–11. 34, 36

Spencer, B., Laramee, R. S., Chen, G. and Zhang, E. [2009], Evenly spaced streamlines for surfaces: An image-based approach, *in* 'Computer Graphics Forum', Vol. 28, Wiley Online Library, pp. 1618–1631. 46

Steuben, J. C., Iliopoulos, A. P. and Michopoulos, J. G. [2016], 'Implicit slicing for functionally tailored additive manufacturing', *Computer-Aided Design* **77**, 107–119. 73, 85

Sun, S., Brandt, M. and Easton, M. [2017], 'Powder bed fusion processes: An overview', *Laser Additive Manufacturing* pp. 55–77. 16

Taboas, J., Maddox, R., Krebsbach, P. and Hollister, S. [2003], 'Indirect solid free form fabrication of local and global porous, biomimetic and composite 3d polymer-ceramic scaffolds', *Biomaterials* **24**(1), 181–194. 59

Tan, X., Tan, Y., Chow, C., Tor, S. and Yeong, W. [2017], 'Metallic powder-bed based 3d printing of cellular scaffolds for orthopaedic implants: A state-of-the-art review on manufacturing, topological design, mechanical properties and biocompatibility', *Materials Science and Engineering: C* **76**, 1328–1343. 59

Tata, K., Fadel, G., Bagchi, A. and Aziz, N. [1998], 'Efficient slicing for layered manufacturing', *Rapid Prototyp J.* **4**(4), 151–167. 38

Thomas, L.-J., Robert, A. and Richard, T. [2016], 'Curved layer fused filament fabrication using automated tool-path generation', *3D Printing and Additive Manufacturing* **3**, 236–243. 62, 63

Thrimurthulu, K., Pandey, P. M. and Reddy, N. V. [2004], 'Optimum part deposition orientation in fused deposition modeling', *International Journal of Machine Tools and Manufacture* **44**(6), 585 – 594. 40

Tricard, T., Claux, F. and Lefebvre, S. [2020], Ribbed support vaults for 3d printing of hollowed objects, *in* 'Computer Graphics Forum', Vol. 39, Wiley Online Library, pp. 147–159. 50

Tricard, T., Efremov, S., Zanni, C., Neyret, F., Martínez, J. and Lefebvre, S. [2019], 'Procedural phasor noise', *ACM Transactions on Graphics (TOG)* **38**(4), 1–13. 53, 91, 92

Tricard, T., Etienne, J., Zanni, C. and Lefebvre, S. [2021], A brick in the wall: Staggered orientable infills for additive manufacturing, *in* 'Symposium on Computational Fabrication', pp. 1–8. 6

Tricard, T., Tavernier, V., Zanni, C., Martínez, J., Hugron, P.-A., Neyret, F. and Lefebvre, S. [2020], 'Freely orientable microstructures for designing deformable 3d prints', *ACM Transactions on Graphics (TOG)* **39**(6), 1–16. 53, 92, 93, 95, 104, 140

Tumbleston, J. R., Shirvanyants, D., Ermoshkin, N., Janusziewicz, R., Johnson, A. R., Kelly, D., Chen, K., Pinschmidt, R., Rolland, J. P., Ermoshkin, A. et al. [2015], 'Continuous liquid interface production of 3d objects', *Science* **347**(6228), 1349–1352. 18

Tyberg, J. and Bøhn, J. H. [1998], 'Local adaptive slicing', *Rapid Prototyp J.* **4**(3), 118–127. 38

Tyberg, J. and Bøhn, J. H. [1999], 'Fdm systems and local adaptive slicing', *Materials & design* **20**(2), 77–82. 38

Umetani, N. and Schmidt, R. [2013], Cross-sectional Structural Analysis for 3D Printing Optimization, *in* 'SIGGRAPH Asia 2013 Technical Briefs', SA '13, pp. 5:1–5:4. 40

Untzelmann, O. [2018], Near-constant density wireframe meshes for 3d printing. 67

Vanek, J., Galicia, J. A. G. and Benes, B. [2014], Clever support: Efficient support structure generation for digital fabrication, *in* 'Computer graphics forum', Vol. 33, Wiley Online Library, pp. 117–125. 40

Vanek, J., Galicia, J. G., Benes, B., Měch, R., Carr, N., Stava, O. and Miller, G. [2014], Packmerger: A 3d print volume optimizer, *in* 'Computer Graphics Forum', Vol. 33, Wiley Online Library, pp. 322–332. 40

Vouga, E., Höbinger, M., Wallner, J. and Pottmann, H. [2012], 'Design of self-supporting surfaces', *ACM Transactions on Graphics (TOG)* **31**(4), 1–11. 39

Wah, P. K., Murty, K. G., Joneja, A. and Chiu, L. C. [2002], 'Tool path optimization in layered manufacturing', *Iie Transactions* **34**(4), 335–347. 25

Wang, C. C. and Chen, Y. [2013], 'Thickening freeform surfaces for solid fabrication', *Rapid Prototyping Journal* . 61

Wang, L. and Whiting, E. [2016], Buoyancy optimization for computational fabrication, *in* 'Computer Graphics Forum', Vol. 35, Wiley Online Library, pp. 49–58. 51

Wang, T., Li, N., Link, G., Jelonnek, J., Fleischer, J., Dittus, J. and Kupzik, D. [2021], 'Load-dependent path planning method for 3d printing of continuous fiber reinforced plastics', *Composites Part A: Applied Science and Manufacturing* **140**, 106181. 154

Wang, T.-M., Xi, J.-T. and Jin, Y. [2007], 'A model research for prototype warp deformation in the fdm process', *The International Journal of Advanced Manufacturing Technology* **33**(11), 1087–1096. 41

Wang, W., Chao, H., Tong, J., Yang, Z., Tong, X., Li, H., Liu, X. and Liu, L. [2015], 'Saliency-preserving slicing optimization for effective 3d printing', *Comput. Graph. Forum* **34**(6), 148–160. 38

Wang, W. M., Zanni, C. and Kobbelt, L. [2016], Improved surface quality in 3d printing by optimizing the printing direction, *in* 'Computer graphics forum', Vol. 35, Wiley Online Library, pp. 59–70. 40

Wang, W., Wang, T. Y., Yang, Z., Liu, L., Tong, X., Tong, W., Deng, J., Chen, F. and Liu, X. [2013], 'Cost-effective printing of 3d objects with skin-frame structures', *ACM Transactions on Graphics (ToG)* **32**(6), 1–10. 52

Weichel, C., Hardy, J., Alexander, J. and Gellersen, H. [2015], Reform: integrating physical and digital design through bidirectional fabrication, *in* 'Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology', pp. 93–102. 43

Werkheiser, M. J., Dunn, J., Snyder, M. P., Edmunson, J., Cooper, K. and Johnston, M. M. [2014], 3d printing in zero-g iss technology demonstration, *in* 'AIAA SPACE 2014 Conference and Exposition', p. 4470. 65

White, D. [2003], 'Ultrasonic object consolidation'. US Patent 6,519,500. 18

Williams, J. D. and Deckard, C. R. [1998], 'Advances in modeling the effects of selected parameters on the sls process', *Rapid Prototyping Journal* . 16

Willis, K. D., Xu, C., Wu, K.-J., Levin, G. and Gross, M. D. [2010], Interactive fabrication: new interfaces for digital fabrication, *in* 'Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction', pp. 69–72. 43

Wu, C., Dai, C., Fang, G., Liu, Y.-J. and Wang, C. C. [2017], Robofdm: A robotic system for support-free fabrication using fdm, *in* '2017 IEEE International Conference on Robotics and Automation (ICRA)', IEEE, pp. 1175–1180. 64

Wu, J., Aage, N., Westermann, R. and Sigmund, O. [2017], 'Infill optimization for additive manufacturing—approaching bone-like porous structures', *IEEE transactions on visualization and computer graphics* **24**(2), 1127–1140. 52

Wu, J., Wang, C. C., Zhang, X. and Westermann, R. [2016], 'Self-supporting rhombic infill structures for additive manufacturing', *Computer-Aided Design* **80**, 32–42. 51, 84

Wu, J., Wang, W. and Gao, X. [2019], 'Design and optimization of conforming lattice structures', *IEEE transactions on visualization and computer graphics* **27**(1), 43–56. 51

Wu, R., Peng, H., Guimbretière, F. and Marschner, S. [2016], 'Printing arbitrary meshes with a 5dof wireframe printer', *ACM Transactions on Graphics (TOG)* **35**(4), 1–9. 68

Wu, W., Geng, P., Li, G., Zhao, D., Zhang, H. and Zhao, J. [2015], 'Influence of layer thickness and raster angle on the mechanical properties of 3d-printed peek and a comparative mechanical study between peek and abs', *Materials* **8**(9), 5834–5846. 17

Wyvill, G., McPheeters, C. and Wyvill, B. [1986], Soft objects, *in* 'Advanced Computer Graphics', Springer, pp. 113–128. 95

Xiaomao, H., Chunsheng, Y. and Yongjun, H. [2011], 'Tool path planning based on endpoint build-in optimization in rapid prototyping', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* **225**(12), 2919–2926. 25

Xie, F., Chen, L., Li, Z. and Tang, K. [2020], 'Path smoothing and feed rate planning for robotic curved layer additive manufacturing', *Robotics and Computer-Integrated Manufacturing* **65**, 101967. 62

Xiong, Y., Park, S.-i., Padmanathan, S., Dharmawan, A. G., Foong, S., Rosen, D. W. and Soh, G. S. [2019], 'Process planning for adaptive contour parallel toolpath in additive manufacturing with variable bead width', *The International Journal of Advanced Manufacturing Technology* **105**(10), 4159–4170. 25, 46

Xu, K., Li, Y., Chen, L. and Tang, K. [2019], 'Curved layer based process planning for multi-axis volume printing of freeform parts', *Computer-Aided Design* **114**, 51–63. 65

Yaman, U. [2018], 'Shrinkage compensation of holes via shrinkage of interior structure in fdm process', *The International Journal of Advanced Manufacturing Technology* **94**(5), 2187–2197. 41

Yamanaka, D., Suzuki, H. and Ohtake, Y. [2014], Density aware shape modeling to control mass properties of 3d printed objects, *in* 'SIGGRAPH Asia 2014 Technical Briefs', pp. 1–4. 51

Yang, H., Lim, J. C., Liu, Y., Qi, X., Yap, Y. L., Dikshit, V., Yeong, W. Y. and Wei, J. [2017], 'Performance evaluation of projet multi-material jetting 3d printer', *Virtual and physical prototyping* **12**(1), 95–103. 16

Yang, L., Harrysson, O., West, H. and Cormier, D. [2015], 'Mechanical properties of 3d re-entrant honeycomb auxetic structures realized via additive manufacturing', *International Journal of Solids and Structures* **69**, 475–490. 61

Yang, Y., Loh, H. T., Fuh, J. and Wang, Y. [2002], 'Equidistant path generation for improving scanning efficiency in layered manufacturing', *Rapid Prototyping Journal* . 45

Yeong, W.-Y., Chua, C.-K., Leong, K.-F., Chandrasekaran, M. and Lee, M.-W. [2005], 'Development of scaffolds for tissue engineering using a 3d inkjet model maker', *Virtual modelling and rapid manufacturing-advanced research in virtual and rapid prototyping* pp. 115–118. 16

Yigit, I. E. and Lazoglu, I. [2019], 'Helical slicing method for material extrusion-based robotic additive manufacturing', *Progress in Additive Manufacturing* **4**(3), 225–232. 59

Zhang, X., Le, X., Panotopoulou, A., Whiting, E. and Wang, C. C. L. [2015], 'Perceptual models of preference in 3d printing direction', *ACM Trans. Graph.* **34**(6), 215:1–215:12. 40

Zhang, X., Xia, Y., Wang, J., Yang, Z., Tu, C. and Wang, W. [2015], 'Medial axis tree—an internal supporting structure for 3d printing', *Computer Aided Geometric Design* **35**, 149–162. 52

Zhao, D., Guo, W. and Gao, F. [2020], 'Research on curved layer fused deposition modeling with a variable extruded filament', *Journal of Computing and Information Science in Engineering* **20**(4). 61

Zhao, D., Li, T., Shen, B., Jiang, Y., Guo, W. and Gao, F. [2020], 'A multi-dof rotary 3d printer: machine design, performance analysis and process planning of curved layer fused deposition modeling (clfdm)', *Rapid Prototyping Journal* . 57, 62, 63

Zhao, G., Ma, G., Xiao, W. and Tian, Y. [2019], 'Feature-based five-axis path planning method for robotic additive manufacturing', *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **233**(5), 1412–1424. 62

Zhao, H., Gu, F., Huang, Q.-X., Garcia, J., Chen, Y., Tu, C., Benes, B., Zhang, H., Cohen-Or, D. and Chen, B. [2016], 'Connected fermat spirals for layered fabrication', *ACM Transactions on Graphics (TOG)* **35**(4), 1–10. 24, 62

Zhao, H., Hong, C., Lin, J., Jin, X. and Xu, W. [2016], 'Make it swing: Fabricating personalized roly-poly toys', *Computer Aided Geometric Design* **43**, 226–236. 51

Ziaee, M. and Crane, N. B. [2019], 'Binder jetting: A review of process, materials, and methods', *Additive Manufacturing* **28**, 781–801. 17